36th Computational Complexity Conference

CCC 2021, July 20–23, 2021, Toronto, Ontario, Canada (Virtual Conference)

Edited by

Valentine Kabanets

LIPICS

COMPUTATIONAL
COMPLEXITY
CONFERENCE

Editors

Valentine Kabanets
School of Computing Science
Simon Fraser University
Burnaby, BC
Canada
kabanets@cs.sfu.ca

ACM Classification 2012
Theory of computation

ISBN 978-3-95977-193-1

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at https://www.dagstuhl.de/dagpub/978-3-95977-193-1.

Publication date
July, 2021

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at https://portal.dnb.de.

Digital Object Identifier: 10.4230/LIPIcs.CCC.2021.0

ISBN 978-3-95977-193-1 **ISSN 1868-8969** **https://www.dagstuhl.de/lipics**

LIPIcs – Leibniz International Proceedings in Informatics

LIPIcs is a series of high-quality conference proceedings across all fields in informatics. LIPIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 1868-8969

https://www.dagstuhl.de/lipics

Contents

Papers

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

The papers in this volume were accepted for presentation at the 36th Computational Complexity Conference (CCC 2021), held between July 20–22, 2021, in a virtual online format. CCC 2021 was originally scheduled to be held in Toronto, Canada, but due to the public health measures related to Covid-19 still in place, the online format was used instead. The conference is organized by the Computational Complexity Foundation (CCF) in cooperation with the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT) and the European Association for Theoretical Computer Science (EATCS).

The call for papers sought original research papers in all areas of computational complexity theory. Of the 116 submissions, the program committee selected 41 for presentation at the conference.

The program committee would like to thank everyone involved in the conference, including all those who submitted papers for consideration as well as the reviewers (listed separately) for their scientific contributions; the board of trustees of the Computational Complexity Foundation and especially its president Venkatesan Guruswami, and secretary Ashwin Nayak for their advice and assistance; Shubhangi Saraf for sharing her knowledge as prior PC chair for CCC; the Local Arrangements Committee chair Benjamin Rossman; Eric Allender for the invited talk; Meena Mahajan for her help with the submission server setup and editing of the proceedings; and Michael Wagner for coordinating the production of these proceedings.

Valentine Kabanets
Program Committee Chair, on behalf of the Program Committee

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

Awards

The program committee of the 36th Computational Complexity Conference is very pleased to present the **Best Student Paper Award** to Yaroslav Alekseev for his paper

A Lower Bound for Polynomial Calculus with Extension Rule.

Conference Organization

Program Committee

Arkadev Chattopadhyay, Tata Institute of Fundamental Research Mumbai
Irit Dinur, Weizmann Institute of Science
Yuval Ishai, Technion
Valentine Kabanets (Chair), Simon Fraser University
Swastik Kopparty, Rutgers University
Nutan Limaye, Indian Institute of Technology Bombay
Ryan O'Donnell, Carnegie Mellon University
Igor Carboni Oliveira, University of Warwick
Alexander Razborov, University of Chicago/Steklov Institute
Barna Saha, University of California Berkeley
Emanuele Viola, Northeastern University
Henry Yuen, University of Toronto/Columbia University

Local Arrangements Committee

Aleksandar Nikolov, University of Toronto
Benjamin Rossman (Chair), Duke University
Sushant Sachdeva, University of Toronto
Henry Yuen, University of Toronto/Columbia University

Board of Trustees

External Reviewers

Eric Allender
Josh Alman
Robert Andrews
Sepehr Assadi
Paul Beame
Shalev Ben-David
Amey Bhangale
Vishwas Bhargava
Vijay Bhattiprolu
Markus Bläser
Ilario Bonacina
Joshua Brody
Peter Buergisser
Boris Bukh
Marco Carmosino
Bruno Cavalar
Amit Chakrabarti
Diptarka Chakraborty
Eshan Chattopadhyay
Prasad Chaugule
Xi Chen
Ashish Chiplunkar
Gil Cohen
Daniel Dadush
Mina Dalirrooyfard
Anindya De
Rafael Mendes de Oliveira
Susanna F. de Rezende
Ronald de Wolf
Holger Dell
Talya Eden
Yuval Filmus
Noah Fleming
Michael Forbes
Cole Franks
Bin Fu
Abdul Ghani
Sumanta Ghosh
Leslie Ann Goldberg
Alexander Golovnev
Sivakanth Gopi
Joshua Grochow
Tom Gur
Rohit Gurjar
Shuichi Hirahara
Samuel Hopkins
William Hoza
Pavel Hrubes
Hsin-Yuan Huang
Xuangui Huang
Yichen Huang
Christian Ikenmeyer
Rahul Ilango
Peter Ivanov
Gabor Ivanyos
Mark Jerrum
C.S. Karthik
Sanjeev Khanna
Alexander Knop
Pascal Koiran
Antonina Kolokolova
Sajin Koroth
Robin Kothari
Michal Koucky
Mrinal Kumar
Dmitriy Kunisky
Chin Ho Lee
Tongyang Li
Xin Li
Andrea Lincoln
Bruno Loff
Zhenjian Lu
Joshua Maglione
Meena Mahajan
Guillaume Malod
Or Meir
Ian Mertz
Sidhanth Mohanty
Chandra Kanta Mohapatra
Jonathan Mosheiff
Partha Mukhopadhyay
Sagnik Mukhopadhyay
Ashwin Nayak
Rafael Oliveira
Shuo Pang
Fahad Panolan
Fedor Part
Shir Peleg
Stephen Piddock
Aaron Potechin
Aditya Potukuchi
Kevin Pratt
Pavel Pudlak
Jaikumar Radhakrishnan
Akshay Ramachandran
Shravas Rao
Ran Raz
Nicolas Resch
David Richerby
Robert Robere
Dana Ron
Noga Ron-Zewi
Chandan Saha
Rahul Santhanam
Swagato Sanyal
Ramprasad Saptharishi
Will Sawin
Nitin Saxena
Gil Segev
Ronen Shaltiel
Suhail Sherif
Alexander Sherstov
Igor Shinkar
Amir Shpilka
Adi Shraibman
Makrand Sinha
Amit Sinhababu
Dmitry Sokolov
Srikanth Srinivasan
Manuel Stoeckl
Xiaoming Sun
Xiaorui Sun
Avishay Tal
Till Tantau
Sébastien Tavenas
Anamay Tengse
Raghunath Tewari
Samarth Tiwari
Iddo Tzameret
Marc Vinyals
Ben Lee Volk
Ilya Volkovich

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

Nikhil Vyas
S. Matthew Weinberg
James Wilson
Jinshan Wu
Penghui Yao
Yuichi Yoshida
Huacheng Yu
Uri Zwick

Rate Amplification and Query-Efficient Distance Amplification for Linear LCC and LDC

Gil Cohen
Department of Computer Science, Tel Aviv University, Israel

Tal Yankovitz
Department of Computer Science, Tel Aviv University, Israel

Abstract

The main contribution of this work is a rate amplification procedure for LCC. Our procedure converts any q-query linear LCC, having rate ρ and, say, constant distance to an asymptotically good LCC with $q^{\text{poly}(1/\rho)}$ queries.

Our second contribution is a distance amplification procedure for LDC that converts any linear LDC with distance δ and, say, constant rate to an asymptotically good LDC. The query complexity only suffers a multiplicative overhead that is roughly equal to the query complexity of a length $1/\delta$ asymptotically good LDC. This improves upon the $\text{poly}(1/\delta)$ overhead obtained by the AEL distance amplification procedure [2, 1].

Our work establishes that the construction of asymptotically good LDC and LCC is reduced, with a minor overhead in query complexity, to the problem of constructing a vanishing rate linear LCC and a (rapidly) vanishing distance linear LDC, respectively.

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases Locally decodable codes, Locally correctable codes

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.1

Funding The research leading to these results has received funding from the ERC starting grant 949499, the Israel Science Foundation grant 1569/18 and from the Azrieli Faculty Fellowship.

1 Introduction

Coding theory addresses the problem of communicating over an imperfect channel. Classically, the setting is as follows. Alice wishes to communicate a message m to Bob over a channel that can be tampered by an adversary. How should Alice encode m so that if the amount of errors is not excessive, Bob would be able to recover m? To this end, error-correcting codes were first introduced [33]. Recall that a function $C\colon \Sigma^k \to \Sigma^n$ is an *error-correcting code* with distance δ if for every distinct $x, y \in \Sigma^k$, $\mathsf{dist}(C(x), C(y)) \geq \delta$, where dist is the relative Hamming distance.[1] The *rate* of the code C is given by $\rho = k/n$. Using an error-correcting code, Alice can encode her message $m \in \Sigma^k$ and send the resulting codeword $C(m)$. Assuming the fraction of errors is less than $\delta/2$, Bob can decode m from the received z by finding the codeword closest to z. When there is more than one possible message length, we consider a *code family*, which is a family of functions in which each function is a code, and there is one code per message length k. A code family is *asymptotically good* if both the rate and distance of every code in the family are uniformly bounded below by constants $\rho > 0$ and $\delta > 0$, respectively.

[1] Note that what we call here distance δ is in many cases referred to as *relative* distance.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 1; pp. 1:1–1:57
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 Locally decodable codes and locally correctable codes

Consider the scenario in which Bob is not interested in the entire original message m, but rather in a specific symbol m_i for some $i \in [k]$. A simple, though wasteful solution, is for Bob to decode the entire message m and ignore all symbols but for m_i. However, it is desirable to compute m_i by reading much fewer than n entries of z. *Locally decodable codes (LDC)* are a class of error-correcting codes that have this very strong decoding capability. Another scenario of interest is the one in which Bob needs to know a specific symbol of the *codeword* $C(m)_j$ for some $j \in [n]$, while reading as few symbols as possible. Codes that allow this are called *locally correctable codes (LCC)*. We turn to give the formal definition.

▶ **Definition 1** (Locally decodable codes (LDC)). *A code $C : \Sigma^k \to \Sigma^n$ is (q, δ, ε)-*locally decodable *if there exists a randomized algorithm D, called a* local decoder*, that is given $i \in [k]$ as input and an oracle access to $z \in \Sigma^n$, and has the following guarantee. For every $i \in [k]$, $m \in \Sigma^k$ and $z \in \Sigma^n$ such that $\mathsf{dist}(C(m), z) \le \delta$ it holds that $\mathbf{Pr}\left[D^z(i) \neq m_i\right] \le \varepsilon$. Moreover, D makes at most q queries to z.*

▶ **Definition 2** (Locally correctable codes (LCC)). *A code $C \subseteq \Sigma^n$ is (q, δ, ε)-*locally correctable *if there exists a randomized algorithm D, called a* local corrector*, that is given $j \in [n]$ as input and an oracle access to $z \in \Sigma^n$, and has the following guarantee. For every $j \in [n]$, $c \in C$ and $z \in \Sigma^n$ such that $\mathsf{dist}(c, z) \le \delta$ it holds that $\mathbf{Pr}\left[D^z(j) \neq c_j\right] \le \varepsilon$. Moreover, D makes at most q queries to z.*

We place z in the upper script in our notation $D^z(i)$ to stress that the number of symbols read from z by D is of importance. The parameter q is called the *query complexity*, and δ is the *local error decoding radius*, in the case LDC, and *local error correction radius* in the case of LCC. However, we also refer to δ, somewhat inaccurately, as the *local distance* of the code. From here on, we do not make any explicit reference to the "global" distance of a code and so we refer to the local distance simply as the distance. Throughout the paper, we only consider non-adaptive LDC and LCC, defined next. Informally, these are code in which the local decoder (or corrector) samples the entries to be read before the querying step takes place. Our results only hold for non-adaptive LDCs and LCCs. For ease of discussion, throughout the introduction we ignore the error parameter ε. More precisely, when stating our results, every LDC or LCC (both in the hypothesis as well as in the LDC or LCC guaranteed by the theorem) has constant error.

A brief history of LDC and LCC

Locally decodable codes were first explicitly defined by Katz and Trevisan [23]. However, codes with local guarantees have been used by complexity theorists even before (e.g., [8, 15, 16, 6]) and have been around, implicitly, in the coding theory community almost from the get going [30]. LDC, LCC, and related notions such as locally testable codes (LTC), were intensively studied by theoretical computer scientists motivated by PCPs [3, 4, 5, 18], program checking [10, 29, 32], circuit lower bounds [12], derandomization [6, 35, 36], and private information retrieval [11] to name a few. LDC and LCC are very related notions. Clearly, an LCC with a systematic encoding[2] is also an LDC and so, in particular, linear LCC induce LDC. Of note, it is not yet known in which scenarios LCC are strictly stronger objects compared to LDC.

[2] An encoding from messages to codewords is called *systematic* if the symbols of each message are embedded in its mapped codeword.

An intensive research effort is devoted to the construction of local codes (see the excellent survey for LDC [39]). Roughly, the literature can be partitioned to two. The first research path (see e.g., [40, 25, 14, 13] and references therein) has the goal of obtaining LDC or LCC with a given, small, number of queries, and an effort is made to maximize the rate while maintaining constant distance. The second research path, which has received much attention in recent years [27, 20, 22, 26, 19], and is the focus of this paper, insists on asymptotically good codes and aims at minimizing the number of queries.

It is known [23, 38] that asymptotically good LDC require $q = \Omega(\log n)$ queries. Whether this bound is tight is a fundamental, major open problem, regardless of explicitness. The Reed Muller code is perhaps the earliest non-trivial example of LDC and LCC. It can achieve query complexity n^{ν} for any desired constant $\nu > 0$. However, the rate deteriorates rapidly as $\nu \to 0$. In fact, up until the introduction of *multiplicity codes* by Kopparty, Saraf and Yekhanin [27] no (non-trivial) LDC or LCC with rate higher than $1/2$ were known. Guo, Kopparty and Saraf [20] introduced the notion of lifting of codes which gave a second high-rate LDC and LCC, also algebraic in nature. A combinatorial high-rate construction of an LCC was obtained by Hemenway, Ostrovsky and Wootters [22] (see also [28]).

Despite this exciting sequence of works which allowed for better rate and introduced various interesting techniques, the above constructions all have query complexity $n^{\Theta(1)}$. The fact that three very different constructions were stuck at polynomial query complexity raised the question of whether $n^{o(1)}$-query asymptotically good LDC or LCC exist. This question was resolved in a seminal work by Kopparty, Meir, Ron-Zewi and Saraf [26] who obtained LCC with query complexity $q = 2^{\widetilde{O}(\sqrt{\log n})} = n^{o(1)}$. To obtain their result, the authors first observed that by instantiating multiplicity codes [27] in a certain regime of parameters, one can get the stated query complexity q above albeit at the cost of having vanishing distance $\delta = 1/(\log n)^{\Theta(1)}$. Then, in order to get codes with constant distance, the authors invoked a distance amplification procedure due to Alon et al. [2, 1]. Kopparty et al. [26] showed that the AEL distance amplification procedure, which was originally introduced in the context of linear-time erasure codes, allows one to convert, in a black-box manner, an LCC with distance δ and query complexity q to an LCC with constant distance and query complexity $q_{\mathrm{new}} = q \cdot \mathrm{poly}(1/\delta)$. This more than sufficed for [26] as, in their setting, $q = (1/\delta)^{\omega(1)}$, and so the cost of the distance amplification is negligible. The LCC constructed in Kopparty et al. [26] are linear and thus yield LDC as well, and in fact in the same work the state-of-the-art LTC are constructed using the AEL distance amplification procedure.

1.2 Our contribution

Given the pivotal role of the AEL distance amplification procedure in the state-of-the-art constructions of LDC and LCC (as well as LTC) one is prompt to ask whether the $\mathrm{poly}(1/\delta)$ multiplicative cost in query complexity is inherent. If such is the case, when aiming at $\mathrm{poly}(\log n)$-query complexity, a requirement for constant distance can only be relaxed to distance $1/\mathrm{poly}(\log n)$ which, although proved extremely useful [26], might be restrictive for obtaining better codes.

More generally, the natural question that is raised is to what extent the construction of asymptotically good LDC/LCC can be reduced to the non-asymptotically good variants as they, in turn, may admit low query constructions. The main contribution of this work is the first rate amplification procedure for linear LCC as we elaborate on next (see Section 1.2.1). As our second contribution, we obtain a significantly improved distance amplification procedure (see Section 1.2.2).

1.2.1 Rate amplification

It is unclear to us if rate can be amplified deterministically in general, regardless of locality, in any meaningful formalization. Puncturing is a coding-theoretic technique that allows one to obtain better rates. However, it only seems to work when tailored to specific codes with certain structure or, otherwise, using a randomized encoding. Nonetheless, our main contribution is a devising a rate amplification procedure for linear non-adaptive LCC. To the best of our knowledge, all known constructions of LCC are of this kind. Among these are Reed-Muller codes (and therefore also the Hadamard code) as well as codes obtained by lifting [20], and Multiplicity codes.

▶ **Theorem 3** (Main result). *Assume one has a non-adaptive linear $(q, \delta = \Omega(1))$-LCC with block-length n_0 having rate $\rho = \rho(n_0)$. Then, for every integers $\ell, c \geq 1$ such that $\ell^2 < c < \log n_0$, one can obtain a non-adaptive linear $(q_{\text{new}}, \delta_{\text{new}})$-LCC with block length $n \approx n_0^\ell$, having rate ρ_{new}, where*

$$\begin{aligned} q_{\text{new}} &= (cq)^{\text{poly}(\ell)}, \\ \delta_{\text{new}} &= (cq)^{-\text{poly}(\ell)}, \\ \rho_{\text{new}} &= 1 - (1-\rho)^\ell - O\left(\frac{\ell^2}{c}\right). \end{aligned}$$

Theorem 3, when invoked with $\ell \approx 1/\rho$ and $c \approx 1/\rho^2$, and combined with a distance amplification procedure, yields the following corollary.

▶ **Corollary 4.** *Assume one has a family of constant distance non-adaptive linear LCC with rate $\rho(n) \geq \frac{1}{\sqrt{\log n}}$ and query complexity $q(n)$. Then, for every constant*[3] *$\alpha > 0$ one can obtain asymptotically good LCC with rate $1 - \alpha$ on block length n with query complexity $q_{\text{new}} = (q(n) \log n)^{\text{poly}(1/\rho(n))}$.*

1.2.2 Query-efficient distance amplification

The second result of this work is a significantly improved distance amplification procedure for LDC. Roughly speaking, we are able to reduce the $\text{poly}(1/\delta)$ multiplicative factor in query complexity to the query complexity of an asymptotically good LDC on message length $1/\delta$. More precisely,

▶ **Theorem 5** (Query-efficient distance amplification; informal). *Assume one has a block-length-n LDC with distance δ, constant rate, and query complexity q. Assume further one has a family of asymptotically good LDC where on message length k, the query complexity is q_k. Then, one can obtain asymptotically good LDC with query complexity* [4]

$$q_{\text{new}} = q \cdot q_{O(1/\delta)} \cdot O(\log(1/\delta) \log n). \tag{1.1}$$

Note that by using a standard error-correcting code, which has $q_k = n = \Theta(k)$, Theorem 5 gives back the parameters of the AEL distance amplification procedure. However, one can do much better. Indeed, by using the state-of-the-art LDC [26] which has $q_k = 2^{\widetilde{O}(\sqrt{\log k})}$,

[3] The result holds also for sub-constant α, and the assumption is made only for simplicity. See Theorem 46 for the formal, more general, version.

[4] If the family of LDC in the hypothesis has sufficiently low error, the query complexity is even smaller $q_{\text{new}} = q \cdot q_{O(1/\delta)} q_{O(\log(1/\delta))}$.

one get $q_{\text{new}} = q \cdot (1/\delta)^{o(1)} \log n$. More generally, Theorem 5 states that the lower the query complexity of the asymptotically good codes which one starts with is, the more query-efficient is the distance amplification. This "rich getting richer" type of result opens a path to recursive constructions as, indeed, several of our applications are based on. We stress that unlike the AEL distance amplification procedure, ours exploits the local *decodability* requirement and so it works for LDC but not for LCC. The only other technique in the literature that we are aware of that exploits the difference between decodability and correctability, and thus separates LDC from LCC in terms of techniques is matching vectors based constructions. We further remark that, for ease of discussion, Theorem 5 is stated without any reference to explicitness. Indeed, we currently lack satisfactory understanding of LDC in the more fundamental information-theoretic level. In any case, explicitness does not cost much in our reduction, and the only change in the theorem's statement when insisting on explicit reductions is replacing Equation (1.1) by roughly $q_{\text{new}} = q \cdot q_{(1/\delta)^{1+\alpha}} \log n$ for any desired constant $\alpha > 0$.

We turn to draw several corollaries of Theorem 5, but first set the context. Given the Katz-Trevisan $\Omega(\log n)$ lower bound on the query complexity of asymptotically good LDC, and reassured by [26] that $n^{o(1)}$-query LDC exist, the next natural goal is to try and construct, or even more fundamentally, prove the existence of LDC with poly-logarithmic (or perhaps a more modest quasi poly-logarithmic $2^{\text{poly}(\log\log n)}$) number of queries. With this goal in mind, the AEL distance amplification procedure allows one to relax her effort and construct LDC with distance $\delta = 1/\text{poly}(\log n)$ or slightly lower. Multiplicity codes are indeed a great example where such a relaxation of the distance requirement allows one to obtain much better query complexity. Using Theorem 5, we are able to obtain a reduction to LDC having exponentially lower distance $\delta = 1/\text{poly}(n)$.

▶ **Corollary 6** (Amplifying polynomially-small distance)**.** *Let $0 < \alpha < 1$ be an arbitrary constant. Assume there exists a family of LDC with distance $\delta = n^{-\alpha}$, rate $1 - 1/(\log n)^2$, and query complexity $q(n)$ for block length n. Then, for infinitely many n's, there exists an asymptotically good LDC on block-length n with query complexity $q_{\text{new}} = q(n)^{O(\log\log n)}$.*

Corollary 6 implies that for constructing asymptotically good LDC with $q = 2^{\text{poly}(\log\log n)}$ queries, it suffices to construct LDC with extremely poor distance $\delta = 1/\text{poly}(n)$ for the same asymptotic query complexity. In fact, we can even amplify extremely small distance $\delta = n^{-(1-o(1))}$ assuming the rate is slightly larger. One instantiation is as follows.

▶ **Corollary 7.** *Let $c \geq 1$ be any constant. Assume there exists a family of LDC with distance $\delta = n^{-(1-\frac{1}{(\log\log n)^c})}$, rate $\rho = 1 - \frac{1}{(\log n)^{c+2}}$, and query complexity $q(n)$ for block-length n. Then, for infinitely many n's, there exists an asymptotically good LDC on block-length n having query complexity $q_{\text{new}} = q(n)^{O((\log\log n)^{c+1})}$.*

A third interesting application of Theorem 5 is when the distance to be amplified is larger than $1/\text{poly}(n)$, though still very small.

▶ **Corollary 8.** *Let $\alpha < 1$ be an arbitrary constant. Assume there exists a distance $\delta = 2^{-(\log n)^\alpha}$ LDC having rate $1 - O(1/\log\log n)$, and query complexity $q(n)$ for block-length n. Then, for infinitely many n's, there exists an asymptotically good LDC on block length n with query complexity $q_{\text{new}} = q(n)^{O(\log\log\log n)}$.*

We conclude this section by noting that the Katz-Trevisan bound [23] holds also for sub-constant distance. Quantitatively, the query complexity of constant rate codes with distance δ is $\Omega(\log(\delta n/\log n))$. Thus, even for distance $n^{-\alpha}$, the $\Omega(\log n)$ lower bound holds.

2 Proof overview

In this section we give a brief and informal overview of the ideas that go into our proofs.

2.1 A characterization of non-adaptive linear LCC

To obtain our rate amplification procedure we lay a characterization of non-adaptive linear LCC. We remark that a very similar characterization was given by [23] for LDC, who defined the notion of smooth-codes.

▶ **Definition 9** (Smooth locally recoverable sets; simplified version). *Let Σ, P be arbitrary sets. We say that $C \subseteq \Sigma^P$ is (q, τ)-*smooth locally recoverable *(SLR for short) if there exists a randomized algorithm* Rec*, called a* recovering procedure*, that when given as input $p \in P$ and an oracle access to $c \in C$, outputs $\mathsf{Rec}^c(p) = c_p$ by making at most q queries to c. Moreover, for every $c \in C$ and $p, r \in P$ (not necessarily distinct),*

$$\mathbf{Pr}[\mathsf{Rec}^c(p) \text{ queries } c_r] \leq \tau. \tag{2.1}$$

We will focus on SLR in which Σ is a field and C is a vector space over Σ. In such case we say that C is linear. Of course, it is trivial to construct a $(1, 1)$-SLR. Indeed, simply query c_p and output the result. The challenge is to recover c_p without being able to "focus" on any particular entry. This is captured by Equation (2.1) where τ–the *smoothness parameter*– bounds the probability a given entry is allowed to be queried. The formal definition of SLR (see Definition 21) also allows the recovering procedure to output a special "failure" symbol $\perp$ with small probability. For ease of discussion, we ignore this here. We have the following easy claim showing that SLR yield LCC. As a result, linear SLR induce LDC.

▷ Claim 10. Let $C \subseteq \Sigma^P$ be a (q, τ)-SLR. Then, C is a (q, δ)-LCC with $\delta = \Omega\left(1/(q\tau|P|)\right)$.

For the straightforward proof, see Section 4 and, in particular, Claim 22. We also have the following (less obvious) claim, showing that, assuming linearity and non-adaptiveness, the other direction also holds, namely, LCC yield SLR.

▷ Claim 11. Let $C \subseteq \Sigma^P$ be a non-adaptive linear (q, δ)-LCC. Then, C is a (linear) (q, τ)-SLR with $\tau = q/(\delta|P|)$.

This claim and its proof correspond to Theorem 1 of [23] with the terminology of smooth-codes. For the more formal statement which also takes into account the error parameter and field size, see Claim 23. We remark here that for the proof of Claim 23, we construct a recovering procedure based on the local corrector of the given LCC. However, the key idea is to consider the distributions this local corrector induces while ignoring how it reconstruct the symbol after performing the queries.

Note that the lowest sensible value for τ is at about $q/|P|$. Indeed, this will be the case if each of the q queries is marginally uniform over P, and assuming nothing about the correlations between the queries. For such τ, if C is linear then, By Claim 10, it yields an LCC with $\delta = \Omega(1/q^2)$. The distance can then be amplified to constant using our distance amplification procedure to yield query complexity $q^{2+o(1)}$ (or using AEL's procedure to get poly(q) queries).

2.1.1 Dual SLR and their induced SLR

By Claim 10 and Claim 11, every linear SLR is an LCC, and every linear non-adaptive LCC is a linear SLR. Our rate amplification procedure works for non-adaptive linear SLR, and thus for any non-adaptive linear LCC. In order to amplify the rate of such an SLR, we show that the dual of every non-adaptive linear SLR has a certain structure, which we use to amplify the rate.

Working with dual of codes in the context of LDC or LCC is a very natural approach, and has been explored previously (e.g., [24, 7]), but to the best of our knowledge, the definition of dual SLR as given below is new. We start by setting some notation. Let P be a set, $\mathbb{F}$ a finite field, and $\mathbb{F}^P$ the set of all functions $\{f : P \to \mathbb{F}\}$. Note that $\mathbb{F}^P$ has a natural $\mathbb{F}$-vector space structure. We consider the natural inner product $\langle \cdot, \cdot \rangle : \mathbb{F}^P \times \mathbb{F}^P \to \mathbb{F}$ that is defined, for $f, g \in \mathbb{F}^P$, by $\langle f, g \rangle = \sum_{p \in P} f(p)g(p)$. For $f \in \mathbb{F}^P$ we denote $|f| = |P \setminus f^{-1}(0)|$. For $p \in P$ define $\mathcal{F}_p = \{f \in \mathbb{F}^P \mid f(p) \neq 0\}$.

The following definition captures the structural properties of the dual of an SLR, which we need for the rate amplification.

▶ **Definition 12** (Dual SLR; simplified version). *Let P be a set, $\mathbb{F}$ a field. Let $\mathcal{D} = \{D_p \mid p \in P\}$ be a collection of distributions, where for each $p \in P$, $\mathsf{supp}(D_p) \subseteq \mathcal{F}_p$. Set $S \triangleq \bigcup_{p \in P} \mathsf{supp}(D_p)$. The collection $\mathcal{D}$ is said to be a (q, τ, ρ)-*dual SLR *provided the following holds:*

1. *$|f| \leq q$ for all $f \in S$.*
2. *For every pair of distinct $p, r \in P$, it holds that*
$$\Pr_{f \sim D_p}[f(r) \neq 0] \leq \tau.$$
3. *Last,* $\dim \mathrm{Span}(S) \leq (1 - \rho)|P|$.

We call q the *query complexity* of the dual SLR, τ its *smoothness* and ρ its *rate*. The linear subspace $S^{\perp}$ of $\mathbb{F}^P$ is called the *induced SLR* from $\mathcal{D}$. As the name suggests, the induced SLR $S^{\perp}$ is indeed an SLR. More precisely, it is a $(q-1, \tau)$ SLR with rate ρ (see Lemma 26). It is for the class of dual-induced SLR that we are able to devise our rate amplification procedures. Let p be a prime power. As an example, one can directly show that, say, the two-dimensional Reed-Muller code over $\mathbb{F}_p$ with total-degree $p - 2$ is an induced SLR from a $(q = p - 1, \tau = \frac{1}{p+1}, \rho = \frac{1}{2} - o(1))$-dual SLR. As mentioned, any linear non-adaptive LCC is a linear SLR, and thus induces a dual SLR.

2.2 Rate amplification for dual-induced SLR

For simplicity, we describe our rate amplification procedure only for $\ell = 2$, where ℓ is as in the notation of Theorem 3. We briefly explain how to handle larger ℓ's in Section 2.2.3. Assume $\mathcal{D}$ is a (q, τ, ρ)-dual SLR on $\mathbb{F}^P$ where the rate ρ is the parameter we wish to amplify. Consider the mapping $\Phi : (\mathbb{F}^P)^2 \to \mathbb{F}^{P^2}$ that maps a pair of functions $f_1, f_2 \in \mathbb{F}^P$ to the function $\Phi(f_1, f_2) : P^2 \to \mathbb{F}$ given by $\Phi(f_1, f_2)(p_1, p_2) = f_1(p_1)f_2(p_2)$. Note that this is simply the tensor product.

We now show how to convert our poor-rate dual SLR $\mathcal{D}$ to a new dual-SLR with a better rate. Formally, consider the (q_2, τ_2, ρ_2)-dual SLR $\mathcal{D}^2 = \{D_p^2 \mid p \in P^2\}$, where for every $p = (p_1, p_2) \in P^2$, the distribution D_p^2 is defined as follows. To sample from D_p^2, sample $f_1 \sim D_{p_1}$, $f_2 \sim D_{p_2}$ independently, and return $\Phi(f_1, f_2)$. That $q_2 \leq q^2$ is straightforward, and that the new rate $\rho_2 \geq 1 - (1 - \rho)^2$ can be shown using the bilinearity of Φ (see Claim 30). As for the smoothness, we prove (see Lemma 32) that for every $p, r \in P^2$,

$$\mathbf{Pr}\left[\Phi(f_1, f_2)(r) \neq 0\right] \leq \tau^{\Delta(p,r)}, \tag{2.2}$$

where $\Delta(p, r)$ is the non-relative Hamming distance between p and r. In particular, for $r \neq p$, we get the bound $\tau_2 \leq \tau$.

Note that as the world is now squared, a bound on the smoothness of merely τ is poor. However, by Equation (2.2), for most points $r \in P^2$ we in fact have a better bound of τ^2. It is only those points of distance one from p that cause the smoothness from "squaring" and, as a result, deteriorate the distance of the induced LCC (recall Claim 10). A natural approach would be to "zero out" the problematic points. To make "zero out" formal, for a set $S \subseteq P^2$, let $\nu_S : P^2 \to \mathbb{F}$ be such that $\nu_S(r) = 0$ if $r \in S$ and $\nu_S(r) = 1$ otherwise. Now, instead of $\Phi(f_1, f_2)$ consider the function $\widehat{\Phi}(f_1, f_2) = \Phi(f_1, f_2) \cdot \nu_L$ where

$$L = \{r \in P^2 \mid \Delta(p, r) = 1 \text{ and } \Phi(f_1, f_2)(r) \neq 0\}.$$

By construction, Equation (2.2) implies that the smoothness of dual SLR defined using $\widehat{\Phi}$ is bounded by τ^2. Unfortunately, however, we can no longer guarantee anything about the rate ρ_2 which, recall, is the parameter we set out to improve.

Our key idea is to construct carefully chosen functions in addition to those from $S^2 = \cup_p \mathsf{supp}(D_p^2)$ which allows us to zero out the problematic points while deteriorating the rate only slightly. To describe our solution, let R be a partition of P^2, where each part has size $c+1$ for some parameter c to be chosen later on. We denote the part, or class, in R containing an element $p \in P^2$ by $[p]$ and write $(p) = [p] \setminus \{p\}$ for the *open class* of p. For each $p \in P^2$ define the function $f_p : P^2 \to \mathbb{F}$ by $f_p(r) = 1$ if $r \in [p]$ and $f_p(r) = 0$ otherwise. We adjoin all $\frac{|P|^2}{c+1}$ functions $\mathcal{L}_R = \{f_p \mid p \in P^2\}$ to S^2 by considering $\mathcal{L}_R^2 = \mathrm{Span}(S^2) + \mathrm{Span}(\mathcal{L}_R)$. That is, our dual-induced SLR is redefined to be $(\mathcal{L}_R^2)^\perp$ rather than $(S^2)^\perp$. This has some cost in rate, but a manageable one. Indeed, note that $\dim(\mathcal{L}_R^2) \leq (1 - \rho_2 + \frac{1}{c+1})|P^2|$. Thus, for sufficiently large c, the rate loss incurred by adding the functions in $\mathcal{L}_R$ can be made small. The advantage we get by adjoining these functions is that we can now zero out any point r we wish by using the points in its open class (r). Indeed, for every $f \in (\mathcal{L}_R^2)^\perp$ and $r \in P^2$ we have $f(r) = -\sum_{w \in (r)} f(w)$. Note that, on top of the $\frac{1}{c+1}$ loss in rate, we expect to pay a multiplicative c cost in query complexity as $|(r)| = c$.

To be more precise, for $p \in P^2$, we define a distribution $(D_R^2)_p$, which will avoid using the problematic points given by L above, as follows. To sample a function $f \sim (D_R^2)_p$ proceed as follows:

1. Sample $g \sim D_p^2$ and let $L = \{r \in P^2 \mid \Delta(p, r) = 1 \text{ and } g(r) \neq 0\}$.
2. For every $r \in L$ and $w \in (r)$ sample $h_{r,w} \sim D_w^2$.
3. Return

$$f = g\nu_L + \sum_{r \in L} g(r) \sum_{w \in (r)} \frac{h_{r,w}\nu_{\{w\}}}{h_{r,w}(w)}. \tag{2.3}$$

Observe that the first summand $g\nu_L$ in Equation (2.3) is the attempt we started with. However, using the partition R, instead of simply zeroing out L (which prevents us from arguing about the rate ρ_2), for every $r \in L$ that was zeroed out, we go over each of the points w in its open class and add a carefully chosen linear combination of the "freshly" sampled functions $\{h_{r,w} \sim D_w^2\}$ to $g\nu_L$ so as to guarantee that $f \in \mathcal{L}_R^2$ (see Claim 41).

There is one technical issue the reader should be aware of. It might not be the case that $f(p) \neq 0$, which is the basic requirement of dual SLR. Indeed, while $g(p) \neq 0$ it might be the case $h_{r,w}(p) \neq 0$ for one or more pairs (r, w) as well. As a result, a cancellation may occur, causing $f(p) = 0$. This is where we make use of the $\perp$ symbol in the formal definition of dual SLR. Before outputting f, we check that this cancellation has not occurred and otherwise return $\perp$.

2.2.1 Axis evasive partitions

The above scheme can be implemented with any partition R. However, not every partition will enable us to improve the smoothness. Informally, we would like the partition to have the property that the union of open classes taken over the set of points of distance one from a given point p, is composed of points that are mostly of distance two from one another. To make this precise, we note that the set of points of distance one from a given point p is contained in the union of a horizontal and a vertical line. We refer to such lines, collectively, as axis-parallel lines. The following definition abstracts what we need from the partition so to argue about the smoothness.

▶ **Definition 13.** *Let P be a set. A partition R of P^2 is said to be (c, s)*-axis evasive *if*

1. *For every $p \in P^2$, $|(p)| = c$.*
2. *For every pair of axis-parallel lines ℓ, ℓ' (possibly equal),*

$$|\ell \cap \bigcup_{p \in \ell'} (p)| \leq s.$$

3. *For every $p \in P^2$ and every axis-parallel line ℓ, $|[p] \cap \ell| \leq 1$.*

We show that by using a (c, s)-axis evasive partition, the dual SLR defined above has smoothness $\tau_2 = O(csq\tau^2)$ (see Claim 43). The reader should think of c, s as constants (or slightly sub-constants) and $q \ll \tau^{-1}$, and so $\tau_2 \approx \tau^2 \ll \tau$.

2.2.2 Constructing axis-evasive partitions

Assume $|P| = m$ is an odd prime power, and let c be an even integer such that $c + 1 \mid m + 1$. Under these assumptions, we are able to give an explicit algebraic construction of (c, s)-axis evasive partitions of P^2 where $s = O(c^2)$ (see Section 5.2). Intuitively, as we want to construct a partition that "breaks" axis-parallel-ness, rotation would be a natural approach. Indeed, for our construction, we identify P with the finite field $\mathbb{F}_m$ and P^2 with $\mathbb{F}_{m^2}$. For every choice of $\alpha \in \mathbb{F}_{m^2} \setminus \mathbb{F}_m$, one can identify $\mathbb{F}_{m^2}$ with $\mathbb{F}_m + \alpha\mathbb{F}_m$. So, informally, $\mathbb{F}_m$ and $\alpha\mathbb{F}_m$ are the horizontal and vertical axes, respectively. To formalize the intuition of rotation, we take an element β of order $c + 1$ in the multiplicative group of $\mathbb{F}_{m^2}$. Being a cyclic group, and since $c+1 \mid m+1 \mid m^2 - 1$, such an element exists. Multiplication by β can, informally, be thought of as a rotation by a $\frac{1}{c+1}$ angle. We take the partition of $\mathbb{F}_{m^2} \setminus \{0\}$ according to the cosets of $\langle\beta\rangle$ - the subgroup generated by β (and do not worry much about the origin). We show that, with this construction, properties (1) and (2) of Definition 13 are satisfied. Property (3), however, does not and so we need to make a certain modification of the construction to resolve this. We do not delve into the required alternation of the construction here.

2.2.3 Rate amplification for dimension higher than two

Our basic rate amplification procedure can be easily generalized to any $\ell > 2$. On the other hand, our distance-efficient rate amplification procedure is designed for $\ell = 2$. To go from $\ell = 2$ to higher powers, we more or less do the obvious thing, namely, apply the dual SLR construction iteratively, where in each iteration we square the size of the previously obtained set. The only technical issue is that the divisibility by $c + 1$ requirement is not maintained throughout the process. Indeed, 2 is the only nontrivial common factor of $m+1$ and m^2+1. To overcome this, we truncate the resulted set, slightly reducing its size from m^2 to a prime m' that is divisible by $c+1$. The truncation deteriorates the rate and so we would like $m' \approx m^2$. Such prime m' is guaranteed to exist by the Siegel–Walfisz Theorem [34, 37] that refines Dirichlet's theorem on primes in arithmetic progressions.

2.3 Query-efficient distance amplification

The AEL distance amplification procedure was originally based on expander graphs [2, 1]. Kopparty et al. [26] used samplers instead - a point of view that we find fruitful for our needs. Informally, an (ε, δ)-sampler is a bipartite graph on vertex set $L \cup R$ with the following property. For every $T \subseteq R$, having density $\mu(T)$, all but δ-fraction of the left vertices have $\mu(T) \pm \varepsilon$ fraction of their neighbours in T (see Definition 14). For simplicity, we assume regularity with left-degree d and right degree D.

Given a code with poor distance δ, AEL amplifies the distance to constant using an (ε, δ)-sampler where, for the reduction, ε is taken to be constant. The AEL procedure has a Dd multiplicative cost in query complexity. Prior works used either expander graphs or "balanced" samplers, namely, samplers with $|L| = |R|$ and $D = d$. With this choice, the lowest possible degree is $d = \Theta(1/(\varepsilon^2\delta))$, which in turn yields a $\Theta((1/\delta)^2)$ multiplicative cost in query complexity.

Our improved distance amplification procedure is based on two simple ideas. Our variant has a lower cost in query complexity: Instead of a Dd factor, our variant has roughly $q_D q_d$ multiplicative cost where, recall, q_k is the query complexity of an asymptotically good LDC on message length k. Our variant also makes use of samplers, and when instantiated with a balanced sampler, the cost is roughly $q_d^2 = q_{1/\delta}^2$. Our second idea allows us to essentially get rid of the square (which is crucial for obtaining our corollaries). It is known that by working with unbalanced samplers, in which $|L| \gg |R|$, one can obtain (ε, δ)-samplers with a much lower left-degree $d = O(\log(1/\delta)/\varepsilon^2)$. We note that, for the original AEL procedure, working with unbalanced samplers cannot yield a significant improvement. Indeed, to achieve this saving in left-degree, the ratio $|L|/|R| = \Omega(1/(\delta \log(1/\delta)))$ which in turn implies $D = |L|d/|R| = \Omega(1/\delta)$. This then only gives a quadratic improvement over AEL. When instantiated with our variant, unbalanced samplers yield query complexity roughly $q_{1/\delta} q_{\log(1/\delta)}$.

3 Preliminaries

Notations and conventions

Unless otherwise stated, all logarithms are taken to the base 2. We denote by $\mathbb{N}$ the set of natural numbers (of course, including 0). For an integer $c \geq 1$, we let $[c] = \{1, 2, \ldots, c\}$. For ease of readability, we avoid the use of floor and ceiling. This does not affect the stated results. For two strings x, y of equal length over a common alphabet, we denote by $\mathsf{dist}(x, y)$ their relative hamming distance, namely, the fraction of indices on which they disagree. Let $A \neq \emptyset$ be an ambient (finite) set. For $B \subseteq A$, we denote by $\mu(B)$ the density of B in A, namely, $\mu(B) = |B|/|A|$.

Let $G = (V, E)$ be an undirected graph with maximal degree D. Assume that the neighbours of every node $v \in V$ are labeled by distinct numbers from $1, \ldots, \deg(v)$. We define the neighbourhood function $\Gamma_G : V \times [D] \to (V \times [D]) \cup \{\perp\}$ as follows. For $v \in V$ and $i \in [\deg(v)]$ we let $\Gamma_G(v, i) = (u, j)$ where u is the i'th neighbour of v and v is the j'th neighbour of u. For $i \in [D] \setminus [\deg(v)]$ the function is defined to be $\perp$ (though this is only for the sake of formality. We will never use such input i). If G is clear from context we sometimes omit it from the subscript. When interested only on the node u as above and not on j, we make a slight abuse of notation and write $\Gamma(v,i)$ when referring to u. Last, we write $\Gamma(v)$ for the set of all neighbours of v.

3.1 Samplers

Our distance amplification procedure makes use of samplers. These are bipartite graphs with a certain pseudo-random property. Let $G = (L, R, E)$ be a bipartite graph. We say G is *left-regular* if all nodes in L have the same degree.

▶ **Definition 14** ([9]). *Let $0 < \varepsilon, \delta < 1$. A bipartite graph $G = (L, R, E)$ is an (ε, δ)-*sampler *if for every subset $T \subseteq R$, for all but δ-fraction of vertices $v \in L$ it holds that*

$$\left| \frac{|\Gamma(v) \cap T|}{|\Gamma(v)|} - \mu(T) \right| \leq \varepsilon.$$

We will be working with “unbalanced” samplers. These are samplers with $|L| \gg |R|$. The state-of-the-art constructions of these samplers rely on their connection to randomness seeded extractors. We refer the interested reader to the excellent survey by Goldreich [17] for more information. When working with samplers, it is rather typical that the bipartite graph is left-regular, that is, the degree of all vertices in L is the same. A small additional technical property we need is that the degree of every vertex in R is close to the average right-degree. We make use of the following theorem which gives (non-explicit) samplers with near-optimal parameters having the above properties with respect to the degrees. We give a proof sketch for completeness.

▶ **Theorem 15.** *There exists a universal constant $c_{\mathsf{samp}} \geq 1$ such that the following holds. For all integers ℓ, r and all $\varepsilon > 0$, $1/2 > \delta > 0$ for which $\ell \geq \frac{r}{\delta \log(1/\delta)}$, there exists a left-regular (ε, δ)-sampler $G = ([\ell], [r], E)$ with left-degree $d = c_{\mathsf{samp}} \cdot \log(1/\delta)/\varepsilon^2$. Moreover, provided that $\log r < 1/(\delta\varepsilon^2)$, every right vertex has degree in $[D/2, 2D]$ where $D = \ell d/r$ is the average right degree.*

For the proof we need the following well-known lemma.

▶ **Lemma 16.** *For every integers $1 \leq k \leq n$ with $\frac{k}{n} = \delta \leq \frac{1}{2}$ it holds that*

$$\sum_{i=0}^{k} \binom{n}{i} \leq 2^{H(\delta)n},$$

where $H(x) = -x\log(x) - (1-x)\log(1-x)$ is the binary entropy function.

Proof sketch for Theorem 15. The proof is via the probabilistic method, where for every left vertex we choose d neighbours independently and uniformly at random, and independently across all left vertices (note that in the above we allow for parallel edges, but if that troubles the reader, that can be avoided as well in the regime of interest $d \ll r$ by arguing that the probability of a right neighbor to be selected more than once is small. In any case, our distance amplification procedure works just as well with parallel edges). Fix $T \subseteq [r]$. For $v \in [\ell]$ let F_v be the indicator random variables that is 1 if and only if $||\Gamma(v) \cap T|/d - \mu(T)| > \varepsilon$. By the Chernoff bound, $\mathbf{Pr}[F_v] \leq e^{-\Omega(\varepsilon^2 d)}$. Fix $S \subseteq [\ell]$ with $|S| = \delta\ell$. The probability that for all vertices $v \in S$ it holds that $F_v = 1$ is bounded above by $e^{-\Omega(\varepsilon^2 d \cdot \delta\ell)}$. By taking the union bound over all S and T, we get that except with probability

$$2^r \binom{\ell}{\delta\ell} e^{-\Omega(\varepsilon^2 d\delta\ell)} \leq 2^{r + H(\delta)\ell - c\varepsilon^2 d\delta\ell}, \tag{3.1}$$

the sampled graph is an (ε, δ)-sampler. Note that the last inequality follows by Lemma 16, where $c > 0$ is some constant. By taking $c_{\mathsf{samp}} \geq 5/c$, one can verify (using that $H(x) \leq 2x \log(1/x)$ for all $x \leq 1/2$) that the right hand side in Equation (3.1) is bounded by 1/4.

As for the moreover part, again, by the Chernoff bound, the probability that there exists a right vertex which has degree outside $[D/2, 2D]$ is bounded above by $re^{-\Omega(\ell d/r)}$, and this is bounded by 1/4 by our choice of parameters and by taking c_{samp} large enough. ◀

We now turn to state the parameters of the explicit construction of samplers that we use.

▶ **Theorem 17** ([31, 17]).[5] *For every constant $\Delta > 0$ there exists a constant $c = c(\Delta) \geq 1$ such that the following holds. For all $\varepsilon > 0$, $\delta > 0$* [6]*, there exists an explicit left-regular (ε, δ)-sampler $G = ([\ell], [r], E)$. The left-degree of G is $d = ((1/\varepsilon)\log(1/\delta))^c$. Furthermore, the average right degree $D = \ell d/r$ of G is in $[D', 2D']$ where*

$$D'(\Delta, \varepsilon, \delta) = \frac{d}{2} \cdot \left(\frac{2}{\delta}\right)^{\Delta+1}. \tag{3.2}$$

3.2 Codes

We make use of “standard” error-correcting codes. In this section we gather some known results we use.

▶ **Theorem 18** (The Gilbert-Varshamov bound). *Let Σ be a set of size $|\Sigma| = q$. For every $n \in \mathbb{N}$, and $0 \leq \delta \leq 1 - \frac{1}{q}$ there exists a code of block-length n over Σ, with distance at least δ and rate $r \geq 1 - H_q(\delta)$. Furthermore, if q is a prime power and $\Sigma = \mathbb{F}_q$, there exists a linear code over Σ with rate $r \geq 1 - H_q(\delta) - g(n)$, where $g(n) = O(\frac{1}{n})$.*

▶ **Lemma 19.** *There exists a constant $\beta_0 > 0$ such that the following holds. Let n be an integer and $\frac{1}{\log n} < \beta < \beta_0$. Let $\Sigma = \mathbb{F}_q$ for $q \geq 2$ a prime power. Then, there exists an explicit linear code of block-length n over Σ with rate $1 - \beta$ and relative distance β^3.*

The existence of these codes follows from a special case of the Zyablov bound [43], but for completeness we describe a construction which attains the stated parameters. For the proof, we make use of the following easy claim whose proof is omitted.

▷ Claim 20. For every $x \in (0, 1/2]$ and $q \geq 2$, $H_q(x) \leq x \log_q(\frac{q^3}{x})$.

Proof of Lemma 19. The proof is obtained by taking the code concatenation of two codes, a Reed-Solomon code and a Gilbert-Varshamov code. Let p be the least prime such that $p \geq n$. Recall that $p \leq 2n$. Set C_{RS} to be the Reed-Solomon code over $\mathbb{F}_p$ of block length $n_{\mathrm{RS}} = \frac{(1-\beta^{1.1})n}{\log_q p}$ and message length $k_{\mathrm{RS}} = (1 - \beta^{1.1})n_{\mathrm{RS}}$. So, C_{RS} has rate $1 - \beta^{1.1}$ and relative distance at least $\beta^{1.1}$. Now take C_{GV} to be a linear code of the following parameters. The message length is $k_{\mathrm{GV}} = \log_q p$, the block length is $\frac{1}{1-\beta^{1.1}} k_{\mathrm{GV}}$ (and therefore the rate is $1 - \beta^{1.1}$), and the relative distance is at least $\beta^{1.4}$. We wish to invoke Theorem 18 so as to prove that such a code exists. To this end, we must verify that $1 - H_q(\beta^{1.4}) - g(n) \geq 1 - \beta^{1.1}$. Indeed, by Claim 20, we have that

[5] The sampler in [31] has a mild requirement on ε which we state the theorem without, as it is explained in [17] how this requirement can be relaxed, by using a more recent extractor.

[6] The sampler in [31] has a number of edges z that is a power of two. We state the theorem for a general z as one can take the subgraph of only part of the left vertices, and get a sampler in which δ is at most doubled.

$$1 - H_q(\beta^{1.4}) - g(n) \geq 1 - \beta^{1.4} \log_q \left(\frac{q^3}{\beta^{1.4}} \right) - g(n) \geq 1 - \beta^{1.1},$$

where the last inequality holds for all sufficiently small $\beta \geq 0$, and since $g(n) = O(\frac{1}{n})$ and $\beta \geq \frac{1}{\log n}$, by assumption.

Note that C_{GV} is not explicit as Theorem 18 only guarantees existence of a code with the stated parameters. However, as the block-length of C_{GV} is $O(\log n)$, such a code can be found by an exhaustive search on generating matrices, in time $2^{O((\log n)^2)}$. To improve on that, we remark that the code C_{GV} can also be found by going only over a limited family of generating matrices (see [21]), and this can be done in time $\mathrm{poly}(n)$.

Consider the concatenated code $C_{\mathrm{RS}} \circ C_{\mathrm{GV}}$. It has block length $n_{\mathrm{RS}} \cdot n_{\mathrm{GV}} = n$, rate $(1-\beta^{1.1})^2$ which is at least $1-\beta$ for all small enough $\beta > 0$, and relative distance $\beta^{1.2}\beta^{1.4} \geq \beta^3$, completing the proof. ◀

4 Rate amplification for dual-induced SLR

In this section we introduce the notion of *smooth locally recoverable sets (SLR)* which under non-adaptive and linearity assumptions is shown to be equivalent to LCC. We consider a certain class of SLR, to which we call *dual-induced SLR*. These are SLR that are obtained by the dual of certain structured sets. The structure of these dual-SLR sets allows us to devise a rate amplification procedures for them. Informally, dual-SLR are sets of tuples (or linear spaces of vectors if the alphabet over which we are working is a field) in which every given entry of a tuple in the set can be recovered using only few queries *and* in a "smooth" manner, which is to say that the distribution of every query has high min-entropy.

▶ **Definition 21** (Smooth locally recoverable sets (SLR)). *Let Σ, P be arbitrary non-empty sets. We say that $C \subseteq \Sigma^P$ is (q, τ, ε)-*smooth locally recoverable *(SLR for short) if there exists a randomized algorithm* Rec*, called a* recovering procedure*, that is given as input $p \in P$ and an oracle access to $c \in C$. The recovering procedure outputs either an element of Σ or a symbol $\perp$ which is assumed not to be in Σ. The algorithm* Rec *has the following properties:*

- *For every $(c, p) \in C \times P$,* $\mathsf{Rec}^c(p)$ *makes at most q queries to c.*
- *For every $c \in C$ and $p, r \in P$ it holds that*

$$\mathbf{Pr}[\mathsf{Rec}^c(p) \text{ queries } c_r] \leq \tau.$$

- *For every $(c, p) \in C \times P$, the random variable* $\mathsf{Rec}^c(p) \in \{c_p, \perp\}$*, and*

$$\mathbf{Pr}\left[\mathsf{Rec}^c(p) = \perp\right] \leq \varepsilon.$$

We assume that for every $p \in P$ whether $\mathsf{Rec}^c(p) = \perp$ is independent of c, and that it is never the case that $\mathsf{Rec}^c(p)$ queries c_p. When Σ is a field and C is a linear subspace of Σ^P, we say that C is linear*. In this case, the* rate *of C is defined as $\dim(C)/|P|$. We will mostly consider* non-adaptive SLR*. These are SLR in which the joint distribution of queries is independent of c.*

We remark that the notion of SLR is very similar to the notion of smooth-codes of [23] for LDC. We now have the following easy claim showing that SLR yield LCC and, assuming linearity, LDC.

▷ Claim 22. Let $C \subseteq \Sigma^P$ be a (q, τ, ε)-SLR. Then, for every $\varepsilon' > 0$, C is a $(q, \delta, \varepsilon + \varepsilon')$-LCC with $\delta = \varepsilon'/(q\tau|P|)$. As a consequence, if C is also linear then C is a $(q, \delta, \varepsilon + \varepsilon')$-LDC.

Proof. To show that C is an LCC, we devise a local corrector for C. Given an oracle access to $c \in \Sigma^P$, and $p \in P$ as input, the local corrector computes $z = \mathsf{Rec}^c(p)$. If $z =\perp$ then the local corrector returns some arbitrary element of Σ, and otherwise return z. To analyze this local corrector, let $c' \in \Sigma^P$ be such that $\mathsf{dist}(c, c') \leq \delta|P|$. Denote $B = \{p \in P \mid c_p \neq c'_p\}$. Note that conditioned on $\mathsf{Rec}^c(p) \neq\perp$, the local corrector returns c_p successfully if all q queries do not fall into B. The probability that any given query falls into B is bounded above by $\tau|B|$ and so, by the union bound, the probability that some query falls into B is bounded above by $\tau|B|q \leq \varepsilon'$. This proves that C is a $(q, \delta, \varepsilon + \varepsilon')$-LCC. Note that linear LCC are systematic and so every linear LCC induces an LDC. ◁

In fact, for linear LCCs, the other direction also holds, meaning that such an LCC is an SLR, as we have in the following claim.

▷ Claim 23. Let $C \subseteq \mathbb{F}^P$ be a linear non-adaptive (q, δ, ε)-LCC where 1-$\varepsilon > 1/|\mathbb{F}|$. Then, C is a (q, τ, ε')-SLR with $\tau = q/(\delta|P|)$ and $\varepsilon' = 0$.

Before we prove the claim, we need to state the following easy to verify fact.

► **Fact 24.** *Let $L \subseteq \mathbb{F}^P$ be a linear subspace, let $p \in P$, let $Q \subseteq P$ and let $x \in \mathbb{F}^{|Q|}$. Then, one of the following cases must hold.*

1. *There is at most one $\alpha \in \mathbb{F}$ for which there exists some $v \in L$ satisfying $v(Q) = x$ [7] and $v(p) = \alpha$;*
2. *For every $\alpha \in \mathbb{F}$ there is an equal number of $v \in L$ for which $v(Q) = x$ and $v(p) = \alpha$.*

In particular, either no function (even randomized) of $v(Q)$ can predict $v(p)$ with probability more than $1/|\mathbb{F}|$, when $v \in L$ is randomly chosen uniformly, or $v(Q)$ always determines $v(p)$.

With that, we now prove Claim 23 [8].

Proof for Claim 23. To show that C is an SLR, we devise a recovering procedure Rec for it, based on the local corrector promised by it being an LCC. Let D [9] be such a local corrector. For every point $p \in P$, we construct a sequence of disjoint sets $Q_1^p, \ldots, Q_{m_p}^p \subseteq P$, where for every i, $c(Q_i^p)$ determines $c(p)$ while satisfying $|Q_i^p| \leq q$, and $m_p \geq \delta|P|/q$. On $p \in P$ and oracle access to $c \in C$, the procedure $\mathsf{Rec}^c(p)$ acts by uniformly choosing $i \in [m_p]$, querying $c(Q_i^p)$, and using it to deduce and output $c(p)$. The correctness of the result of Rec is immediate (since Rec always succeeds, $\varepsilon' = 0$), and indeed the number of queries is no more than q. Since the sets are disjoint, the probability that a point is queried is no more than $\tau = q/(\delta|P|)$. It only remains to show how the assumed sets can be constructed, to conclude that C is a (q, τ, ε')-SLR, which we now turn to do.

For every $p \in P$, $Q_1^p, \ldots, Q_{m_p}^p$ are constructed as follows. Set $Q_0^p = \emptyset$. For $i = 1, 2, \ldots$, set $S_i = Q_0^p \cup \cdots \cup Q_{i-1}^p$. If $|S_i| > \delta|P|$, halt and set $m_p = i - 1$. Otherwise, it holds that for every $c \in C$, for every modification of the coordinates in S_i to some erroneous values, the decoder D correctly outputs $c(p)$ with probability at least $1 - \varepsilon$. An equivalent description of this case is the following: for every $c \in C$ and $z : S_i \to \mathbb{F}$, define $c_z \in \mathbb{F}^P$ such that for $r \notin S_i$, $c_z(r) = c(r)$, and for $r \in S_i$, $c_z(r) = z(r)$; the decoder D chooses a set of queries $Q \subseteq P$, $|Q| \leq q$, according to a distribution and applies a function f_Q on $c_z(Q)$; with probability at least $1 - \varepsilon$, $f_Q(c_z(Q)) = c(p)$. Since Q is chosen in a manner independent of c and z, one can

[7] For a set $A = \{a_1, \ldots, a_{|A|}\}$, $v(A)$ denotes the sequence $(v(a_1), \ldots, v(a_{|A|}))$.

[8] This proof is inspired by the proof of [23] of their Theorem 1 and by a proof in [41] for a different claim.

[9] We make the slight assumption that $D^c(p)$ never directly queries $c(p)$. If however it does, then similarly C can be shown to be a (q, τ, ε')-SLR for $\tau = \frac{1}{(\delta|P|/q)-1}$.

verify that this implies that there exists some fixed Q for which when $c \in C$ and $z : \mathbb{F} \to S_i$ are chosen randomly in a uniform manner, with probability at least $1-\varepsilon$ (this time over the choice of c and z), $f_Q(c_z(Q)) = c(p)$. Therefore, we can define another function f'_Q that only gets $c(Q \setminus S_i)$, chooses z at random, and outputs $f_Q(c_z(Q))$. If $c \in C$ is chosen uniformly at random, $f'_Q(Q \setminus S_i) = c(p)$ with probability at least $1-\varepsilon > 1/|F|$. By Fact 24, this implies that $c(Q \setminus S_i)$ determines $c(p)$, for every $c \in C$. We therefore set $Q^p_i = Q \setminus S_i$, and proceed to next i. As this process only halts when $|S_i| > \delta|P|$, and for every i $|S_i| \le q(i-1)$, we have that indeed $m_p \ge \delta|P|/q$. Further note that by the choice of each Q^p_i, the sets $Q^p_1, \dots, Q^p_{m_p}$ are disjoint, as required. ◁

4.1 Dual SLR and their induced SLR

Our construction of SLR sets will be via constructing and analyzing sets which we call *dual SLR* sets. The SLR will then be induced from these dual SLR. We start by setting some notation. Let P be a non-empty finite set and $\mathbb{F}$ a finite field. We make use of the standard notation $\mathbb{F}^P$ to denote the set of all functions $\{f : P \to \mathbb{F}\}$. Note that $\mathbb{F}^P$ has a natural $\mathbb{F}$-vector space structure where addition is point-wise, namely, for every $f, g \in \mathbb{F}^P$ and $a, b \in \mathbb{F}$ we have that $af + bg \in \mathbb{F}^P$ is defined by $(af + bg)(p) = af(p) + bg(p)$ for all $p \in P$. We consider the natural inner product map $\langle \cdot, \cdot \rangle : \mathbb{F}^P \times \mathbb{F}^P \to \mathbb{F}$ that is defined, for $f, g \in \mathbb{F}^P$, by $\langle f, g \rangle = \sum_{p \in P} f(p)g(p)$. Given $f \in \mathbb{F}^P$, we let $f^\perp = \{g \in \mathbb{F}^P \mid \langle f, g \rangle = 0\}$. Note that $f^\perp$ is a linear subspace of $\mathbb{F}^P$. More generally, given a set $S \subseteq \mathbb{F}^P$ we define the linear subspace $S^\perp = \bigcap_{f \in S} f^\perp$. For $f \in \mathbb{F}^P$ we denote $|f| = |P \setminus f^{-1}(0)|$.

For the sake of readability, the field $\mathbb{F}$ and the set P will be omitted from the notation that we are about the introduce in this section. Both will be clear from context. For $p \in P$ define $\mathcal{F}_p = \{f \in \mathbb{F}^P \mid f(p) \neq 0\}$. Informally, a dual SLR is a collection of distributions over $\mathbb{F}^P$, one for each point $p \in P$. The distribution D_p, that corresponds to p, outputs a function $g \in \mathcal{F}^P$. We think of g as "passing through" p. We also allow D_p to output a special "failed symbol" $\perp$ with some small probability. A dual SLR has the guarantee that g does not pass through many other points, namely, $|g|$ is bounded, and that the dimension of all functions that can be sampled, when considering all distributions D_p, $p \in P$, is also bounded. Perhaps most importantly is the requirement that for every other fixed $r \in P$, the sampled $g \sim D_p$ is likely to have the property that $g \notin \mathcal{F}_r$. Formally,

▶ **Definition 25** (Dual SLR). *Let P be a set, $\mathbb{F}$ a field. Let $\mathcal{D} = \{D_p \mid p \in P\}$ be a collection of distributions, where for each $p \in P$, $\mathsf{supp}(D_p) \subseteq \mathcal{F}_p \cup \{\perp\}$. Denote $S = \bigcup_{p \in P} \mathsf{supp}(D_p)$. Let $\mathcal{L}$ be a linear subspace of $\mathbb{F}^P$ such that $S \subseteq \mathcal{L} \cup \{\perp\}$. The pair $(\mathcal{D}, \mathcal{L})$ is said to be a $(q, \tau, \varepsilon, \rho)$-*dual SLR *on $\mathbb{F}^P$ provided the following holds:*

1. *$|g| \le q$ for all $g \in S \setminus \{\perp\}$.*
2. *For every pair of distinct $p, r \in P$ (not necessarily distinct), it holds that*

$$\Pr_{g \sim D_p}[g(r) \neq 0 \mid g \neq \perp] \le \tau.$$

3. *For every $p \in P$, $\mathbf{Pr}[D_p = \perp] \le \varepsilon$.*
4. *$\dim(\mathcal{L}) \le (1-\rho)|P|$.*

The linear subspace $\mathcal{L}^\perp$ of $\mathbb{F}^P$ is called the induced SLR *from the dual SLR $(\mathcal{D}, \mathcal{L})$. The parameter τ of a dual-SLR is referred to as its* smoothness.

Let $(\mathcal{D}, \mathcal{L})$ be a dual SLR. We turn to show that, as the name suggests, the induced SLR $\mathcal{L}^\perp$ is indeed an SLR.

▶ **Lemma 26.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}, \mathcal{L})$ be $(q, \tau, \varepsilon, \rho)$-dual SLR on $\mathbb{F}^P$. Then the induced SLR $\mathcal{L}^\perp$ is a $(q-1, \tau, \varepsilon)$-SLR. Furthermore, $\mathcal{L}^\perp$ is linear and has rate ρ or larger.*

Proof. The moreover part readily follows since $\mathcal{L}^\perp$ is a linear subspace of $\mathbb{F}^P$ and since

$$\dim(\mathcal{L}^\perp) = |P| - \dim(\mathcal{L}) \geq \rho |P|.$$

We describe a recovering procedure for $\mathcal{L}^\perp$, namely, a randomized algorithm that is given an oracle access to $f \in \mathcal{L}^\perp$ as well as a point $p \in P$ as input. The recovering procedure proceeds as follows:

1. Sample $g \sim D_p$. If $g =\perp$ return $\perp$; Otherwise,
2. Query f on all points $Q = \{r \in P \setminus \{p\} \mid g(r) \neq 0\}$.
3. Return
$$-\frac{1}{g(p)} \sum_{r \in Q} g(r) f(r).$$

The query complexity of Rec is bounded by $q-1$ as $|Q| = |g| - 1 \leq q-1$. The probability that $\perp$ is returned is at most ε by construction. We turn to prove that $\mathsf{Rec}^f(p) \in \{f(p), \perp\}$. By construction, $\mathsf{Rec}^f(p) =\perp$ if and only if $g =\perp$. Assume than that $g \neq\perp$, hence, $g \in \mathsf{supp}(D_p) \subseteq \mathcal{L}$. As $f \in \mathcal{L}^\perp$ we have that $0 = \langle f, g \rangle$, and so

$$0 = \sum_{r \in P} g(r) f(r) = g(p) f(p) + \sum_{r \in Q} g(r) f(r).$$

As $g \in \mathsf{supp}(D_p) \subseteq \mathcal{F}_p$ we have $g(p) \neq 0$, and so

$$f(p) = -\frac{1}{g(p)} \sum_{r \in Q} g(r) f(r) = \mathsf{Rec}^f(p).$$

To conclude the proof, we turn to analyze the smoothness of Rec. First, note that, by construction, f is never queried on p itself. Consider then any $r \in P \setminus \{p\}$. Conditioned on $g \neq\perp$, the function f is queried on r if and only if $g(r) \neq 0$. Thus,

$$\mathbf{Pr}\left[f(r) \text{ is queried}\right] = \mathop{\mathbf{Pr}}_{g \sim D_p}\left[g(r) \neq 0 \mid g \neq\perp\right] \leq \tau,$$

and the proof follows. ◀

We now show that the opposite holds as well, that any linear, non-adaptive, SLR induces a dual-SLR.

▶ **Lemma 27.** *Let P be a set, $\mathbb{F}$ a field, and let $C \subseteq \mathbb{F}^P$ be a linear non-adaptive (q, τ, ε)-SLR with rate ρ. Then for some set $\mathcal{D}$, $(\mathcal{D}, C^\perp)$ is a $(q+1, \tau, \varepsilon, \rho)$-dual SLR*

Proof. Let Rec be a recovering procedure promised by C being a SLR. Assume that Rec uses R random bits. For every point $p \in P$, denote by $\mathcal{E}_p$ the set of choices of the random bits $r \in \{0,1\}^R$ for which $\mathsf{Rec}^c(p) =\perp$ (for any $c \in C$). Note that $\mu(\mathcal{E}_p \subseteq \{0,1\}^R) \leq \varepsilon$.

For any $p \in P$ and $r \in \{0,1\}^R$, $r \notin \mathcal{E}_p$, denote by $Q_{p,r} \subseteq P$ the set of query locations which $\mathsf{Rec}(p)$ makes when r is the choice of randomness. Define a function $f_{p,r} : C \to \mathbb{F}$ such that $f(c)$ is the output of $\mathsf{Rec}^c(p)$ when fixing its randomness to r and note that $f_{p,r}(c)$ only depends on $\{c(w) \mid w \in Q_{p,r}\}$, and that $f_{p,r}(c) = c(p)$. Since C is linear, one can easily verify that $f_{p,r}$ is a linear map. Therefore, for some $u_{p,r} \in \mathbb{F}^P$, $f_{p,r}(c) = \langle u_{p,r}, c \rangle$ for every $c \in C$, where $u_{p,r}(w) = 0$ if $w \notin Q_{p,r}$. We have that $c(p) = \langle u_{p,r}, c \rangle$ for every $c \in C$. If we define a function $g_{p,r} \in \mathbb{F}^P$ such that

$$g_{p,r}(w) = \begin{cases} -1, & w = p; \\ u_{p,r}(q), & w \neq p, \end{cases}$$

it follows that $g_{p,r} \in C^\perp$. Note that $|g_{\{p,r\}}| \leq q+1$.

For every $p \in P$, define D_p to be the following distribution. To sample from D_p, draw $r \in \{0,1\}^R$ uniformly at random. If $r \in \mathcal{E}_p$, output $\perp$; otherwise, output $g_{p,r}$. Set $\mathcal{D} = \{D_p \mid p \in P\}$ and $\mathcal{L} = C^{\perp}$. It follows trivially by the definitions that $(\mathcal{D}, \mathcal{L})$ is a $(q+1, \tau, \varepsilon, \rho)$-dual SLR. ◀

4.2 Rate amplification for dual-induced SLR

In this section we describe our first rate amplification procedure for SLR that are induced by dual SLR. Unlike the previous section, it will be more convenient to explicitly state within the notation the set P over which we are working as we will be dealing with several such sets. The field $\mathbb{F}$, however, remains suppressed from the notation as it remains fixed in all SLR under consideration. We start by defining the following map of functions.

▶ **Definition 28.** *Let P be a set and $\mathbb{F}$ a field. For an integer $\ell \geq 1$ we define the map $\Phi\colon (\mathbb{F}^P)^{\ell} \to \mathbb{F}^{P^{\ell}}$ as follows. Let $g_1, \dots, g_\ell \in \mathbb{F}^P$. The function $\Phi(g_1, \dots, g_\ell) : P^{\ell} \to \mathbb{F}$ is defined by*

$$\Phi(g_1, \dots, g_\ell)(p_1, \dots, p_\ell) = \prod_{i=1}^{\ell} g_i(p_i)$$

for every $(p_1, \dots, p_\ell) \in P^{\ell}$.

Observe that Φ is multi-linear. Further, when $\ell = 2$ and g_1, g_2 are viewed as vectors rather than functions, Φ is the outer product of the vectors.

▶ **Definition 29.** *Let P be a set, $\mathbb{F}$ a field. Let $\mathcal{L}^P$ be a linear subspace of $\mathbb{F}^P$. For an integer $\ell \geq 1$, we define*

$$\mathcal{L}^{P^{\ell}} = \operatorname{Span}\{\Phi(g_1, \dots, g_\ell) \mid g_1, \dots, g_\ell \in \mathcal{L}^P\}.$$

▷ Claim 30. With the notation of Definition 29,

$$\dim\left(\mathcal{L}^{P^{\ell}}\right) \leq \left(\dim\left(\mathcal{L}^P\right)\right)^{\ell}.$$

Proof. Let $B = \{g_1, \dots, g_b\}$ be a basis for $\mathcal{L}^P$, where $b = \dim(\mathcal{L}^P)$. Define

$$B' = \{\Phi(h_1, \dots, h_\ell) \mid (h_1, \dots, h_\ell) \in B^{\ell}\}.$$

Observe that to prove the claim, it suffices to show that for every $f_1, \dots, f_\ell \in \mathcal{L}^P$ it holds that $\Phi(f_1, \dots, f_\ell) \in \operatorname{Span}(B')$. As $f_1, \dots, f_\ell \in \mathcal{L}^P$, for every $i \in [\ell]$ we can write $f_i = \sum_{j=1}^{b} \lambda_{i,j} g_j$ with $\lambda_{i,j} \in \mathbb{F}$. We have that

$$\begin{aligned}\Phi(f_1, \dots, f_\ell) &= \Phi\left(\sum_{j_1=1}^{b} \lambda_{1,j_1} g_{j_1}, \dots, \sum_{j_\ell=1}^{b} \lambda_{\ell,j_\ell} g_{j_\ell}\right) \\ &= \sum_{j_1 \dots, j_\ell \in [b]} \left(\prod_{t=1}^{\ell} \lambda_{t,j_t}\right) \cdot \Phi(g_{j_1}, \dots, g_{j_\ell}),\end{aligned}$$

where the last equality follows by the multi-linearity of Φ. ◁

▶ **Definition 31.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be $(q, \tau, \varepsilon, \rho)$-dual SLR. Let $\ell \geq 1$ be an integer. For $p \in P^{\ell}$ we define the distribution $D_p^{P^{\ell}}$ as follows. Write $p = (p_1, \dots, p_\ell)$. To sample an element from $D_p^{P^{\ell}}$ proceed as follows:*

1. *Sample $g_1 \sim D^P_{p_1}, \dots, g_\ell \sim D^P_{p_\ell}$ independently.*
2. *If there exists $i \in [\ell]$ such that $g_i = \perp$, return $\perp$; Otherwise*
3. *Return $\Phi(g_1, \dots, g_\ell)$.*

The collection of distributions $\{D^{P^\ell}_p \mid p \in P^\ell\}$ is denoted by $\mathcal{D}^{P^\ell}$.

We have the following lemma.

▶ **Lemma 32.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q, \tau, \varepsilon, \rho)$-dual SLR. Let $\ell \geq 1$ be an integer and $\mathcal{D}^{P^\ell}$ as in Definition 31. Then, for every $p, r \in P^\ell$,*

$$\Pr_{g \sim D^{P^\ell}_p} [g(r) \neq 0 \mid g \neq \perp] \leq \tau^{\mathsf{dist}(p,r)}.$$

Proof. Write $p = (p_1, \dots, p_\ell)$, $r = (r_1, \dots, r_\ell)$. By Definition 31, conditioned on $g \neq \perp$ we have that $g = \Phi(g_1, \dots, g_\ell)$ with $g_i \sim D^P_{p_i}$ for each $i \in [\ell]$ independently. Thus, $g(r) \neq 0$ is the event

$$\Phi(g_1, \dots, g_\ell)(r_1, \dots, r_\ell) = \prod_{i=1}^{\ell} g_i(r_i) \neq 0.$$

By the independence of $g_1, \dots, g_\ell$, and since we are working over a field $\mathbb{F}$ (and so a product is nonzero if and only if each of the terms is nonzero), we get

$$\Pr_{g \sim D^{P^\ell}_p} [g(r) \neq 0 \mid g \neq \perp] = \prod_{i=1}^{\ell} \Pr_{g_i \sim D^P_{p_i}} [g_i(r_i) \neq 0 \mid g_i \neq \perp] . \tag{4.1}$$

Let $T = \{i \in [\ell] \mid p_i \neq r_i\}$. As $\mathcal{D}^P$ is a $(q, \tau, \varepsilon, \rho)$-dual SLR, for each $i \in T$ it holds that

$$\Pr_{g_i \sim D^P_{p_i}} [g_i(r_i) \neq 0 \mid g_i \neq \perp] \leq \tau.$$

Substituting to Equation (4.1), we get

$$\Pr_{g \sim D^{P^\ell}_p} [g(r) \neq 0 \mid g \neq \perp] \leq \tau^{|T|},$$

which completes the proof. ◀

▶ **Definition 33.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q, \tau, \varepsilon, \rho)$-*dual SLR*. For an integer $\ell \geq 1$ let $\mathcal{L}^{P^\ell}$, $\mathcal{D}^{P^\ell}$ be as in Definition 29 and Definition 31, respectively. We denote the pair $(\mathcal{D}^{P^\ell}, \mathcal{L}^{P^\ell})$ by $(\mathcal{D}^P, \mathcal{L}^P)^\ell$.*

▶ **Proposition 34.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q, \tau, \varepsilon, \rho)$-*dual SLR*. Then, for every integer $\ell \geq 1$ we have that $(\mathcal{D}^P, \mathcal{L}^P)^\ell$ is a $(q_\ell, \tau_\ell, \varepsilon_\ell, \rho_\ell)$-*dual SLR*, where*

$$q_\ell \leq q^\ell,$$
$$\tau_\ell \leq \tau,$$
$$\varepsilon_\ell \leq \ell\varepsilon,$$
$$\rho_\ell \geq 1 - (1 - \rho)^\ell.$$

Proof. First note that for every $p \in P^\ell$, the distribution $D^{P^\ell}_p$ is supported on $\mathcal{F}^{P^\ell}_p \cup \{\perp\}$. Indeed, if we write $p = (p_1, \dots, p_\ell)$ then, conditioned on $g \neq \perp$, we have that $g = \Phi(g_1, \dots, g_\ell)$ where $g_i \in D^P_{p_i}$. Thus,

$$g(p) = \Phi(g_1, \dots, g_\ell)(p_1, \dots, p_\ell) = \prod_{i=1}^{\ell} g_i(p_i) \neq 0.$$

Moreover, by Definition 29,

$$\bigcup_{p\in P^\ell} \mathsf{supp}(D_p^{P^\ell}) \subseteq \mathcal{L}^{P^\ell} \cup \{\perp\}.$$

We turn to show that $q_\ell \le q^\ell$. Let $p=(p_1,\dots,p_\ell)\in P^\ell$ and consider any $g\in \mathsf{supp}(D_p^{P^\ell})$. By Definition 31, if $g\neq\perp$ then $g=\Phi(g_1,\dots,g_\ell)$ where $g_i\in\mathsf{supp}(D_{p_i}^P)\setminus\{\perp\}$. Now, for every $r=(r_1,\dots,r_\ell)\in P^\ell$ we have that

$$g(r)\neq 0 \quad\Longleftrightarrow\quad \prod_{i=1}^{\ell} g_i(r_i)\neq 0.$$

Since $\mathbb{F}$ is a field, the above is equivalent to $g_i(r_i)\neq 0$ for all $i\in[\ell]$. Hence there are at most q^ℓ points $r\in P^\ell$ for which $g(r)\neq 0$, and so $q_\ell\le q^\ell$.

The bound on the smoothness readily follows by Lemma 32. Indeed, consider any pair of distinct $p,r\in\mathbb{F}^{P^\ell}$. We have that $\mathsf{dist}(p,r)\ge 1$ and so, by Lemma 32,

$$\mathop{\mathbf{Pr}}_{g\sim D_p^{P^\ell}}[g(r)\neq 0 \mid g\neq\perp]\le \tau^{\mathsf{dist}(p,r)}\le\tau. \tag{4.2}$$

To bound the probability that $\perp$ is returned, note that the event $D^{P^\ell}=\perp$ holds only if for some $i\in[\ell]$, $g_i=\perp$. Hence, by the union bound, $\mathbf{Pr}[D_p^{P^\ell}=\perp]\le\ell\varepsilon$. We conclude the proof by bounding the dimension of $\mathcal{L}^{P^\ell}$. By assumption, $\dim(\mathcal{L}^P)\le(1-\rho)|P|$. Claim 30 then implies that

$$\dim\left(\mathcal{L}^{P^\ell}\right)\le\left(\dim\left(\mathcal{L}^P\right)\right)^\ell\le((1-\rho)|P|)^\ell=(1-\rho)^\ell|P^\ell|. \qquad \blacktriangleleft$$

Discussion on the smoothness $\tau_\ell=\tau$

The downside of the rate amplification procedure that was given in this section is that τ_ℓ does not decrease with ℓ (which is bad as, recall, we wish τ to be small as, by Claim 22, the distance δ of the resulted LCC is proportional to $1/\tau$). Indeed, with the notation of Proposition 34, $\tau_\ell=\tau$. By examining the proof and Lemma 32 one natural idea is to consider an SLR not over the entire set P^ℓ but on some subset of it which is a code with distance, say, $d>1$. This will indeed guarantee that for every two points p,r we have $\mathsf{dist}(p,r)\ge d$ and so the bound in Equation (4.2) will be τ^d rather than τ. While natural, this idea fails to yield better parameters as the rate-loss incurred by using a code (even an MDS) is larger than the improvement on the rate guaranteed via the rate amplification procedure.

In the next sections we give a more elaborate rate amplification procedure (that is based on the one that was given in this section) in which τ does decrease with ℓ. Roughly, $\tau_\ell=(q\cdot\log|P|)^{\mathrm{poly}(\ell)}\tau^\ell$, and so there is a slight loss in the smoothness, which the reader should think as negligible. The query complexity q_ℓ as well as the rate ρ_ℓ and ε_ℓ are all slightly worse than those obtained in the above rate amplification procedure and so the two techniques are incomparable.

4.3 Distance-efficient rate amplification

Let P be a set, and R a partition of P^2. We denote the part containing p by $[p]_R$ or $[p]$ when R is clear from context. We call $(p)=[p]\setminus\{p\}$ the *open class* of p. For a set $A\subseteq P^2$ we let $(A)=\cup_{p\in A}(p)$. Given $p\in P$ we say that $\{p\}\times P\subseteq P^2$ is *vertical line* and $P\times\{p\}$ is a *horizontal line*. Horizontal and vertical lines are referred to as *axis-parallel lines*, and we denote the set of such lines by

$$\mathcal{X}=\bigcup_{p\in P}\{\{p\}\times P, P\times\{p\}\}.$$

For a point $p = (p_1, p_2) \in P^2$ we denote $S_p = (\{p_1\} \times P) \cup (P \times \{p_2\}) \setminus \{p\}$. That is, S_p is the set of points in P^2 of distance exactly 1 from p. Key to our distance-efficient rate amplification procedure is a partition of the "square" P^2 with certain properties.

▶ **Definition 35** (Axis-evasive partitions). *Let P be a set. A partition R of P^2 is said to be* (c, s)-axis evasive *if*

1. *For every $p \in P^2$, $|(p)| \leq c$.*
2. *For every $\ell, \ell' \in \mathcal{X}$ (possibly equal), $|\ell' \cap (\ell)| \leq s$.*
3. *For every $p \in P^2$ and $\ell \in \mathcal{X}$, $|[p] \cap \ell| \leq 1$.*

In Section 5 we study such partitions. We prove their existence with certain parameters and give explicit constructions. In this section, however, we work with abstract axis-evasive partitions and analyze our rate amplification procedure with respect to the parameters c, s of the axis-evasive partition as well as the number of parts which we typically denote by t.

▷ Claim 36. Let $p, p' \in P^2$ (possibly equal). Then,

$$|\{r \in S_p \mid (r) \cap S_{p'} \neq \emptyset\}| \leq 4s.$$

Proof. Note that each of $S_p, S_{p'}$ is a subset of the union of two axis-parallel lines. Thus, to prove the claim, it suffices to show that for every $\ell, \ell' \in \mathcal{X}$, not necessarily distinct,

$$|\{r \in \ell \mid (r) \cap \ell' \neq \emptyset\}| \leq s.$$

Let $r_1, \ldots, r_t \in \ell$ be such that $(r_i) \cap \ell' \neq \emptyset$. Note that for every distinct $i, j \in [t]$ it holds that $((r_i) \cap \ell') \cap ((r_j) \cap \ell') = \emptyset$. Indeed, since R is a partition, if $((r_i) \cap \ell') \cap ((r_j) \cap \ell') \neq \emptyset$ then $r_i \in [r_j]$, but this implies that $|\ell \cap [r_j]| \geq 2$ in contradiction axis evasiveness. Thus,

$$R = \bigcup_{i=1}^{t} ((r_i) \cap \ell')$$

is a disjoint union of size t. However, $R \subseteq (\ell) \cap \ell'$, and so $t \leq |R| \leq |(\ell) \cap \ell'| \leq s$. ◁

▶ **Definition 37.** *Let P be a set, $\mathbb{F}$ a field. Let R be a (c, s)-axis evasive partition of P^2. For every $p \in P^2$ define the function $g_{[p]} : P^2 \to \mathbb{F}$ as follows:*

$$g_{[p]}(r) = \begin{cases} 1, & r \in [p]; \\ 0, & \text{otherwise.} \end{cases}$$

We define $\mathcal{L}_R = \{g_{[p]} \mid p \in P^2\}$.

▶ **Definition 38.** *Let P be a set, $\mathbb{F}$ a field. For $S \subseteq P$ define the function $\nu_S : P \to \mathbb{F}$ by*

$$\nu_S(r) = \begin{cases} 0, & r \in S; \\ 1, & \text{otherwise.} \end{cases}$$

For ease of readability, when S is a singleton $S = \{p\}$, we write ν_p instead of $\nu_{\{p\}}$.

With the notations and definitions above, we are ready to start developing our second rate amplification procedure. We start with the following.

▶ **Definition 39.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q, \tau, \varepsilon, \rho)$-dual SLR. Let $\mathcal{L}^{P^2}$ be as in Definition 29. Let R be a (c, s)-axis evasive partition of P^2. We define for every $p \in P^2$ the distribution $(D_R^{P^2})_p$ as follows. To sample u from $(D_R^{P^2})_p$:*

1. *Sample $g \sim D_p^{P^2}$.*
2. *If $g = \perp$ return $\perp$; Otherwise, denote $L = \{r \in S_p \mid g(r) \neq 0\}$ and proceed as follows.*
3. *For every $r \in L$ and $w \in (r)$ sample $h_{r,w} \sim D_w^{P^2}$.*
4. *If there exist $r \in L$ and $w \in (r)$ such that either $h_{r,w} = \perp$ or $h_{r,w}(p) \neq 0$ return $\perp$. Otherwise return*

$$u = g\nu_L + \sum_{r\in L} g(r) \sum_{w\in(r)} \frac{h_{r,w}\nu_w}{h_{r,w}(w)}. \tag{4.3}$$

Note that, upon reaching Step (4), u is well-defined as $h_{r,w}(w) \neq 0$ for all $r \in L$ and $w \in (r)$. We denote the collection of distributions $\{(D_R^{P^2})_p \mid p \in P^2\}$ by $\mathcal{D}_R^{P^2}$.

We start by analyzing the function u that is given by Equation (4.3) above.

▷ Claim 40. With the notation of Definition 39, if $\perp$ is not returned then $u \in \mathcal{F}_p$.

Proof. As $\perp$ was not returned, for every $r \in L$ and $w \in (r)$ it holds that $h_{r,w} \neq \perp$ and $h_{r,w}(p) = 0$. Substituting to Equation (4.3), we get

$$u(p) = g(p)\nu_L(p) = g(p) \neq 0,$$

where the second equality holds as $p \notin L$ and the last inequality follows since $g \in \mathsf{supp}(D_p^{P^2}) \setminus \{\perp\}$. ◁

▷ Claim 41. With the notation of Definition 39, if $\perp$ is not returned then $u \in \mathcal{L}^{P^2} + \mathcal{L}_R$.

Proof. Take $f \in (\mathcal{L}^{P^2} + \mathcal{L}_R)^\perp$. To prove the claim, it suffices to show that $\langle u, f\rangle = 0$. Indeed, this would imply $u \in ((\mathcal{L}^{P^2} + \mathcal{L}_R)^\perp)^\perp = \mathcal{L}^{P^2} + \mathcal{L}_R$. As $u \neq \perp$ we have that $g \neq \perp$. Note that

$$\langle g\nu_L, f\rangle = \langle g, f\rangle - \sum_{r\in L} g(r)f(r).$$

Since $g \in \mathsf{supp}(D_p^{P^2})$ we get that $g \in \mathcal{L}^{P^2}$. However, $f \in (\mathcal{L}^{P^2} + \mathcal{L}_R)^\perp \subseteq (\mathcal{L}^{P^2})^\perp$, implying $\langle g, f\rangle = 0$. Thus,

$$\langle g\nu_L, f\rangle = -\sum_{r\in L} g(r)f(r). \tag{4.4}$$

Now, fix $r \in L$ and $w \in (r)$. By Definition 39, as $u \neq \perp$ we have that $h_{r,w} \neq \perp$ and so $h_{r,w} \in \mathcal{L}^{P^2}$. However, by the above, $f \in (\mathcal{L}^{P^2})^\perp$ and so $\langle h_{r,w}, f\rangle = 0$. Thus,

$$\langle h_{r,w}\nu_w, f\rangle = \langle h_{r,w}, f\rangle - h_{r,w}(w)f(w) = -h_{r,w}(w)f(w).$$

Therefore, for every fixed $r \in L$ one has that

$$\begin{aligned}\left\langle \sum_{w\in(r)} \frac{h_{r,w}\nu_w}{h_{r,w}(w)}, f\right\rangle &= \sum_{w\in(r)} \left\langle \frac{h_{r,w}\nu_w}{h_{r,w}(w)}, f\right\rangle \\ &= \sum_{w\in(r)} \frac{1}{h_{r,w}(w)} \langle h_{r,w}\nu_w, f\rangle \\ &= -\sum_{w\in(r)} f(w). \end{aligned}\tag{4.5}$$

Now, $f \in (\mathcal{L}^{P^2} + \mathcal{L}_R)^\perp \subseteq (\mathcal{L}_R)^\perp$ whereas $g_{[r]} \in \mathcal{L}_R$, and so

$$0 = \langle f, g_{[r]} \rangle = \sum_{w \in [r]} f(w).$$

Substituting this to Equation (4.5), we get

$$\Big\langle \sum_{w \in (r)} \frac{h_{r,w} \nu_w}{h_{r,w}(w)}, f \Big\rangle = f(r).$$

Therefore,

$$\begin{aligned} \Big\langle \sum_{r \in L} g(r) \sum_{w \in (r)} \frac{h_{r,w} \nu_w}{h_{r,w}(w)}, f \Big\rangle &= \sum_{r \in L} g(r) \Big\langle \sum_{w \in (r)} \frac{h_{r,w} \nu_w}{h_{r,w}(w)}, f \Big\rangle \\ &= \sum_{r \in L} g(r) f(r). \end{aligned}$$

The above equation together with Equation (4.4) yield $\langle u, f \rangle = 0$. ◁

▷ Claim 42. With the notation of Definition 39, for every $p \in P^2$,

$$\mathbf{Pr}[(D_R^{P^2})_p = \perp] \le 18csq\tau^2 + 2cq\varepsilon.$$

Proof. First, the probability that $g = \perp$ is bounded by ε. Similarly, the probability that for any specific $r \in L$ and $w \in (r)$, $h_{r,w} = \perp$ is bounded by ε. Thus, by the union bound, and since $|L| \le 2q - 1$ and $|(r)| \le c$, we have that expect with probability $(1 + (2q-1)c)\varepsilon \le 2qc\varepsilon$, the sampling process above will result in $u \neq \perp$.

To complete the analysis, we turn to bound the probability that $h_{r,w}(p) = 0$ for some $r \in L$ and $w \in (r)$. Let $L = \{r_1, \ldots, r_{|L|}\}$. While the random variables in L may be dependent, marginally, it holds that for every $i \in [|L|]$ and every fixed $r \in S_p$, $\mathbf{Pr}[r_i = r] \le \tau$. With this notation, by Definition 39, $(D_R^{P^2})_p = \perp$ only if there exist $i \in [|L|]$ and $w \in (r_i)$ such that $h_{r_i,w}(p) \neq 0$.

For a fixed $r \in S_p$ define the event $\mathcal{E}_r$ in which there exists $w \in (r)$ such that $h_{r,w}(p) \neq 0$, (when conditioned on $h_{r,w} \neq \perp$). Note that this event is with respect to the randomness of sampling $h_r = \{h_{r,w} \mid w \in (r)\}$ whereas r is fixed. By the union bound,

$$\mathop{\mathbf{Pr}}_{h_r}[\mathcal{E}_r] \le \sum_{w \in (r)} \mathop{\mathbf{Pr}}_{h_{r,w}}[h_{r,w}(p) \neq 0 \mid h_{r,w} \neq \perp].$$

Observe first that $w \neq p$. Indeed, as $r \in S_p$, both r and p are on some common axis-parallel line $\ell \in \mathcal{X}$. Thus, $w = p$ would imply $|[r] \cap \ell| \ge 2$ which stands in contradiction to the definition of axis-evasiveness. Consider $w \in (r) \setminus S_p$. As $w \neq p$ we have that $\mathsf{dist}(w, p) = 2$. By Lemma 32, as $h_{r,w} \sim D_w^{P^2}$ we have that

$$\mathop{\mathbf{Pr}}_{h_{r,w}}[h_{r,w}(p) \neq 0 \mid h_{r,w} \neq \perp] \le \tau^2.$$

If, on the other hand, $w \in (r) \cap S_p$ then $\mathsf{dist}(w, p) = 1$, and Lemma 32 then implies that

$$\mathop{\mathbf{Pr}}_{h_{r,w}}[h_{r,w}(p) \neq 0 \mid h_{r,w} \neq \perp] \le \tau.$$

As $|(r)| \le c$ we conclude that

$$\mathop{\mathbf{Pr}}_{h_r}[\mathcal{E}_r] \le c\tau^2 + \tau|(r) \cap S_p|.$$

Fix $i \in [|L|]$ and consider the random variable r_i. The above equation, together with $|(r_i)| \le c$, yields

$$\begin{aligned}\mathop{\mathbf{Pr}}_{r_i,h_{r_i}}[\mathcal{E}_{r_i}] &\le \mathop{\mathbf{Pr}}_{r_i,h_{r_i}}[\mathcal{E}_{r_i} \mid (r_i) \cap S_p = \emptyset] + \mathop{\mathbf{Pr}}_{r_i,h_{r_i}}[\mathcal{E}_{r_i} \mid (r_i) \cap S_p \neq \emptyset] \mathop{\mathbf{Pr}}_{r_i}[(r_i) \cap S_p \neq \emptyset] \\ &\le c\tau^2 + (c\tau^2 + c\tau) \mathop{\mathbf{Pr}}_{r_i}[(r_i) \cap S_p \neq \emptyset]. \end{aligned} \tag{4.6}$$

Consider now the set $B = \{r \in S_p \mid (r) \cap S_p \neq \emptyset\}$. As R is (c,s)-axis evasive, Claim 36 implies that $|B| \le 4s$, and so

$$\mathop{\mathbf{Pr}}_{r_i}[(r_i) \cap S_p \neq \emptyset] = \mathbf{Pr}[r_i \in B] \le 4s\tau.$$

Substituting to Equation (4.6), we get $\mathbf{Pr}[\mathcal{E}_i] \le 9cs\tau^2$. The proof then follows by taking the union bound over all $i \in [|L|]$ as, indeed, $|L| = 2q - 1$. ◁

▷ Claim 43. With the notation of Definition 39, for every pair of distinct $p, r \in P^2$,

$$\mathop{\mathbf{Pr}}_{u \sim (D_R^{P^2})_p}[u(r) \neq 0 \mid u \neq \perp] \le 10csq\tau^2.$$

Proof. By Equation (4.3), u is a linear combination of the (sampled) functions $g\nu_L$, $\{h_{r,w}\nu_w\}$. To prove the claim, we will show that, with high probability, all these functions evaluate to 0 at the point r, implying $u(r) = 0$. We start by bounding $\mathbf{Pr}[(g\nu_L)(r) \neq 0]$. To this end, consider two cases. First, if $r \in P^2 \setminus S_p$ then, as $L \subseteq S_p$, we have that $\nu_L(r) = 1$ and so in such case

$$\mathbf{Pr}[(g\nu_L)(r) \neq 0] = \mathbf{Pr}[g(r) \neq 0] \le \tau^2, \tag{4.7}$$

where the last inequality follows by Lemma 32 and since $\mathsf{dist}(r,p) = 2$ per our assumption $r \notin S_p$ and since $r \neq p$. If, on the other hand, $r \in S_p$ then, by the definition of L,

$$g(r) \neq 0 \implies r \in L \implies \nu_L(r) = 0,$$

and so in this case $(g\nu_L)(r) = 0$.

Let $L = \{r_1, \ldots, r_{|L|}\}$. Consider a fixed $i \in [|L|]$ and denote $(r_i) = \{w_{i,1}, \ldots, w_{i,b}\}$, where $b \le c$. Fix $j \in [b]$. We turn to bound $\mathbf{Pr}\left[(h_{r_i,w_{i,j}}\nu_{w_{i,j}})(r) \neq 0\right]$. First note that

$$\mathbf{Pr}\left[(h_{r_i,w_{i,j}}\nu_{w_{i,j}})(r) \neq 0 \mid (r_i) \cap S_r = \emptyset\right] \le \tau^2. \tag{4.8}$$

Indeed, conditioned on the event $(r_i) \cap S_r = \emptyset$, either $w_{i,j} = r$ or $\mathsf{dist}(w_{i,j}, r) = 2$. In the first case,

$$(h_{r_i,w_{i,j}}\nu_{w_{i,j}})(r) = h_{r_i,r}(r)\nu_r(r) = 0.$$

In the second case, the bound follows by Lemma 32. Second, note that

$$\mathbf{Pr}\left[(h_{r_i,w_{i,j}}\nu_{w_{i,j}})(r) \neq 0 \mid (r_i) \cap S_r \neq \emptyset\right] \le \tau. \tag{4.9}$$

Indeed, as before, we may only consider the case $r \neq w_{i,j}$ and then observe that $\mathsf{dist}(r, w_{i,j}) = 1$ and invoke Lemma 32. Now, let $B = \{v \in S_p \mid (v) \cap S_r \neq \emptyset\}$. By Claim 36, and since R is (c,s)-axis evasive, $|B| \le 4s$. Recall that $\mathbf{Pr}[r_i = v] \le \tau$ for every fixed $v \in S_p$, and so

$$\mathbf{Pr}[(r_i) \cap S_r \neq \emptyset] = \mathbf{Pr}[r_i \in B] \le 4s\tau. \tag{4.10}$$

By Equations (4.8), (4.9), (4.10) we get

$$\mathbf{Pr}\left[(h_{r_i,w_{i,j}}\nu_{w_{i,j}})(r) \neq 0\right] \le \tau^2 + 4s\tau^2 \le 5s\tau^2.$$

The proof then follows by the union bound over all $i \in [|L|]$ and $j \in [|(w_i)|]$. ◁

▶ **Definition 44.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q, \tau, \varepsilon, \rho)$-dual SLR. Let $\mathcal{L}^{P^2}$ be as in Definition 29. Let R be a (c, s)-axis evasive partition of P^2 and let $\mathcal{D}_R^{P^2}$ be as in Definition 39. We denote by $(\mathcal{D}^P, \mathcal{L}^P)_R^2$ the pair $(\mathcal{D}_R^{P^2}, \mathcal{L}^{P^2} + \mathcal{L}_R)$.*

▶ **Proposition 45.** *Let P be a set, $\mathbb{F}$ a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q, \tau, \varepsilon, \rho)$-dual SLR. Let R be a (c, s)-axis evasive partition of P^2 that consists of t parts. Then, $(\mathcal{D}^P, \mathcal{L}^P)_R^2$ is a $(q_R, \tau_R, \varepsilon_R, \rho_R)$-dual SLR with*

$$\begin{aligned}
&q_R \le 2cq^3 \\
&\tau_R \le 10csq\tau^2 \\
&\varepsilon_R \le 18csq\tau^2 + 2cq\varepsilon \\
&\rho_R \ge 1 - (1-\rho)^2 - \frac{t}{|P|^2}.
\end{aligned}$$

Proof. Claim 40 implies that for every $p \in P^2$, $\mathsf{supp}((D_R^{P^2})_p) \subseteq \mathcal{F}_p \cup \{\perp\}$. To bound q_R, note that by Equation (4.3),

$$|u| \le |g\nu_L| + \sum_{r \in L} \sum_{w \in (r)} |h_{r,w}\nu_w|$$

Now, $|g\nu_L| \le |g| \le q^2$ and $|h_{r,w}\nu_w| \le |h_{r,w}| \le q^2$. Hence, $|u| \le q^2 + |L|cq^2 \le 2cq^3$. The stated bounds on τ_R and ε_R readily follows by Claim 43 and Claim 42, respectively. As for the rate, we have that

$$\begin{aligned}
\dim(\mathcal{L}^{P^2} + \mathcal{L}_R) &\le \dim(\mathcal{L}^{P^2}) + \dim(\mathcal{L}_R) \\
&\le (1-\rho)^2|P|^2 + t,
\end{aligned}$$

where the second inequality follows by Proposition 34 and since R consists of t parts, implying $|\mathcal{L}_R| = t$. ◀

4.4 Proofs of Theorem 3 and Corollary 4

With the machinery developed in the previous section, and using in a black-box manner, the construction of axis-evasive partitions we obtain in Section 5, we are finally ready to prove Theorem 3 and Corollary 4. We start by giving a more formal statement of Corollary 4.

▶ **Theorem 46.** *There exist universal constants $m_0, c' \ge 1$ such that the following holds. Let P be a set of size $m \ge m_0$. Let $\mathbb{F}$ be a field, and let $(\mathcal{D}_{\text{in}}^P, \mathcal{L}_{\text{in}}^P)$ be a $(q_{\text{in}}, \tau_{\text{in}}, \varepsilon_{\text{in}}, \rho_{\text{in}})$-dual SLR over $\mathbb{F}^P$. Let $0 < \alpha < 1$ be such that*

$$\rho_{\text{in}} \ge \frac{c'}{\sqrt{\alpha \cdot \log m}} \log\left(\frac{1}{\alpha}\right). \tag{4.11}$$

Then, there exists a $(q_{\text{out}}, \tau_{\text{out}}, \varepsilon_{\text{out}}, \rho_{\text{out}})$-dual SLR $(\mathcal{D}_{\text{out}}^P, \mathcal{L}_{\text{out}}^P)$ over $\mathbb{F}^{P_{\text{out}}}$, with $m^\ell/2 \le |P_{\text{out}}| \le m^\ell$, where

$$\ell = \Theta\left(\frac{1}{\rho_{\text{in}}} \log \frac{1}{\alpha}\right), \tag{4.12}$$

having the following parameters:

$$\begin{aligned}
&q_{\text{out}} \le q_{\text{in}}^{\text{poly}(\ell)}, \\
&\tau_{\text{out}} \le q_{\text{in}}^{\text{poly}(\ell)} \tau_{\text{in}}^{\ell}, \\
&\varepsilon_{\text{out}} \le q_{\text{in}}^{\text{poly}(\ell)} (\tau_{\text{in}} + \varepsilon_{\text{in}}), \\
&\rho_{\text{out}} \ge 1 - \alpha.
\end{aligned}$$

A remark regarding the error

Note that there is another implicit constraint on ρ_{in} and α that originates from the error. Indeed, to make the result non-trivial, one must have $\varepsilon_{\text{out}} < 1$ which, in turn, implies some bound on ℓ and then, through Equation (4.12), a constraint on ρ_{in} and α. However, if that turns out to be a problem for the regime of parameters one is interested in, the probability to output $\perp$ can be reduced by repetition. Thus, by performing an alternating sequence of such error (or failure) reductions and rate amplifications, one can resolve this issue. Note that unlike for LDC, the error reduction has no cost in query complexity, and it certainly has no effect on the smoothness nor on the rate. It does, however, effects the running-time.

As mentioned above, our proof relies on an explicit axis-evasive partition that we construct in Section 5. Formally,

▶ **Theorem 47.** *Let P be a set of size q, where q is an odd prime power. Let c be an even integer such that $c+1 \mid q+1$, and $c \leq \sqrt{q}/10$. Then, there exists a $(c, 4c^2)$-axis evasive partition of P^2 with at most $2q^2/(c+1)$ parts.*

Our proof of Theorem 46 is done by applying Proposition 45 several times, iteratively, where in each iteration we square the size of the set P obtained by the previous iterative step. Note, however, that Theorem 47 requires the set size $|P|$ to be an odd prime power q with the property that $c+1 \mid q+1$. It is best to choose c the same in all applications of Proposition 45. However, note that if we start an iteration with a set of size q and so end the iteration with a set of size q^2 then the condition will fail to hold at the beginning of the following iteration. Indeed if $c+1 \mid q+1$ then $q \equiv -1 \pmod{c+1}$ and so $q^2 \equiv 1 \pmod{c+1}$. To overcome this technicality, we do not work with the set obtained by the previous iteration as is. Instead, we find a prime–not much smaller than q^2–that has the desired residue -1 modulo $c+1$. To this end we rely on the Siegel–Walfisz Theorem [34, 37] which refines Dirichlet's theorem on primes in arithmetic progressions. The state the Siegel–Walfisz Theorem we set some notation. For an integer $m \geq 1$, we denote Euler's totient function, that counts the positive integers up to m that are relatively prime to m, by $\phi(m)$. For integers n, m, r, we denote the number of (positive) primes less than or equal to n which are congruent to r modulo m by $\pi(n; m, r)$. The *Eulerian logarithmic integral* is given by

$$\mathrm{Li}(x) = \int_2^x \frac{dt}{\ln t} .$$

▶ **Theorem 48** ([34, 37])**.** *For every constant $e \geq 1$ there exists a constant $c = c(e)$ such that the following holds. Let n, m, r be positive integers such that $m \leq (\log n)^e$, and m, r coprimes. Then,*

$$\left| \pi(n; m, r) - \frac{\mathrm{Li}(n)}{\phi(m)} \right| = O\left(n \cdot 2^{-c\sqrt{\log n}} \right).$$

We have the following straightforward corollary.

▶ **Corollary 49.** *For every constant $e \geq 1$ there exist constants $c = c(e)$, $n_0 = n_0(e)$ such that the following holds. Let m, r be coprime integers, $m > 0$. Let $n \geq n_0$ be an integer such that $m \leq (\log n)^e$. Then, there exists a prime $p \in [n - \Delta, n]$, where $\Delta = cn/\log n$, such that $p \equiv r \pmod{m}$.*

Proof. To prove the corollary, it suffices to show that $\pi(n; m, r) > \pi(n - \Delta; m, r)$. By Theorem 48, there exist constants n_0, c' such that for every $n \geq n_0$,

$$\left| \pi(n; m, r) - \frac{\mathrm{Li}(n)}{\phi(m)} \right| \leq c' n \cdot 2^{-c\sqrt{\log n}}.$$

Thus, it suffices to show that

$$\frac{\mathrm{Li}(n)}{\phi(m)} - c'n \cdot 2^{-c\sqrt{\log n}} > \frac{\mathrm{Li}(n-\Delta)}{\phi(m)} + c'(n-\Delta) \cdot 2^{-c\sqrt{\log (n-\Delta)}}.$$

As we may assume that $\Delta \leq n/2$, it suffices to prove that

$$\mathrm{Li}(n) - \mathrm{Li}(n-\Delta) \geq 2c'\phi(m)n \cdot 2^{-c\sqrt{\log(n/2)}}. \tag{4.13}$$

It is well-known that

$$\mathrm{Li}(x) = c_1 + \frac{x}{\ln x} + O\left(\frac{x}{\ln^2 x}\right),$$

where $c_1 = \int_{t=0}^{2} \frac{dt}{\ln t}$ is some constant. Therefore,

$$\mathrm{Li}(n) - \mathrm{Li}(n-\Delta) \geq \frac{\Delta}{\ln(n/2)} - \frac{c''n}{\ln^2 n}.$$

for some constant c''. By our assumption on Δ we can choose the parameter c in the definition of Δ such that the right hand side is bounded below by $n/\ln^2 n$. The proof then follows by Equation (4.13) and noting that $\phi(m) \leq m \leq (\log n)^e = o(2^{-c\sqrt{\log (n/2)}})$. ◀

We turn to formally define and analyze the operation of projecting a dual SLR over $\mathbb{F}^P$ on a (large) subset of P.

▶ **Definition 50.** *Let P a set and $P' \subseteq P$. Let $p' \in P'$ and D be a distribution with $\mathsf{supp}(D) \subseteq \mathcal{F}_{p'} \cup \{\perp\}$. We define the $D|_{P'}$ as follows: To sample from $D|_{P'}$, sample $f \sim D$. If $f =\perp$, output $\perp$; if $f \in \mathcal{F}_{p'}$, output $f|_{P'}$. We refer to $D|_{P'}$ as the distribution D projected to P'.*

▶ **Definition 51.** *Let P be a set, $\mathbb{F}$ a field. Let $\mathcal{D} = \{D_p \mid p \in P\}$ be a collection of distributions, where for each $p \in P$, $\mathsf{supp}(D_p) \subseteq \mathcal{F}_p \cup \{\perp\}$. Let $P' \subseteq P$. We define $\mathcal{D}|_{P'}$ to be the collection $\mathcal{D}$ projected to P', that is, $\mathcal{D}|_{P'} = \{D_{p'}|_{P'} \mid p' \in P'\}$.*

▶ **Definition 52.** *Let P be a set, $\mathbb{F}$ a field and let $\mathcal{L}$ be a linear subspace of $\mathbb{F}^P$. Let $P' \subseteq P$. We denote by $\mathcal{L}|_{P'}$ the linear subspace $\mathcal{L}$ projected to P', namely, $\mathcal{L}|_{P'} = \{f|_{P'} \mid f \in \mathcal{L}\}$.*

▷ Claim 53. Let P be a set, $\mathbb{F}$ a field, $(\mathcal{D}, \mathcal{L})$ a $(q, \tau, \varepsilon, \rho)$-dual SLR over $\mathbb{F}^P$, and let $P' \subseteq P$. Then, $(\mathcal{D}|_{P'}, \mathcal{L}_{P'})$ is a $(q, \tau, \varepsilon, \rho')$-dual SLR over $\mathbb{F}^{P'}$, where $\rho' = 1 - \frac{|P|}{|P'|}(1-\rho)$.

Proof. That the smoothness τ, as well as q and ε, all stay the same after projecting the dual SLR to P', follows immediately from the definitions. The assertion regarding the rate of the induced SLR, ρ', readily follows as we have that

$$\dim(\mathcal{L}|_{P'}) \leq \dim(\mathcal{L}) \leq (1-\rho)|P| = \left(1 - (1 - \frac{|P|}{|P'|}(1-\rho))\right)|P'|. \quad ◁$$

▷ Claim 54. There exists a universal constant m_0 such that the following holds. Let P be a set of size $m \geq m_0$. Let $\mathbb{F}$ be a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q_{\mathrm{in}}, \tau_{\mathrm{in}}, \varepsilon_{\mathrm{in}}, \rho_{\mathrm{in}})$-dual SLR over $\mathbb{F}^P$. Let $c \leq \log m$ be an integer. Then, there exists a set P' of size

$$|P'| \geq \left(1 - O\left(\frac{1}{\log m}\right)\right)m^2,$$

and a $(q_{\text{out}}, \tau_{\text{out}}, \varepsilon_{\text{out}}, \rho_{\text{out}})$-dual SLR $(\mathcal{D}^{P'}, \mathcal{L}^{P'})$ over $\mathbb{F}^{P'}$, where

$$\begin{aligned}
&q_{\text{out}} \le 2cq_{\text{in}}^3,\\
&\tau_{\text{out}} \le 40c^3 q_{\text{in}} \tau_{\text{in}}^2,\\
&\varepsilon_{\text{out}} \le 80c^3 q_{\text{in}}(\tau_{\text{in}}^2 + \varepsilon_{\text{in}}),\\
&\rho_{\text{out}} \ge 1 - (1-\rho_{\text{in}})^2 - O\left(1/c\right).
\end{aligned}$$

Proof. By Corollary 49 applied with n, m, r in the notation of Corollary 49 set to $m, c+1, -1$ in the notation of this claim, respectively, there exists some prime $p \le m$ such that $m - p = O(\frac{m}{\log m})$, and $c+1 \mid p+1$. Take P' to be an arbitrary subset of P of size p. By Claim 53, $(\mathcal{D}|_{P'}, \mathcal{L}|_{P'})$ is a $(q_{\text{in}}, \tau_{\text{in}}, \varepsilon_{\text{in}}, \rho')$-dual SLR on P', where

$$\rho' = 1 - \frac{m}{p}(1-\rho_{\text{in}}) \ge \rho_{\text{in}} - O\left(\frac{1}{\log m}\right).$$

By Theorem 47 applied to P', which observe is indeed applicable as $c+1 \mid p+1$, there exists an explicit $(c, 4c^2)$-axis evasive partition R of $(P')^2$ with at most $t = 2p^2/(c+1)$ parts. With that partition, we can now apply Proposition 45 to $(\mathcal{D}|_{P'}, \mathcal{L}|_{P'})$ and get that $(\mathcal{D}|_{P'}, \mathcal{L}|_{P'})^2_R$ is a $(q_{\text{out}}, \tau_{\text{out}}, \varepsilon_{\text{out}}, \rho_{\text{out}})$-dual SLR with the stated parameters. Note that the assertion regarding the rate follows as $c \le \log m$, ◁

The following proposition is a more formal and accurate restatement of Theorem 3.

▶ **Proposition 55.** *There exist universal constants $0 < c' < 1$ and $c'', m', \ell' \ge 1$ such that the following holds. Let P be a set of size $m \ge m'$. Let $\mathbb{F}$ be a field, and let $(\mathcal{D}^P, \mathcal{L}^P)$ be a $(q_{\text{in}}, \tau_{\text{in}}, \varepsilon_{\text{in}}, \rho_{\text{in}})$-dual SLR over $\mathbb{F}^P$. Let $\ell = 2^r$ for an integer $r \ge 1$, and assume that $\ell \ge \ell'$. Let c be an integer such that $c''\ell^2 \le c \le c' \log m$. Then, there exists a set P_ℓ of size $m^\ell/2 \le |P_\ell| \le m^\ell$, and a $(q_\ell, \tau_\ell, \varepsilon_\ell, \rho_\ell)$-dual SLR $(\mathcal{D}^{P_\ell}, \mathcal{L}^{P_\ell})$ over $\mathbb{F}^{P_\ell}$, where*

$$\begin{aligned}
&q_\ell \le (2cq_{\text{in}})^{\ell^{\log 3}},\\
&\tau_\ell = O((c^3 q_{\text{in}})^{\ell^{\log 3}}) \cdot \tau_{\text{in}}^\ell,\\
&\varepsilon_\ell \le O((c^4 q_{\text{in}})^{\ell^{\log 3}}) \cdot (\tau_{\text{in}} + \varepsilon_{\text{in}}),\\
&\rho_\ell \ge 1 - (1-\rho_{\text{in}})^\ell - O\left(\frac{\ell^2}{c}\right),
\end{aligned}$$

where, recall, the log *function is taken base* 2*.*

Proof. We construct a sequence of $(q_t, \tau_t, \varepsilon_t, \rho_t)$-dual SLR $(\mathcal{D}^{P_t}, \mathcal{L}^{P_t})$ for $t = 0, 1, \ldots, r = \log \ell$, and show that the last dual-SLR in the sequence has the stated parameters. The first dual-SLR, $(\mathcal{D}^{P_0}, \mathcal{L}^{P_0})$, is taken to be the $(q_{\text{in}}, \tau_{\text{in}}, \varepsilon_{\text{in}}, \rho_{\text{in}})$-dual SLR $(\mathcal{D}^P, \mathcal{L}^P)$ that is given by the hypothesis of the proposition. After constructing $(\mathcal{D}^{P_t}, \mathcal{L}^{P_t})$, we obtain $(\mathcal{D}^{P_{t+1}}, \mathcal{L}^{P_{t+1}})$ by applying Claim 54 to $(\mathcal{D}^{P_t}, \mathcal{L}^{P_t})$ with the parameter c taken to be c from the statement of this proposition. Note that, as required by the claim, $c \le \log m$. Note that, by taking m' to be a large enough constant, all other dual SLR in the sequence will have $|P_t| \ge m$ as well, and so we can apply Claim 54 to them. Denote $m_t = |P_t|$. We begin by bounding m_t from below. Indeed, by Claim 54, and using that $1 - x \ge e^{-2x}$ for $x \le 1/2$, we can pick the constant c'' such that

$$m_t \ge e^{-\frac{c''}{\log m_{t-1}}} m_{t-1}^2 \ge e^{-\frac{c''}{\log m_0}} m_{t-1}^2,$$

where the last inequality follows as, for a large enough constant m', the sequence $(m_t)_t$ is monotone increasing. We invoke Claim 82 with $a = e^{\frac{c''}{2\log m_0}}$ and $b = 2$ to conclude that

$$m_t \geq m_0^{2^t} e^{-\frac{c''2^t}{\log m_0}} \geq \frac{1}{2} m_0^{2^t},$$

where the last inequality follows as $t \leq r = \log \ell$ and, recall, we take $\ell \leq c' \log m$ for a sufficiently small constant $c' > 0$. In particular, $m_r \geq m^\ell/2$ as stated.

By Claim 54, for every $t \geq 1$ we have $q_t \leq 2cq_{t-1}^3$. It is straightforward to prove by that

$$q_t \leq (2cq_{\text{in}})^{3^t}, \tag{4.14}$$

which readily implies the assertion regarding the query complexity. We turn to analyze the rate. Denote $\beta_t = 1 - \rho_t$. Claim 54 implies that $\beta_t \leq \beta_{t-1}^2 + c'''/c$, for some constant $c''' > 0$. By induction on t, we get that $\beta_t \leq \beta_0^{2^t} + c'''4^t/c$. Indeed, the base case $t = 0$ is obvious. Now, by the induction hypothesis,

$$\beta_t \leq \beta_{t-1}^2 + \frac{c'''}{c} \leq \left(\beta_0^{2^{t-1}} + 4^{t-1}\frac{c'''}{c}\right)^2 + \frac{c'''}{c}.$$

One can easily verify that the right hand side is bounded above by the desired bound $\beta_0^{2^t} + c'''4^t/c$ provided that $2^t c'''/c \leq 1$. As $t \leq r$ and $2^r = \ell$, the latter inequality follows assuming $c'''\ell \leq c$. As we assume $c \geq c''\ell^2$, it suffices to choose ℓ' from the statement of the proposition to be a constant larger than the constant c'''/c''. We conclude that,

$$\beta_r \leq \beta_0^\ell + O\left(\frac{4^r}{c}\right) = \beta_0^\ell + O\left(\frac{\ell^2}{c}\right),$$

which implies the assertion regarding the rate.

As for the smoothness, by Claim 54, and using Equation (4.14), we have that

$$\tau_t \leq 40c^3 q_{t-1}\tau_{t-1}^2 \leq 40c^3 \left(2cq_{\text{in}}\right)^{3^{t-1}} \tau_{t-1}^2,$$

from which it is easy to verify that

$$\tau_t \leq (40c^3)^{2^t} (2cq_{\text{in}})^{3^t} \tau_0^{2^t},$$

and the assertion regarding the smoothness readily follows. Last is the error which we leave to the reader to verify. ◀

We can now easily deduce Theorem 46

Proof of Theorem 46. The proof readily follows from Proposition 55 by taking ℓ as defined in Equation (4.12), and with c in the notation of Proposition 55 taken to be $c = \Theta(\ell^2/\alpha)$. Note that this choice of parameters satisfies the hypothesis of Proposition 55 as indeed implied by Equation (4.11) and by taking c' to be a sufficiently large constant. It is easy to verify that the rate is $1 - \alpha$ with our choice of c, ℓ, and the remaining assertions readily follow by Proposition 55. ◀

5 Axis-evasive partitions

The distance-efficient rate amplification procedure that was developed in the previous section is built on axis-evasive partitions. Note that, by Proposition 45, the number of parts t effects the rate, c effects the query complexity and both c, s the deterioration of the distance and error. It is perhaps best to consider the following goal: for a given c we wish to obtain a (c, s)-axis evasive partition with both s, t as small as possible.

We start this section by proving the existence of axis-evasive partitions with great parameters. However, our probabilistic proof does not work for every c but rather, it requires $c = \Omega(\log m)$, where $m = |P|$. Unfortunately, for our distance-efficient rate amplification procedure, we are interested in $c < \log m$ (see Proposition 55). Luckily, and perhaps somewhat surprisingly, our explicit construction, described in Section 5.2, does work for every c albeit it requires $c + 1 \mid m + 1$ to hold.

5.1 Existential proof

As mentioned above, while we do not use the following non-constructive proof for the existence of axis-evasive sets, as given by the following lemma, we believe the reader might benefit from reading it still, as it gives an intuition on what is it about axis-evasive partitions which is random and what requires structure.

▶ **Lemma 56.** *Let P be a set of size m, and let c be an integer such that $50 \log m \le c \le \sqrt{m}$. Then, there exists a $(c, s = c)$-axis evasive partition of P^2 with $t \le 5m^2/c$ parts.*

Proof. Let $k = 2m^2/c$. The proof is by a probabilistic argument. We form a partition by assigning to each point $p \in P^2$ a "color" or, more formally, a number in $[k]$. The k parts are then formed by grouping together points with the same color. To this end, for every $p \in P^2$ define a random variable C_p that is uniformly distributed over $[k]$, where $\{C_p \mid p \in P^2\}$ are independent. For $i \in [k]$ let R_i be the number of random variables C_p for which $C_p = i$. Note that R_i is the size of part i, and that $\mathbf{E}[R_i] = c/2$. For every fixed $i \in [k]$, by the Chernoff bound,

$$\mathbf{Pr}\left[R_i \notin [c/4, c]\right] \le 2e^{-c/16}.$$

Thus, by the union bound over $i \in [k]$ and per our assumption $c \ge 50 \log m$, we have that except for probability $1/4$, for every $i \in [k]$, $R_i \in [c/4, c]$.

Now, we would want to claim that this partition satisfies the third condition, meaning that for every $p \in P^2$ and $\ell \in \mathcal{X}$, $|[p] \cap \ell| \le 1$. However, with high probability, this property in fact does not hold. To fix this, we make a slight modification to the random partition above so that it does satisfy the requirement. The change, is simply, given a partition - whenever there is a "collision" on a line $\ell \in \mathcal{X}$, meaning that for some distinct $p, r \in \ell$, $C_p = C_r$, assign new and distinct parts to both p and r. To analyze the number of additional parts we need, we introduce the following notation. For $\ell \in \mathcal{X}$ let

$$\nu(\ell) = \{\{p, r\} \mid p, r \in \ell \text{ and } p \ne r\}.$$

For $v = \{p, r\} \in \nu(\ell)$ define $\mathbb{I}_v^\ell$ to be an indicator for the event that $C_p = C_r$. With this notation, the number of collisions is bounded by $\sum_{\ell \in \mathcal{X}} \sum_{v \in \nu(\ell)} \mathbb{I}_v^\ell$. It holds that

$$\mathbf{E}\left[\sum_{\ell \in \mathcal{X}} \sum_{v \in \nu(\ell)} \mathbb{I}_v^\ell\right] = 2m\binom{m}{2}\frac{1}{k} < \frac{mc}{2}.$$

Therefore, by Markov's inequality, with probability at least $1/2$, the number of collisions is less than mc. In such case, we can add at most mc parts to the partition and be guaranteed that for every $p \in P^2$ and $\ell \in \mathcal{X}$, $|[p] \cap \ell| \leq 1$. Recall that since, prior to the procedure above, every part has size at least $c/4$ the total number of parts is now bounded by

$$t \leq mc + \frac{m^2}{c/4} \leq \frac{5m^2}{c},$$

where the last inequality follows as we assume $c \leq \sqrt{m}$.

To conclude the proof, it suffices to show that, with probability larger than $7/8$, it holds that for every $\ell, \ell' \in \mathcal{X}$, $|\ell' \cap (\ell)| \leq c$. Note that it suffices to prove this with respect to the partition obtained prior to the procedure above since, by introducing new parts of size one each, one only decrease the intersection size we aim to bound from above. Denote by $C_\ell = \{C_p \mid p \in \ell\} \subseteq [k]$. We have that

$$|\ell \cap (\ell')| = \left|\ell \cap \bigcup_{p \in \ell'} (p)\right| \leq 1 + \left|\{p \in \ell' \setminus \ell \mid C_p \in C_\ell\}\right|.$$

Now, by the union bound,

$$\mathbf{Pr}\left[C_p \in C_\ell\right] \leq \frac{m}{k} = \frac{c}{2m}.$$

As $\{C_p \mid p \in \ell'\}$ are chosen independently, by the Chernoff bound,

$$\mathbf{Pr}\left[|\{p \in \ell' \setminus \ell \mid C_p \in C_\ell\}| \geq c\right] \leq e^{-c/6} \leq \frac{1}{m^3},$$

where for the last inequality was used our assumption $c \geq 50 \log m$. The proof then follows by taking the union bound over all $\ell, \ell' \in \mathcal{X}$. ◀

5.2 Explicit constructions

In this section we give explicit constructions of axis-evasive partitions (see Definition 35). Our constructions are based on quadratic field extensions. We identify a set P of size q–a prime power–with the finite field $\mathbb{F}_q$ in an arbitrary manner, namely, by using an arbitrary bijection which, for ease of readability, we do not make explicit in the notation. We start by giving some basic background on finite fields.

Let $h(x) \in \mathbb{F}_q[x]$ be a degree 2 irreducible monic polynomial. It is a well-known fact that $\mathbb{F}_q[x]/\langle h(x)\rangle$ is a field of size q^2 which we denote, somewhat less informatively, by $\mathbb{F}_{q^2}$. Note that there exists $\alpha \in \mathbb{F}_{q^2}$ such that $h(\alpha) = 0$ (indeed, take $\alpha = x + \langle h(x)\rangle$). Since h is irreducible over $\mathbb{F}_q$ and has degree 2, we can write every element of $\mathbb{F}_{q^2}$ in the form $a + \alpha b$, where $a, b \in \mathbb{F}_q$, in a unique manner. That is, we can identify in the set-theoretic level, $\mathbb{F}_{q^2}$ with $\mathbb{F}_q + \alpha\mathbb{F}_q$. Using this identification, we identify P^2 with $\mathbb{F}_{q^2}$ in the natural way, namely, a point $(a, b) \in P^2$ is identified with $a + \alpha b$ in $\mathbb{F}_{q^2}$. Note that, with this identification, the horizontal lines in P^2 are of the form $b\alpha + \mathbb{F}_q$ where $b \in \mathbb{F}_q$ can be thought of as the fixed height of the line. Similarly, the vertical lines are given by $b + \alpha\mathbb{F}_q$. Given $\delta \in \mathbb{F}_{q^2} \setminus \{0\}$, we say that $\ell_\delta = \delta\mathbb{F}_q \subseteq \mathbb{F}_{q^2}$ is the line through the origin with slope δ.

Our construction of exis-evasive partitions is based on an equivalence relation that we are about to define. The partition is then obtained by considering the respective equivalence classes. We begin the construction by ignoring the "origin" $0 \in \mathbb{F}_{q^2}$ and work only with $\mathbb{F}_{q^2} \setminus \{0\}$. Note that this is the set of invertible elements of $\mathbb{F}_{q^2}$ which has a group structure under the field multiplication. When referring to this multiplicative group we write $(\mathbb{F}_{q^2})^\times$.

Let $\beta \in (\mathbb{F}_{q^2})^\times$. Denote by $o(\beta)$ the order of β in the multiplicative group $(\mathbb{F}_{q^2})^\times$. It will be convenient to denote $c = o(\beta) - 1$. We define an equivalence relation on $(\mathbb{F}_{q^2})^\times$, parameterized by β, as follows: For $\gamma, \delta \in (\mathbb{F}_{q^2})^\times$

$$\gamma \sim \delta \quad \iff \quad \gamma\delta^{-1} \in \langle\beta\rangle, \tag{5.1}$$

where $\langle\beta\rangle$ is the subgroup of $(\mathbb{F}_{q^2})^\times$ that is generated by β. Observe that this is an equivalence relation. Indeed, the classes are the different cosets, that is, the elements of the quotient group $(\mathbb{F}_{q^2})^\times/\langle\beta\rangle$. For completeness, we quickly prove that this is an equivalence relation: as $1 \in \langle\beta\rangle$, we have that $\gamma \sim \gamma$. Secondly, if $\gamma\delta^{-1} \in \langle\beta\rangle$ then $\delta\gamma^{-1} \in \langle\beta^{-1}\rangle = \langle\beta\rangle$ which establishes symmetry. As for transitivity, if $\gamma \sim \delta$ and $\delta \sim \varepsilon$ then

$$\gamma\varepsilon^{-1} = \gamma(\delta^{-1}\delta)\varepsilon^{-1} = (\gamma\delta^{-1})(\delta\varepsilon^{-1}) \in \langle\beta\rangle.$$

One can easily see that the equivalence class of an element $\gamma \in (\mathbb{F}_{q^2})^\times$ is $[\gamma] = \gamma\langle\beta\rangle = \{\gamma, \beta\gamma, \ldots, \beta^c\gamma\}$. Note further that $|[\gamma]| = c + 1$. Indeed, if there are $0 \le j < i \le c$ such that $\beta^i\gamma = \beta^j\gamma$ then $0 = (\beta^i - \beta^j)\gamma = (\beta^{i-j} - 1)\beta^j\gamma$, which is a contradiction as none of the factors in the product is zero.

In the following claim we show that, under some conditions on α, β, the second property of axis-evasiveness is met by the construction above. We mention already here that the third condition in Definition 35 is not met by the construction as is (regardless of the choice of α, β), and we will alter it afterwards to meet that property as well.

▷ Claim 57. Assume that $\langle\beta\rangle \cap \ell_\alpha = \langle\beta\rangle \cap \ell_{\alpha^{-1}} = \emptyset$ and that $\langle\beta\rangle \cap \mathbb{F}_q = \{1\}$. Then, for every $\ell, \ell' \in \mathcal{X}$ (not necessarily distinct) it holds that $\left|\ell' \cap (\ell)\right| \le c$.

Proof. Recall that $(\gamma) = \{\beta\gamma, \ldots, \beta^c\gamma\}$. Thus,

$$\bigcup_{\gamma\in\ell} (\gamma) = \bigcup_{\gamma\in\ell}\bigcup_{i=1}^{c} \{\beta^i\gamma\} = \bigcup_{i=1}^{c} \beta^i\ell.$$

Therefore,

$$\ell' \cap (\ell) = \ell' \cap \bigcup_{\gamma\in\ell} (\gamma) = \bigcup_{i=1}^{c} (\ell' \cap \beta^i\ell). \tag{5.2}$$

Fix $i \in [c]$ and consider two cases. First, if ℓ is vertical, namely, $\ell = b + \alpha\mathbb{F}_q$ for some $b \in \mathbb{F}_q$, then $\beta^i\ell = \beta^i b + \alpha\beta^i\mathbb{F}_q$. Since, by assumption, $\langle\beta\rangle \cap \mathbb{F}_q = \{1\}$ we have that $\alpha\beta^i\mathbb{F}_q \ne \alpha\mathbb{F}_q$ and so the line $\beta^i\ell$ is not vertical. As, by assumption, $\langle\beta\rangle \cap \ell_{\alpha^{-1}} = \emptyset$, we have that $\alpha\beta^i \notin \mathbb{F}_q$ and so the line $\beta^i\ell$ is not horizontal either.

Second, consider the case that ℓ is horizontal $\ell = b\alpha + \mathbb{F}_q$ for some $b \in \mathbb{F}_q$. Then, $\beta^i\ell = b\alpha\beta^i + \beta^i\mathbb{F}_q$. Per our assumption that $\langle\beta\rangle \cap \ell_\alpha = \emptyset$, we have that $\beta^i\mathbb{F}_q \ne \alpha\mathbb{F}_q$ and so the line $\beta^i\ell$ is not vertical. As we assume $\langle\beta\rangle \cap \mathbb{F}_q = \{1\}$, we have that $\beta^i\mathbb{F}_q \ne \mathbb{F}_q$, and so the line $\beta^i\ell$ cannot be horizontal either. To summarize, we have that $\beta^i\ell \notin \mathcal{X}$. However, $\ell' \in \mathcal{X}$ and so $\beta^i\ell$ and ℓ' are two distinct lines. As such, the two lines intersect in at most one point. Equation (5.2) then yield $|\ell' \cap (\ell)| \le c$. ◁

Informal discussion regarding the third property

As mentioned above, the partition of $(\mathbb{F}_{q^2})^\times$ as defined above does not have the third property required for axis-evasiveness. Namely, there are $\gamma \in (\mathbb{F}_{q^2})^\times$ such that $[\gamma]$ intersects some axis-parallel line at more than one point. To get some idea on which equivalence classes

$[\gamma]$ are problematic, let us first ask when do $\gamma, \beta\gamma$ are on some common axis-parallel line. We first observe that two points $\delta, \varepsilon \in (\mathbb{F}_{q^2})^\times$ are on a common axis-parallel line if and only if $\delta - \varepsilon \in \{1, \alpha\}\mathbb{F}_q$. Thus, γ and $\beta\gamma$ are on the same axis-parallel line if and only if $\gamma - \beta\gamma = (1-\beta)\gamma \in \{1, \alpha\}\mathbb{F}_q$. This is equivalent to saying that γ is on one of the two lines through the origin with slopes $\frac{1}{1-\beta}$, $\frac{\alpha}{1-\beta}$.

More generally, $[\gamma]$ intersects with some axis-parallel line in more than one point if and only if $\beta^i\gamma - \beta^j\gamma \in \{1, \alpha\}\mathbb{F}_q$ for some $0 \le j < i \le c$. Equivalently, γ is on a line ℓ_δ with

$$\delta \in \left\{ \frac{1}{\beta^i - \beta^j}, \frac{\alpha}{\beta^i - \beta^j} \;\middle|\; 0 \le j < i \le c \right\}. \tag{5.3}$$

The key observation is that although there are a fair amount of "bad" points γ, they are all contained in a small number of lines. By "small" here we mean that the number is polynomial in c and is independent of q. Thus, the hope is that by redefining the partition on these few problematic lines we will not harm the previous analysis by much. Indeed, no matter how we alter the partition restricted to these lines, if we make sure none of them is axis-parallel (by requiring more properties from α, β) then each of these lines intersect an axis-parallel line at one point. As a result, the bound obtained in Claim 57 will deteriorate proportionally to the number of lines above.

The only small technical issue is that even if $\gamma \in \ell_\delta$ for some slope δ as above, it is not the case that $[\gamma] \subseteq \cup_\varepsilon \ell_\varepsilon$ where ε is taken from the set of slopes given by Equation (5.3). As we wish to alter the partition defined above, it would be cleaner to have all of the points in $[\gamma]$ of a problematic point γ contained in the set of points on which we redefine the partition. Thus, we "close" the set of slopes given by Equation (5.3) to multiplication by β.

Ending the informal discussion and returning to the formal analysis, we consider the set of slopes.

$$\Delta = \left\{ \frac{\beta^k}{\beta^i - \beta^j}, \frac{\alpha\beta^k}{\beta^i - \beta^j} \;\middle|\; 0 \le j < i \le c \text{ and } 0 \le k \le c \right\} \tag{5.4}$$

Further define the set of all points in $(\mathbb{F}_{q^2})^\times$ covered by the lines with slopes from Δ by

$$U = \bigcup_{\delta \in \Delta} \ell_\delta.$$

This definition of Δ indeed fixes the technical caveat discussed above, as the following claim states.

▷ Claim 58. For every $\gamma \in (\mathbb{F}_{q^2})^\times$ either $[\gamma] \subseteq U$ or $[\gamma] \cap U = \emptyset$.

Proof. If an element $\varepsilon \in U$ then $\varepsilon \in \ell_\delta$ for some $\delta \in \Delta$. Note that $\beta\varepsilon \in \ell_{\beta\delta}$ and that $\beta\delta \in \Delta$. Hence, $\beta\varepsilon \in U$. Therefore, $\varepsilon \in U \implies \varepsilon\langle\beta\rangle \subseteq U$. Assume now that $[\gamma] \cap U \neq \emptyset$, and take $\gamma\beta^i \in U$. By the above, $\gamma\beta^i\langle\beta\rangle \subseteq U$. The proof then follows as $\gamma\beta^i\langle\beta\rangle = \gamma\langle\beta\rangle = [\gamma]$. ◁

Define a new partition of $\mathbb{F}_{q^2}$ (including 0) which agrees with the one that is given by Equation (5.1) on $\mathbb{F}_{q^2}^\times \setminus U$. By Claim 58, this is well-defined. The new partition, restricted to U, is done as follows. Let $\delta_0 \in \Delta$ be an arbitrary element. Note that

$$U = \ell_{\delta_0} \cup \bigcup_{\delta \in \Delta \setminus \{\delta_0\}} (\ell_\delta \setminus \{0\})$$

is a disjoint union. To partition U, we partition ℓ_{δ_0} as well as each of $\ell_\delta \setminus \{0\}$ where $\delta \in \Delta \setminus \{\delta_0\}$ in an arbitrary way provided it has the least number of parts under the conditions that each part has size at most $c+1$. For ease of readability, we denote by $[\gamma]$ the class with respect to the new partition.

▷ Claim 59. Assume, on top of the assumptions of Claim 57 that for every $\delta \in \Delta$, $\ell_\delta \notin \mathcal{X}$. Then, the new partition defined above is $(c, 4c^2)$-axis evasive.

Proof. First, observe that by construction, every class intersects any axis-parallel line in at most one point. Indeed, classes that are outside of U have this property by the definition of U as can be easily verified (and discussed above). Moreover, by the way we redefined the partition restricted to U, every class that is a subset of U is also a subset of a line ℓ_δ for some $\delta \in \Delta$. As $\ell_\delta \notin \mathcal{X}$ by hypothesis, we have that the line and, as a result, the class it contains, intersects any axis-parallel line in at most one point. This establishes the third property of axis-evasiveness. The second property follows as, by construction, every part has size at most $c+1$.

Moving on to the second property, consider $\ell, \ell' \in \mathcal{X}$, not necessarily distinct. As outside of U the partition is defined as before, Claim 58 yields

$$\left|\ell' \cap \bigcup_{\gamma \in \ell \setminus U} (\gamma)\right| \le c. \tag{5.5}$$

Take $\gamma \in U \cap \ell$. Since, by construction $(\gamma) \subseteq \ell_\delta$ for some $\delta \in \Delta$, and since by hypothesis $\ell_\delta \notin \mathcal{X}$ we have that $|\ell' \cap \ell_\delta| = 1$ and $(\gamma) \cap \ell' \subseteq \ell_\delta \cap \ell'$.Therefore, $|(\gamma) \cap \ell'| \le 1$. Together with Equation (5.5) we get that $|\ell' \cap (\ell)| \le c + |U \cap \ell|$. Now, since $\ell \in \mathcal{X}$ and every line ℓ_δ with slope $\delta \in \Delta$ is not in $\mathcal{X}$ we have that $|\ell \cap \ell_\delta| = 1$. Thus, $|U \cap \ell| \le |\Delta|$ which implies $|\ell' \cap (\ell)| \le c + |\Delta|$.

To conclude the proof, we turn to bound $|\Delta|$. It is straightforward to give a bound of $O(c^3)$ though one can optimize the bound a bit. Indeed, with the notation of Equation (5.4), by multiplying by $\beta^{-\min(j,k)}$, one can rewrite

$$\Delta = \left\{ \frac{1}{\beta^i - \beta^j}, \frac{\alpha}{\beta^i - \beta^j} \;\middle|\; 0 < j < i \le c \right\} \bigcup \left\{ \frac{\beta^j}{\beta^i - 1}, \frac{\alpha\beta^j}{\beta^i - 1} \;\middle|\; 0 < i \le c,\, 0 \le j \le c \right\}. \tag{5.6}$$

Thus, $|\Delta| \le 3c^2$, and the proof follows. ◁

We summarize the discussion so far.

▶ **Proposition 60.** *Let $\mathbb{F}_q$ be finite field. Let $h(x) \in \mathbb{F}_q[x]$ be a degree 2 irreducible monic polynomial, and consider the field $\mathbb{F}_q[x]/\langle h(x) \rangle$ which we denote by $\mathbb{F}_{q^2}$. Let $\alpha, \beta \in \mathbb{F}_{q^2}$ be two elements satisfying:*

1. $h(\alpha) = 0$,
2. $\langle \beta \rangle \cap \mathbb{F}_q = \{1\}$,
3. $c + 1 = o(\beta) \le \sqrt{q}/10$,
4. $\langle \beta \rangle \cap \ell_\alpha = \langle \beta \rangle \cap \ell_{\alpha^{-1}} = \emptyset$,
5. $(\langle \beta \rangle - \langle \beta \rangle) \cap \mathbb{F}_q = \{0\}$,
6. $(\langle \beta \rangle - \langle \beta \rangle) \cap \ell_\alpha = (\langle \beta \rangle - \langle \beta \rangle) \cap \ell_{\alpha^{-1}} = \{0\}$.

Then, there exists a partition of $(\mathbb{F}_q)^2$ that is $(c, 4c^2)$-axis-evasive, where $c = o(\beta) - 1$. The number of parts in the partition is bounded above by $2q^2/(c+1)$.

To prove Proposition 60 we need the following easy claim.

▷ Claim 61. Let $\delta \in (\mathbb{F}_{q^2})^\times$ be such that $\delta \notin \mathbb{F}_q \cup \ell_\alpha$ then, $\ell_\delta \notin \mathcal{X}$.

Proof. Write $\delta = a + \alpha b$ with $a, b \in \mathbb{F}_q$. Then, $\ell_\delta = (a + \alpha b)\mathbb{F}_q$. Observe that if ℓ_δ is vertical then $a = 0$ and so $\delta \in \ell_\alpha$. Similarly, if ℓ_δ is horizontal then $b = 0$ implying $\delta \in \mathbb{F}_q$. ◁

Proof of Proposition 60. To bound the number of parts, recall that in the original partition, each part has size $c+1$. Moreover, in the altered partition we partition each line ℓ_δ with slope $\delta \in \Delta$ (excluding the origin from all but for one of the lines ℓ_{δ_0}) to parts of size $c+1$ each, except for possibly one part. As $|\Delta| \le 3c^2$, the number of parts it bounded by

$$\frac{q^2-1}{c+1} + |\Delta| \left(1 + \frac{q}{c+1}\right) \le \frac{q^2-1}{c+1} + 6cq \le \frac{2q^2}{c},$$

where the last inequality follows by our assumption that $o(\beta) \le \sqrt{q}/10$.

To conclude the proof of the proposition, it suffices to show that for every $\delta \in \Delta$ it holds that $\ell_\delta \notin \mathcal{X}$. By Claim 61, it suffices to prove that $\delta \notin \mathbb{F}_q \cup \ell_\alpha = \{1, \alpha\}\mathbb{F}_q$. There are two types of slopes $\delta \in \Delta$, according to whether they appear in the first or second set in Equation (5.6). The first kind is of the form

$$\delta = \frac{\alpha^k}{\beta^i - \beta^j},$$

with $0 < j < i \le c$ and $k \in \{0,1\}$. If $\delta \in \{1,\alpha\}\mathbb{F}_q$ then $\delta^{-1} \in \{1, \alpha^{-1}\}\mathbb{F}_q$ and so $\beta^i - \beta^j \in \{\alpha^k, \alpha^{k-1}\}\mathbb{F}_q$ in contradiction to our hypothesis. Consider now the other kind of slope

$$\delta = \frac{\alpha^k \beta^j}{\beta^i - 1}$$

where $0 < i \le c$, $0 \le j \le c$ and $k \in \{0,1\}$. If $\delta \in \{1,\alpha\}\mathbb{F}_q$ then $\delta^{-1} \in \{1,\alpha^{-1}\}\mathbb{F}_q$ and so $(\beta^i - 1)\beta^{-j} \in \{\alpha^k, \alpha^{k-1}\}\mathbb{F}_q$. Note that $(\beta^i - 1)\beta^{-j} = \beta^{i-j} - \beta^{-j} \in \langle\beta\rangle - \langle\beta\rangle$ and so we again get a contradiction. ◀

We are now ready to prove Theorem 47. For the sake of readability, we repeat its statement here.

▶ **Theorem 62.** *Let P be a set of size q, where q is an odd prime power. Let c be an even integer such that $c+1 \mid q+1$, and $c \le \sqrt{q}/10$. Then, there exists a $(c, 4c^2)$-axis evasive partition of P^2 with at most $2q^2/(c+1)$ parts.*

Proof. As above, we identify P^2 with $\mathbb{F}_{q^2}$. It is a well-known fact that the multiplicative group $(\mathbb{F}_{q^2})^\times$ is cyclic. A basic result in group theory states that a cyclic group has a (unique) subgroup of every given size which divides the group size. Now, $|(\mathbb{F}_{q^2})^\times| = q^2 - 1 = (q-1)(q+1)$. Thus, as $c+1 \mid q+1$, there exists a subgroup H of $(\mathbb{F}_{q^2})^\times$ of size $c+1$. The subgroup H is cyclic, being a subgroup of a cyclic group. Let β be a generator for H. We first prove that β satisfies those hypothesis of Proposition 60 that do not involve α, namely, conditions (2) and (5).

▷ Claim 63. $(\langle\beta\rangle - \langle\beta\rangle) \cap \mathbb{F}_q = \{0\}$ and $\langle\beta\rangle \cap \mathbb{F}_q = \{1\}$.

Proof. Assume towards a contradiction that $\beta^i - \beta^j \in \mathbb{F}_q$ for some $0 \le j < i \le c$. Since $x^q = x$ for every $x \in \mathbb{F}_q$, we get

$$\beta^i - \beta^j = \left(\beta^i - \beta^j\right)^q = \beta^{iq} - \beta^{jq},$$

where the last equality follows since q is divisible by the characteristic of the field. Recall that $o(\beta) = c+1 \mid q+1$ and so $\beta^{i(q+1)} = 1$, implying $\beta^{iq} = \beta^{-i}$. Thus,

$$\beta^i - \beta^j = \frac{1}{\beta^i} - \frac{1}{\beta^j} = \frac{\beta^j - \beta^i}{\beta^{i+j}}.$$

As $\beta^i \neq \beta^j$ the above equation implies $\beta^{i+j} = -1$, and so $-1 \in H$. Since q is odd, the characteristic of the field $\mathbb{F}_{q^2}$ is odd and so $o(-1) = 2$. Lagrange's Theorem then implies that $2 \mid |H| = c+1$, which stands in contradiction to c being even.

To prove that $\langle\beta\rangle \cap \mathbb{F}_q = \{1\}$, take β^i with $0 < i \leq c$. If $\beta^i \in \mathbb{F}_q$ then $\beta^{iq} = \beta^i$. On the other hand, we proved above that $\beta^{iq} = \beta^{-i}$, and so $\beta^i = \beta^{-i}$ implying $\beta^{2i} = 1$. Therefore, $o(\beta) = c+1 \mid 2i$, but this is impossible as $0 < i \leq c$ and, recall, c is even. $\lhd$

We proceed with the proof of Theorem 47 by finding $\alpha \in \mathbb{F}_{q^2}$ that, together with the already chosen β, satisfies the remaining conditions in the hypothesis of Proposition 60. Since $\mathbb{F}_{q^2}$ is a quadratic field extension of $\mathbb{F}_q$, every element $\gamma \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$ has degree 2. That is, the minimal polynomial h_γ of every such γ over $\mathbb{F}_q$ is of degree 2 (and can be made monic by dividing by the leading coefficient, if necessary). Indeed, $\deg(h_\gamma)$ cannot equal 1 as this would imply $\gamma \in \mathbb{F}_q$. On the other hand,

$$2 = [\mathbb{F}_{q^2} : \mathbb{F}_q] = [\mathbb{F}_{q^2} : \mathbb{F}_q(\gamma)][\mathbb{F}_q(\gamma) : \mathbb{F}_q] = [\mathbb{F}_{q^2} : \mathbb{F}_q(\gamma)] \deg(h_\gamma),$$

which shows that if $\deg(h_\gamma) \neq 1$ then $\deg(h_\gamma) = 2$.

Thus, condition (1) in the hypothesis of Proposition 60 holds for every element in $\mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Hence, to prove that all the remaining conditions in the hypothesis of Proposition 60 hold, it suffices to prove that there exists $\alpha \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$ which satisfies conditions (4) and (6). To this end, pick a set of "slopes" $\Delta' = \{\delta_1, \ldots, \delta_{q+1}\} \subseteq (\mathbb{F}_{q^2})^\times$ such that $(\mathbb{F}_{q^2})^\times$ is the disjoint union

$$(\mathbb{F}_{q^2})^\times = \bigcup_{\delta \in \Delta'} (\ell_\delta \setminus \{0\}).$$

For example, $\Delta' = \{a + \alpha \mid a \in \mathbb{F}_q\} \cup \{1\}$ will do. For $\delta \in (\mathbb{F}_{q^2})^\times$ let

$$I_\delta = |\langle\beta\rangle \cap \ell_\delta| + |(\langle\beta\rangle - \langle\beta\rangle) \cap (\ell_\delta \setminus \{0\})|.$$

Since the $\ell_\delta \setminus \{0\}$ with $\delta \in \Delta'$ are disjoint, $0 \notin \langle\beta\rangle$, and since $|\langle\beta\rangle| = c+1$ and $|\langle\beta\rangle - \langle\beta\rangle| \leq (c+1)^2$, we have that

$$\mathop{\mathbf{E}}_{\delta}[I_\delta] \leq \frac{(c+1)^2 + (c+1)}{q+1} \leq \frac{2(c+1)^2}{q+1},$$

where δ is sampled uniformly from Δ'. By Markov's inequality, for at least $3/4$ of the elements $\delta \in \Delta'$ it holds that

$$|I_\delta| \leq \frac{8(c+1)^2}{q+1}.$$

Note that $(\mathbb{F}_{q^2})^\times$ is also a disjoint union of $\{\ell_{\delta^{-1}} \setminus \{0\} \mid \delta \in \Delta\}$. Thus, using the same argument as above, we get that for at least $1/2$ the elements $\delta \in \Delta'$, both $|I_\delta|$ and $|I_{\delta^{-1}}|$ are bounded by $8(c+1)^2/(q+1)$. But, as $c \leq \sqrt{q}/10$, this bound is strictly smaller than 1, implying that $I_i = I_{q+1-i} = 0$. That is, at least half the elements $\delta \in \Delta'$ satisfy conditions (4) and (6). Take α to be any of these elements. To conclude, we found α and β for which all the conditions in the hypothesis of Proposition 60 are met, and the proof follows. ◀

6 Query-efficient distance amplification

In this section we construct our query-efficient distance amplification procedure. We start by giving a somewhat more formal definition of locally decodable codes (compared to Definition 1) or, more precisely, a more formal definition of their non-adaptive counterparts. Recall that, informally, these are LDC in which the joint distribution of queries depends solely on the index one wishes to decode and is independent of the received word. By inspection, it is our understanding that the AEL distance amplification procedure also requires non-adaptivity.

▶ **Definition 64** (Locally decodable codes). *Let (C, Q, R) be a tuple of functions*

$$\begin{aligned} &C : \Sigma_{\mathsf{in}}^k \to \Sigma_{\mathsf{out}}^n, \\ &Q : [k] \times \{0,1\}^r \to [n]^q, \\ &R : [k] \times \Sigma_{\mathsf{out}}^q \times \{0,1\}^r \to \Sigma_{\mathsf{in}}. \end{aligned}$$

Define

$$D : [k] \times \Sigma_{\mathsf{out}}^n \times \{0,1\}^r \to \Sigma_{\mathsf{in}}$$

as follows. For $v \in [k]$, $y \in \Sigma_{\mathsf{out}}^n$, and $s \in \{0,1\}^r$, let

$$\begin{aligned} Q(v, s) &= (u_1, \dots, u_q), \\ D(v, y, s) &= R(v, y_{u_1}, \dots, y_{u_q}, s). \end{aligned}$$

*The tuple (C, Q, R) is called a (q, δ, ε)-*locally decodable code *(or (q, δ, ε)-LDC for short) if the following holds. For every $v \in [k]$, $x \in \Sigma_{\mathsf{in}}^k$, and $y \in \Sigma_{\mathsf{out}}^n$ for which $\mathsf{dist}(y, C(x)) \le \delta$, it holds that*

$$\Pr_{s \sim U_r}[D(v, y, s) = x_v] \ge 1 - \varepsilon.$$

We call the function C the encoding function*, Q the* querying function*, and R the* reconstruction function*. The induced function D is called the* decoding function*. The parameters k, n are referred to as the* message length *and the* block length*, respectively. The sets $\Sigma_{\mathsf{in}}, \Sigma_{\mathsf{out}}$ are called the input alphabet and output alphabet, respectively. We will be interested mostly in locally decodable codes in which $\Sigma_{\mathsf{in}} = \Sigma_{\mathsf{out}}$ in which case we refer to both as the* alphabet *of the code. The parameter r is called the* randomness complexity *of the LDC. We say the LDC is* explicit *if all three functions C, Q, R are polynomial-time computable. Note that then the decoding function D is also polynomial-time computable.*

6.1 The distance amplification procedure

In this section we present our query-efficient distance amplification procedure. We start by describing the building blocks we use and specify their parameters.

Building blocks

- For $i = 1, 2$ let (C_i, Q_i, R_i) be a $(q_i, \delta_i, \varepsilon_i)$-LDC with message length k_i and block length n_i over the same alphabet Σ. We denote the rate k_i/n_i of C_i by ρ_i.
- Let (C_3, Q_3, R_3) be a family of $(q_3(k_3), \delta_3(k_3), \varepsilon_3(k_3))$-LDC having rate $\rho_3(k_3)$ for message length k_3. The code C_3 is also over the alphabet Σ. We will always work with functions $q_3, \delta_3, \varepsilon_3, \rho_3$ that are monotone. More precisely, q_3 and ρ_3 are non-decreasing and δ_3, ε_3

are non-increasing. We sometimes write $q_3, \delta_3, \varepsilon_3, \rho_3$ without mentioning explicitly the message length, and by that refer to the largest k_3 considered in the construction for q_3, δ_3 and the smallest k_3 when considering ε_3, ρ_3. In any case, we assume (mostly for simplicity) that $\rho_3(k_3) \geq 1/2$ for all k_3.

- Set $\ell = n_1/k_2$. Let $G = (L, R, E)$ be a $(\delta_2/2, \delta_1)$-sampler with $|L| = \ell$ and $|R| = r$. Assume G is left-regular with left-degree $d = n_2$. Assume further that every right-vertex v of G has degree $\deg(v) \in [D/2, 2D]$, where D is the average right degree $D = \ell d/r = n_1/(r\rho_2)$.

How to think of the parameters?

We think of C_1 as the code whose distance δ_1 we wish to amplify. Typically, the code C_2 has a much shorter message length $n_2 \ll n_1$. In all applications in this paper we take δ_2 to be either constant or slightly sub constant in n_1. The code C_3 has a larger block length than C_2 and, depending on the application, it has either a somewhat smaller or much smaller message length than n_1. We typically take $\delta_3 \approx \delta_2$. The rates of all three codes is taken to be constant and even close to one. Note that we take C_3 to be a family of codes, whereas C_1 and C_2 are codes with predetermined message lengths. The reason is that the sampler G is not necessarily right-regular, and in the construction, we associate codes from C_3 with the right vertices of G. Recall, though that the ratio of largest to smallest right-degree is bounded by 4, so that is a minor technicality.

To describe the LDC that is composed of these building blocks, we need to specify the encoding function, querying function and reconstruction function. We start by describing the encoding function.

The encoding function

Let $n = \sum_{v \in R} n_v$ where n_v is the block length of the code from the family C_3 having message length $k_v = \deg(v)$. We define the function $C : \Sigma^{k_1} \to \Sigma^n$ as follows. Let $x \in \Sigma^{k_1}$.

1. Compute $y = C_1(x) \in \Sigma^{n_1}$.
2. Partition y to $y = y^{(1)} \circ \cdots \circ y^{(\ell)}$ consecutive blocks, each of length k_2. Recall that, indeed, $n_1 = \ell k_2$.
3. For every $u \in [\ell]$ compute $z^{(u)} = C_2(y^{(u)}) \in \Sigma^{n_2}$.
4. For every $v \in [r]$ and $j \in [\deg(v)]$ let $(u, j') = \Gamma(v, j) \in [\ell] \times [n_2]$. Define the string $w^{(v)} \in \Sigma^{\deg(v)} = \Sigma^{k_v}$ as follows: for $j \in [\deg(v)]$, $\left(w^{(v)}\right)_j = \left(z^{(u)}\right)_{j'}$.
5. For every $v \in [r]$ compute $t^{(v)} = C_3(w^{(v)}) \in \Sigma^{n_v}$.
6. The output of the encoding function on input x is then defined by $C(x) = t^{(1)} \circ \cdots \circ t^{(r)} \in \Sigma^n$, where as usual we identify R with $[r]$.

By the construction of the encoding function, the message length and block length of the resulted code are k_1 and n, respectively. From here on we denote $k = k_1$.

The querying function

We denote the randomness complexity of C_1, C_2, C_3 by r_1, r_2, r_3, respectively. The randomness complexity of the resulting querying function will be $r = r_1 + r_2 + r_3$, and the query complexity will be $q \leq q_1 q_2 q_3$, where q_3 is taken to be the maximum query complexity taken over all right vertices. We turn to define the querying function $Q : [k] \times \{0,1\}^r \to [n]^q$ as follows. On inputs $p \in [k], s \in \{0,1\}^r$ we proceed as follows.

1. Partition $s = s_1 \circ s_2 \circ s_3$ where $|s_1| = r_1$, $|s_2| = r_2$, $|s_3| = r_3$.
2. Compute $(a_1, \dots, a_{q_1}) = Q_1(p, s_1) \in [n_1]^{q_1}$.
3. For $i = 1, \dots, q_1$
    a. Set $u_i = \lceil a_i / k_2 \rceil$ and $b_i = 1 + ((a_i - 1) \bmod k_2)$. Informally, u_i is the "bucket" in which a_i resides and b_i is its location within the bucket. Note that we start the counting from 1 rather than 0, hence the slightly annoying addition and subtraction by 1 in the definition of b_i.
    b. Compute $(t_1^{(i)}, \dots, t_{q_2}^{(i)}) = Q_2(b_i, s_2) \in [n_2]^{q_2}$.
    c. For $j = 1, \dots, q_2$
        i. Let $(v^{(i,j)}, \hat{t}_j^{(i)}) = \Gamma(u_i, t_j^{(i)}) \in [r] \times [k_{v^{(i,j)}}]$.
        ii. Compute $(e_1^{(i,j)}, \dots, e_{q_3}^{(i,j)}) = Q_3(\hat{t}_j^{(i)}, s_3) \in [n_{v^{(i,j)}}]^{q_3}$.
        iii. As before, we endow the right vertices of the sampler in a fixed (arbitrary) order by identifying R with $[r]$. For $h = 1, \dots, q_3$ set $c^{(i,j,h)}$ to be the absolute location of $e_h^{(i,j)}$ in the ordering of R. That is, $c^{(i,j,h)} = e_h^{(i,j)} + \sum_{v < v^{(i,j)}} n_v$.
4. The result is then given by $Q(p, s) = (c^{(i,j,h)})_{(i,j,h) \in [q_1] \times [q_2] \times [q_3]}$.

Note that, indeed, the query complexity q of the querying function defined above is at most $q_1 q_2 q_3$ where, recall, $q_3 = q_3(2D)$. From here on we identify $[q]$ with $[q_1] \times [q_2] \times [q_3]$.

The reconstruction procedure

We define the reconstruction procedure $R : [k] \times \Sigma^q \times \{0,1\}^r \to \Sigma$ as follows. On inputs $p \in [k]$, $\sigma = (\sigma^{(i,j,h)})_{(i,j,h) \in [q_1] \times [q_2] \times [q_3]} \in \Sigma^q$, and $s \in \{0,1\}^r$, we proceed as follows.

1. Partition $s = s_1 \circ s_2 \circ s_3$ where $|s_1| = r_1$, $|s_2| = r_2$, $|s_3| = r_3$ as in the querying function.
2. For $i = 1, \dots, q_1$
    a. For $j = 1, \dots, q_2$
        i. Denote $(z_1, \dots, z_{q_3}) = (\sigma^{(i,j,1)}, \dots, \sigma^{(i,j,q_3)})$.
        ii. Compute $y_j^{(i)} = R_3(\hat{t}_j^{(i)}, z_1, \dots, z_{q_3}, s_3)$, where $\hat{t}_j^{(i)} = \hat{t}_j^{(i)}(p, s)$ as defined in the querying function.
    b. Set $x_i = R_2(b_i, y_1^{(i)}, \dots, y_{q_2}^{(i)}, s_2)$ where $b_i = b_i(p, s)$ as defined in the querying function.
3. The output is then given by $R(p, \sigma, s) = R_1(p, x_1, \dots, x_{q_1}, s_1)$.

6.2 Analysis

In this section we analyze the LDC obtained above. We prove

▶ **Proposition 65.** *With the notation of the previous section, C is a (q, δ, ε)-LDC, where*

$$q \leq q_1 q_2 q_3,$$
$$\delta \geq \frac{\delta_2 \delta_3}{16},$$
$$\varepsilon \leq \varepsilon_1 + (\varepsilon_2 + \varepsilon_3) n.$$

Furthermore, C has rate $\rho_1 \rho_2 \rho_3$, where ρ_1, ρ_2 are as defined in the building blocks paragraph, and per our convention set above, $\rho_3 = \rho_3(D/2)$.

Remark regarding the distance

Note that the distance δ of the resulted code C is independent of δ_1, the poor distance of C_1 we set out to amplify. This is the key feature of the AEL distance amplification procedure (which our variant above, of course, maintains). It is yet another instance of a general strategy in pseudo-randomness that combines objects in such a way that the resulted object enjoys the upsides of the different parts and avoid their shortcomings. The Zig-Zag product is another classic example. But, of course, δ_1 has some effect on the resulted code. The effect δ_1 has on the code is via the query complexity. Indeed, as the analysis will show, the smaller δ_1 is (i.e., the weaker the guarantee we have on the distance of C_1), the larger $k_2 = k_2(\delta_1)$ and $k_3 = k_3(\delta_1)$ must be, with a far stronger effect on k_3. More quantitatively, roughly speaking, by taking a sufficiently good sampler (e.g., the one that is given by Theorem 15), $k_2 \approx \operatorname{poly}\log(1/\delta_1)$ and $k_3 \approx \operatorname{poly}(1/\delta)$. This, in turn, effects the query complexities $q_2 = q_2(k_2)$ and $q_3 = q_3(k_3)$.

Proof. That the query complexity is $q \le q_1 q_2 q_3$ readily follows by the querying function, where recall that per our convention $q_3 = q_3(2D)$. To analyze the rate, recall that ρ_3 is a non-decreasing function. Further, our convention dictates that by writing ρ_3 without explicitly mentioning the message length, we refer to ρ_3 applied with the smallest message length taken by the construction, namely, $\rho_3 = \rho_3(D/2)$. Thus,

$$n = \sum_{v \in R} n_v = \sum_{v \in R} \frac{k_v}{\rho_3(k_v)} \le \frac{1}{\rho_3} \sum_{v \in R} k_v = \frac{\ell n_2}{\rho_3} = \frac{n_1}{\rho_2 \rho_3}.$$

Recall that $k = k_1 = \rho_1 n_1$ which shows that $\rho = k/n \ge \rho_1 \rho_2 \rho_3$.

We turn to analyze the distance δ and error ε. Let $x \in \Sigma^k$ and let $\widetilde{C}(x) \in \Sigma^n$ be such that $\mathsf{dist}(\widetilde{C}(x), C(x)) \le \delta$. Define the set of "errors", namely, the disagreements between $C(x)$ and $\widetilde{C}(x)$ by

$$B = \{i \in [n] \mid \widetilde{C}(x)_i \ne C(x)_i\}.$$

By assumption, $\mu(B) \le \delta$. The error set B induces errors "backwards" throughout the construction. We proceed by analyzing these induced errors. Recall that, in the encoding function, we defined for each $v \in [r]$ an element $t^{(v)} = t^{(v)}(x) \in \Sigma^{n_v}$. Partition $\widetilde{C}(x)$ to r substrings $\widetilde{C}(x) = \widetilde{t}^{(1)} \circ \cdots \circ \widetilde{t}^{(r)}$, where $\widetilde{t}^{(v)}$ has length n_v, and define the set

$$B_t = \left\{ v \in [r] \mid \mathsf{dist}\left(t^{(v)}, \widetilde{t}^{(v)}\right) \ge \delta_3 \right\}.$$

Informally, $v \in B_t$ if the adversary has introduced too many errors on the respective block to allow for correct decoding via D_3.

▷ Claim 66. $\mu(B_t) \le 8\delta/\delta_3$.

Proof. For $v \in R$ let $e_v = \mathsf{dist}(t^{(v)}, \widetilde{t}^{(v)})$. We have that $\sum_{v \in R} e_v n_v \le \delta n$. On the other hand,

$$\sum_{v \in R} e_v n_v \ge \delta_3 \sum_{v \in B_t} n_v \ge \frac{\delta_3 D |B_t|}{2}.$$

But, per our assumption that $\rho_3 \ge 1/2$, and since $k_v \le 2D$ for all $v \in R$,

$$n = \sum_{v \in R} n_v \le 2 \sum_{v \in R} k_v \le 4Dr.$$

The proof follows by the above three inequalities. ◁

For convenience we also denote $B_w = B_t$. Next, we define

$$B_z = \{u \in [\ell] \mid |\Gamma(u) \cap B_w| \geq \delta_2 n_2\}. \tag{6.1}$$

▷ Claim 67. $\mu(B_z) \leq \delta_1$.

Proof. By Claim 66 and by our assumption on δ,

$$\mu(B_w) \leq \frac{8\delta}{\delta_3} = \frac{\delta_2}{2}.$$

Recall that G is a $(\delta_2/2, \delta_1)$-sampler. Thus, at most δ_1-fraction of the left vertices $u \in [\ell]$ satisfy $\mu(\Gamma(u) \cap B_w) \geq \mu(B_w) + \delta_2/2$. The proof then follows since $\mu(B_w) \leq \delta_2/2$. ◁

Lastly, define

$$B_y = \left\{a \in [n_1] \;\middle|\; \left\lceil \frac{a}{k_2} \right\rceil \in B_z \right\}.$$

For $v \in [r]$, $b \in [k_v]$ we define the function $\widetilde{w}_b^{(v)} : \{0,1\}^{r_3} \to \Sigma$ as follows: on input $s_3 \in \{0,1\}^{r_3}$

$$\widetilde{w}_b^{(v)}(s_3) = D_3(b, \widetilde{t}^{(v)}, s_3).$$

▷ Claim 68. There exists a set $\mathcal{E}_3 \subseteq \{0,1\}^{r_3}$ with $\mu(\mathcal{E}_3) \leq \varepsilon_3 n$ such that for every $s_3 \in \{0,1\}^{r_3} \setminus \mathcal{E}_3$, $v \in [r] \setminus B_t$, and $b \in [k_3]$ it holds that $\widetilde{w}_b^{(v)}(s_3) = w_b^{(v)}$.

Proof. Fix $v \in [r] \setminus B_t$. By the definition of B_t, one has that $\mathsf{dist}\left(t^{(v)}, \widetilde{t}^{(v)}\right) \leq \delta_3$. By the encoding function, $t^{(v)} = C_3(w^{(v)})$. Therefore, for every $b \in [k_3]$,

$$\Pr_{s_3 \sim U_{r_3}}\left[D_3(b, \widetilde{t}^{(v)}, s_3) \neq w_b^{(v)}\right] \leq \varepsilon_3.$$

The proof then follows by taking the union bound over all $v \in [r] \setminus B_t$ and $b \in [k_v]$ as indeed $\sum k_v \leq n$. ◁

For $(u, j) \in [\ell] \times [n_2]$ we define the function $\widetilde{z}_j^{(u)} : \{0,1\}^{r_3} \to \Sigma$ as follows. For $s_3 \in \{0,1\}^{r_3}$ we have $\widetilde{z}_j^{(u)}(s_3) = \widetilde{w}_{j'}^{(v)}(s_3)$, where $(v, j') = \Gamma(u, j) \in [r] \times [k_v]$. Further define the function $\widetilde{z}^{(u)} : \{0,1\}^{r_3} \to \Sigma^{n_2}$ by

$$\widetilde{z}^{(u)}(s_3) = \widetilde{z}_1^{(u)}(s_3) \circ \cdots \circ \widetilde{z}_{n_2}^{(u)}(s_3).$$

▷ Claim 69. For every $u \notin B_z$ and $s_3 \in \{0,1\}^{r_3} \setminus \mathcal{E}_3$ it holds that

$$\mathsf{dist}\left(\widetilde{z}^{(u)}(s_3), z^{(u)}\right) \leq \delta_2.$$

Proof. Fix $s_3 \in \{0,1\}^{r_3} \setminus \mathcal{E}_3$ and consider any $u \in [\ell] \setminus B_z$. By the encoding function, for every $j \in [n_2]$ it holds that $z_j^{(u)} = w_{j'}^{(v)}$, where $(v, j') = \Gamma(u, j)$. As $v \notin B_z$, at most $\delta_2 n_2$ of $j \in [n_2]$ satisfy $v \in B_w$. For every other j,

$$\widetilde{z}_j^{(u)} = \widetilde{w}_{j'}^{(v)}(s_3) = w_{j'}^{(v)} = z_j^{(u)},$$

proving the claim. ◁

For $u \in [\ell], a \in [k_2]$ we define the function $\widetilde{y}_a^{(u)} : \{0,1\}^{r_2} \times \{0,1\}^{r_3} \to \Sigma$ as follows. On $(s_2, s_3) \in \{0,1\}^{r_2} \times \{0,1\}^{r_3}$,

$$\widetilde{y}_a^{(u)}(s_2, s_3) = D_2(a, \widetilde{z}^{(u)}(s_3), s_2).$$

▷ Claim 70. There exists a set $\mathcal{E}_2 \subseteq \{0,1\}^{r_2}$ with $\mu(\mathcal{E}_2) \le \varepsilon_2 n$ such that for every $u \in [\ell] \setminus B_z$, $a \in [k_2]$, and $(s_2, s_3) \in (\{0,1\}^{r_2} \setminus \mathcal{E}_2) \times (\{0,1\}^{r_3} \setminus \mathcal{E}_3)$ it holds that $\widetilde{y}_a^{(u)}(s_2, s_3) = y_a^{(u)}$.

Proof. Fix $u \in [\ell] \setminus B_z$. By the encoding function $z^{(u)} = C_2(y^{(u)})$. Recall that

$$\widetilde{y}_a^{(u)}(s_2, s_3) = D_2(a, \widetilde{z}^{(u)}(s_3), s_2).$$

As $s_3 \notin \mathcal{E}_3$, $u \notin B_z$, Claim 69 implies $\mathsf{dist}(\widetilde{z}^{(u)}(s_3), z^{(u)}) \le \delta_2$. Therefore

$$\Pr_{s_2 \sim U_{r_2}} \left[D_2(a, \widetilde{z}^{(u)}(s_3), s_2) \neq y_a^{(u)} \right] \le \varepsilon_2.$$

The proof then follows by taking the union bound over all $a \in [k_2]$ and $u \in [\ell] \setminus B_z$, and noting that $k_2 \ell = n_1 \le n$. ◁

▷ Claim 71. For every $(s_2, s_3) \in (\{0,1\}^{r_2} \setminus \mathcal{E}_2) \times (\{0,1\}^{r_3} \setminus \mathcal{E}_3)$, it holds that

$$\mathsf{dist}(\widetilde{y}(s_2, s_3), y) \le \delta_1,$$

where $\widetilde{y}(s_2, s_3)$ is the concatenation of the k_2-length strings $\left(\widetilde{y}^{(u)}(s_2, s_3) \mid u \in [\ell]\right)$.

Proof. Note that by Claim 70, the projection of the two strings $\widetilde{y}(s_2, s_3)$, y to a block corresponding to $u \notin B_z$ are in full agreement. The proof then follows by Claim 67. ◁

We now conclude the proof of Proposition 65. Let $p \in [k]$, by Claim 71, for every $(s_2, s_3) \in (\{0,1\}^{r_2} \setminus \mathcal{E}_2) \times (\{0,1\}^{r_3} \setminus \mathcal{E}_3)$, we have that $\mathsf{dist}(\widetilde{y}(s_2, s_3), y) \le \delta_1$. Since by the encoding function $y = C_1(x)$, it holds

$$\Pr_{s_1 \sim U_{r_1}} [D_1(p, \widetilde{y}(s_2, s_3), s_1) \neq x_p] \le \varepsilon_1.$$

The proof then follows since $\mu(\mathcal{E}_2) \le \varepsilon_2 n$ and $\mu(\mathcal{E}_3) \le \varepsilon_3 n$. ◀

6.2.1 Proof of Theorem 5

In this short section we prove Theorem 5. We focus on the version that is based on non-explicit samplers, yielding non-explicit reductions. The explicit reduction, which entails a bit more technical work, is deferred to Section 6.3 and Section 6.6. We choose to focus on the non-explicit version first because we believe that understanding LDC in the information-theoretic level is, at present, a deeper and more urgent problem than the question of explicitness. Also, the parameters are easier to work with. For the information-theoretic version, we make use of the sampler that is given by Theorem 15. From here on we refer to the constant $c_{\mathsf{samp}} \ge 1$ that appears in that theorem.

▶ **Theorem 72.** *Let C be a block-length-n $(q, \delta, 1/5)$-LDC over alphabet Σ having a constant rate. Let C' be a family of asymptotically good $(q'_n, \delta', 1/5)$-LDC, where q'_n is the query complexity when the code from the family is taken with block length n. Then, there exists an asymptotically good LDC over Σ, with constant error, having block length $\Theta(n)$ and query complexity*

$$q_{\text{new}} = O\left(q \cdot q'_{O(1/\delta)} \log(1/\delta) \log n\right). \tag{6.2}$$

Proof. Take C_1 to be the code C in the hypothesis of the theorem, namely, a code with block length $n_1 = n$ and distance $\delta_1 = \delta$. Recall that in the distance amplification procedure from Section 6.1, we make use of a $(\delta_2/2, \delta_1)$ sampler $G = ([\ell], [r], E)$ with $\ell = n_1/k_2$ and left-degree $d = n_2$. For the proof, we will instantiate the distance amplification procedure with the sampler that is given by Theorem 15. We take C_2 to be an asymptotically good code over Σ set with block length

$$n_2 = c_{\mathsf{samp}} \cdot \frac{\log(1/\delta_1)}{(\delta_2/2)^2} = O(\log(1/\delta)),$$

where δ_2 is the (constant) distance of C_2, having rate at least $1/2$. Note that this choice of parameters is as required by Theorem 15 from the left degree of the sampler. Clearly, C_2 has query complexity $O(\log(1/\delta))$ and error $\varepsilon_2 = 0$. As for the degree D_v of any given right vertex v of the sampler, note that the average right degree is

$$D = \frac{\ell d}{r} = \frac{d}{\delta \log(1/\delta)} = \frac{4c_{\mathsf{samp}}}{\delta_1 \delta_2^2} = \Theta\left(\frac{1}{\delta}\right).$$

Recall that, by Theorem 15, $D_v \in [D/2, 2D]$. For every length in this range, we take a code from the family C' having the required message length. We would like take the family of codes C_3 to be C' though we must reduce the error first. Indeed, note that the error ε of the code obtained by Proposition 65 is $\varepsilon_1 + n(\varepsilon_2 + \varepsilon_3)$. As mentioned in the introduction, one can reduce the error from $1/5$ to $1/(10n)$ by applying the decoding procedure for $c \log n$ times, where c is some large enough constant, and output the symbol according to plurality. This has no effect on the rate or distance of C', and has a multiplicative $O(\log n)$ cost in query complexity. That is, the query complexity of C_3 is $O(q'_{O(1/\delta_1)} \log n)$. The proof then readily follows by Proposition 65. ◀

6.2.1.1 Improving the query complexity further given low-error LDC

We remark that, if C' has error $O(1/n)$ to begin with, n being the block length of C, then one can skip the error reduction in the proof of Theorem 72, and get a slightly better query complexity. Indeed, this will save the $\log n$ factor in Equation (6.2). Moreover, observe that C_2 can be taken to be an LDC as well, rather than a standard code, which will reduce its deterioration on the query complexity from $O(\log(1/\delta))$ to $q'_{O(\log(1/\delta))}$. However, for that, one need the error of C_2 to be $O(1/n)$ as well. Assuming one can obtain such low-error LDC (note that an error of $1/n$ is at least exponentially-small in the length of C_2 since $\delta > 1/n$), the query complexity can be improved further to

$$q_{\text{new}} \leq q \cdot q'_{O(1/\delta)} q'_{O(\log(1/\delta))}.$$

We conclude this section by instantiating Theorem 72 with C' taken to be the state-of-the-art construction of asymptotically good LDC.

▶ **Theorem 73** ([26]). *Let Σ be a finite alphabet. Then, there exist constants δ, ρ and an explicit infinite family of $(q_k, \delta, 1/5)$-LDC, k being the message length, having query complexity $q_k = 2^{O(\sqrt{\log(k) \log\log k})}$.*

Using it, one gets query complexity

$$q_{\text{new}} \leq q \log(n) \cdot 2^{O(\sqrt{\log(1/\delta) \cdot \log\log(1/\delta)})} = q \log(n) (1/\delta)^{o(1)}.$$

6.3 Relaxing the assumption on the sampler G

In the distance amplification procedure described in Section 6.1, the sampler G is assumed to be a left-regular $(\delta_2/2, \delta_1)$-sampler in which every right degree is in $[D/2, 2D]$. In order for the reduction to result in an explicit code, we want to be able to plug in an explicit sampler in the distance amplification procedure, for which the bounds on the right degree may not hold. We now describe how a sampler that does not satisfy this assumption can be used even so. The change to the construction is detailed as follows.

Modified construction

- For $i = 1, 2, 3$ let (C_i, Q_i, R_i) be as in Section 6.1. Assume further that $\delta_1 \leq \delta_2/8$.
- Set $\ell = n_1/k_2$. Let $G = (L, R, E)$ be a $(\delta_2/8, \delta_1)$-sampler with $|L| = \ell$ and $|R| = r$. Assume G is left-regular with left-degree $d = n_2$, and denote by $D = \frac{\ell d}{r}$ the average right degree (the right degrees may be arbitrary).
- The encoding function $C : \Sigma^{k_1} \to \Sigma^n$ is the same as in Section 6.1, but for the following change: if $v \in [r]$ has degree outside $[D/2, 2D]$ then discard it.
- The querying function is the same as in Section 6.1, but for the following change: if $v^{(i,j)}$ is a vertex with degree not in $[D/2, 2D]$, then set $(c^{(i,j,h)})_{h \in [q_3]}$ to be an empty tuple.
- The reconstruction procedure is the same as in Section 6.1, but for the following change: if i, j are such that $v^{(i,j)}$ is a vertex with degree not in $[D/2, 2D]$, then set $y_j^{(i)} = \perp$ (or, if one prefers to avoid the use of $\perp$, any $\sigma \in \Sigma$ can be used).

The amendments above have the effect that when encoding the blocks corresponding to right vertices, that are either too big or too small, the encoding discards such blocks and their contents, as if they were deleted. The querying function is changed so that whenever a location in these blocks needs to be queried, that query is skipped. The reconstruction procedure is accordingly changed so that whenever a location was not queried on the account of it residing in a block too big or too small, some arbitrary symbol (or $\perp$) is passed on instead. To analyze the altered distance-amplification procedure we start by proving two simple statements about samplers.

▶ **Lemma 74.** *Let $G = ([\ell], [r], E)$ be a left-regular (ε, δ)-sampler with average right-degree D. Assume $\delta \leq 1/4$. Then, G has at most $3\varepsilon r$ right vertices with degree less than $D/2$.*

Proof. Denote by d the left-degree of G. Define $A = \{v \in [r] \mid \deg(v) < D/2\}$. Since G is an (ε, δ) sampler, at least $(1-\delta)$ fraction of the left vertices have (at least) $(\frac{|A|}{r} - \varepsilon)d$ neighbors in A. Hence, A has at least $(1-\delta)\ell(\frac{|A|}{r} - \varepsilon)d$ edges entering it. Therefore, it must hold that

$$\frac{(1-\delta)\ell\left(\frac{|A|}{r} - \varepsilon\right)d}{|A|} < \frac{D}{2}.$$

As the average right degree is $D = \frac{\ell d}{r}$, and since by assumption $\delta \leq 1/4$, we conclude that the average right-degree of A is at least

$$\frac{(1-\delta)\ell\left(\frac{|A|}{r} - \varepsilon\right)d}{|A|} = (1-\delta)D\left(1 - \frac{r\varepsilon}{|A|}\right) \geq \frac{3D}{4}\left(1 - \frac{r\varepsilon}{|A|}\right).$$

By the above two equation it follows that $|A| < 3\varepsilon r$. ◀

▶ **Lemma 75.** *Let $G = ([\ell], [r], E)$ be an (ε, δ)-sampler, which is d-left-regular and has average right-degree D. Assume $\varepsilon \geq \delta$. Then, G has at most $2\varepsilon r$ right vertices with degree larger than $2D$.*

Proof. Define $B = \{v \in [r] \mid \deg(v) > 2D\}$. At least $(1-\delta)$-fraction of the left vertices have at least $(1 - \frac{|B|}{r} - \varepsilon)d$ neighbors in $[r] \setminus B$, so $[r] \setminus B$ has at least $(1-\delta)\ell(1 - \frac{|B|}{r} - \varepsilon)d$ edges going into it. We therefore have that

$$2D|B| + (1-\delta)\ell\left(1 - \frac{|B|}{r} - \varepsilon\right)d \leq rD.$$

As $rD = \ell d$, it follows that

$$|B| \leq \left(\frac{\varepsilon + \delta - \delta\varepsilon}{1+\delta}\right) r \leq 2\varepsilon r. \qquad ◀$$

We now wish to state the correctness of the changed construction.

▶ **Proposition 76.** *The encoding function C of the modified construction is a (q, δ, ε)-LDC, where*

$$q \leq q_1 q_2 q_3,$$
$$\delta \geq \frac{\delta_2 \delta_3}{32},$$
$$\varepsilon \leq \varepsilon_1 + (\varepsilon_2 + \varepsilon_3)n.$$

Furthermore, C has rate $\rho_1\rho_2\rho_3$, where ρ_1, ρ_2 are as defined in the building blocks paragraph, and per our convention set above, $\rho_3 = \rho_3(D/2)$.

Proof. That the rate and query complexity are as stated is trivial, since the rate and query complexity can only be improved by this modification to the construction in which we discard some of the codeword symbols, and skip some of the queries. We now discuss the distance δ and error ε. Since the proof is almost identical to the proof of Proposition 65, we only state how to change the proof above to get a proof for the current proposition. Let

$$X = \{v \in R \mid \deg(v) \notin [D/2, 2D]\}$$

be the set of right vertices with "bad" degrees. Recall that these vertices are ignored by the modified construction. In particular, $n = \sum_{v \in R \setminus X} n_v$. The proof of Proposition 65 starts by defining the set

$$B = \{i \in [n] \mid \widetilde{C}(x)_i \neq C(x)_i\},$$

which is the set of "errors". It then goes on by defining another set, B_t, which is the set of "bad" right vertices, for which the adversary has introduced too many errors on the respective block. This is where we make a slight modification, ignoring the vertices in X. Formally, we define

$$B_t = \left\{v \in R \setminus X \mid \mathsf{dist}\left(t^{(v)}, \widetilde{t}^{(v)}\right) \geq \delta_3\right\}.$$

In the following claim we bound the density of B_t with respect to the set R (rather than with respect to $R \setminus X$).

▷ Claim 77. $\mu_R(B_t) \leq \frac{8\delta}{\delta_3}$.

Proof. The proof is similar to the proof of Claim 66 though it takes into account our modifications as described above. For $v \in R \setminus X$ let $e_v = \mathsf{dist}(t^{(v)}, \widetilde{t}^{(v)})$. We have that $\sum_{v \in R \setminus X} e_v n_v \leq \delta n$. On the other hand,

$$\sum_{v \in R \setminus X} e_v n_v \geq \delta_3 \sum_{v \in B_t} n_v \geq \frac{\delta_3 D |B_t|}{2},$$

where the last inequality follows as for every $v \in B_t \subseteq R \setminus X$ it holds that $\deg(v) \geq D/2$. We also have, per our assumption, that $\rho_3 \geq 1/2$, and since $k_v \leq 2D$ for all $v \in R \setminus X$,

$$n = \sum_{v \in R\setminus X} n_v \leq 2 \sum_{v \in R\setminus X} k_v \leq 4Dr.$$

The proof follows by the above three inequalities, ◁

As in Proposition 65, we also denote $B_w = B_t$. The definition of the set B_z is the same as in the proof of Proposition 65 with the modification that it "treats" the vertices in X as errors. Formally,

$$B_z = \{u \in [\ell] \mid |\Gamma(u) \cap (B_w \cup X)| \geq \delta_2 n_2\}, \tag{6.3}$$

▷ Claim 78. $\mu(B_z) \leq \delta_1$.

Proof. By Claim 77, $\mu_R(B_w) \leq \frac{8\delta}{\delta_3}$. Now, G is a $(\delta_2/8, \delta_1)$-sampler. Thus, by Lemma 74 and Lemma 75 (which are applicable as $\delta_1 \leq \delta_2/8$ per our assumption), $\mu_R(X) \leq \frac{5\delta_2}{8}$. Hence, the density of $B_w \cup X$ with respect to R is

$$\mu_R(B_w \cup X) \leq \frac{8\delta}{\delta_3} + \frac{5\delta_2}{8} \leq \frac{7\delta_2}{8},$$

where the last inequality holds per our assumption $\delta \leq \delta_2\delta_3/32$. Recall that G is a $(\delta_2/8, \delta_1)$-sampler. Thus, at most δ_1-fraction of the left vertices $u \in [\ell]$ satisfy

$$\mu_{\Gamma(u)}(\Gamma(u) \cap (B_w \cup X)) \geq \mu_R(B_w \cup X) + \frac{\delta_2}{8},$$

and the proof follows. ◁

The rest of the proof is identical to the proof of Proposition 65. ◀

6.4 Reduction to LDC with polynomially-small (and even smaller) distance

In this section we prove the following corollary of Proposition 65. We then deduce from it Corollary 6 and Corollary 7 from the introduction.

▶ **Corollary 79.** *There exists a universal constant c' such that the following holds. Let $c \geq 1$ be any constant. Let $\alpha : \mathbb{N} \to (0, 1)$, $\beta : \mathbb{N} \to (0, 1)$ be two monotone non-increasing functions that satisfy*

$$\alpha(n^{1.01}) \geq c'\beta(\log n) \cdot \log\log n. \tag{6.4}$$

Assume further that $\alpha(n) \leq 0.009$ and that $\beta(n) \leq 0.1$ for all $n \geq 1$. Assume there exists a family of $(q_\alpha(n), n^{-(1-\alpha(n))}, 1/5)$-LDC over alphabet Σ having rate $1 - \beta(n)$. Then, for every sufficiently large n there exists a $(q, \delta, 1/5)$-LDC on block length $m \in [n, n^{1.01}]$ [10] over Σ, where

$$q = (q_\alpha(n)\log n)^{O\left(\frac{\log\log n}{\alpha(n^{1.01})}\right)},$$
$$\rho = 1 - O\left(\frac{\beta(\log n)\log\log n}{\alpha(n^{1.01})}\right),$$
$$\delta = \beta(\log n)^{O\left(\frac{\log\log n}{\alpha(n^{1.01})}\right)}.$$

[10] The constant 1.01 in the exponent, which determines the density of lengths for which we can construct the stated codes, can be replaced by any constant strictly larger than 1, and even by $1 + o(1)$ for a "sufficiently large" $o(1)$. However, for ease of presentation, we stick with this fixed choice.

To prove Corollary 79, we prove the following claim. In its statement, we refer to the constant $c_{\mathsf{samp}} \geq 1$ that is given by Theorem 15.

▷ Claim 80. Let $\beta_2 < 1/2$. Assume there exists a $(q_{\text{in}}, \delta_{\text{in}}, \varepsilon_{\text{in}})$-LDC C_{in} over alphabet Σ for every message length $k_{\text{in}} \in [D/2, 2D]$ where

$$D = \frac{4c_{\mathsf{samp}} n^{1-\alpha(n)}}{\beta_2^6}, \tag{6.5}$$

having rate $\rho_{\text{in}} \geq 1/2$. Then, under the hypothesis of Corollary 79, there exists a $(q_{\text{out}}, \delta_{\text{out}}, \varepsilon_{\text{out}})$-LDC over Σ with block-length n having rate ρ_{out}, where

$$\begin{aligned}
\frac{q_{\text{out}}}{q_{\text{in}}} &\leq \frac{4c_{\mathsf{samp}} \log n}{\beta_2^6} \cdot q_\alpha(n), \\
\frac{\delta_{\text{out}}}{\delta_{\text{in}}} &\geq \frac{\beta_2^3}{16}, \\
\frac{\rho_{\text{out}}}{\rho_{\text{in}}} &\geq (1-\beta_2)\,(1-\beta(n))\,, \\
\varepsilon_{\text{out}} &\leq \frac{1}{5} + n\varepsilon_{\text{in}}.
\end{aligned}$$

Proof. Let C_1 be the LDC from the hypothesis of Corollary 79 taken with block length $n_1 = n$. Let C_2 be a code set with message length $k_2 = \frac{4c_{\mathsf{samp}} \log n}{\beta_2^6}$, over Σ having rate $1-\beta_2$ and distance $\delta_2 = \beta_2^3$. A code with such parameters exists, over any alphabet, by the Gilbert-Varshamov bound.

Recall that in the distance amplification procedure (Section 6.1), we make use of a $(\delta_2/2, \delta_1)$-sampler $G = ([\ell], [r], E)$ with $\ell = n_1/k_2$ and left-degree n_2. For the proof of the claim, we will instantiate the distance amplification procedure with the sampler that is given by Theorem 15. To be able to use this sampler, we must verify that the left-degree is indeed large enough with respect to the parameters of the sampler. As, in our case, the left degree is n_2, we need to verify that

$$n_2 \geq c_{\mathsf{samp}} \cdot \frac{\log(1/\delta_1)}{(\delta_2/2)^2} = \frac{4c_{\mathsf{samp}}(1-\alpha(n))\log n}{\beta_2^6}. \tag{6.6}$$

However,

$$\frac{4c_{\mathsf{samp}}(1-\alpha(n))\log n}{\beta_2^6} \leq \frac{4c_{\mathsf{samp}} \log n}{\beta_2^6} = k_2,$$

and so, Equation (6.6) holds.

As for the degree D_v of any given right vertex v of the sampler, we have by Theorem 15 that $D_v \in [D/2, 2D]$, where

$$D = \frac{\ell d}{r} = \frac{4c_{\mathsf{samp}} n^{1-\alpha(n)}}{\beta_2^6},$$

which equals to D as defined in Equation 6.5. Thus, we may use C_{in} as in the hypothesis of the claim. We are therefore in a position to apply Proposition 65. The assertions regarding the query complexity, distance and rate readily follow by Proposition 65. That the error is bounded as stated readily follows by noting that $\varepsilon_2 = 0$. ◁

It will be more convenient to have no error loss in the reduction that is given by Claim 80. This is easily achievable by amplifying the error of the input code before applying the previous claim.

▶ **Corollary 81.** *Let $\beta_2 < 1/2$. Assume there exists a $(q_{\text{in}}, \delta_{\text{in}}, 1/4)$-LDC C_{in} over alphabet Σ for every message length $k_{\text{in}} \in [D/2, 2D]$, where D is as in Equation (6.5), having rate $\rho_{\text{in}} \geq 1/2$. Then, under the hypothesis of Corollary 79, there exists a $(q_{\text{out}}, \delta_{\text{out}}, 1/4)$-LDC over Σ with block-length n having rate ρ_{out}, where*

$$\frac{q_{\text{out}}}{q_{\text{in}}} \leq \frac{100 c_{\mathsf{samp}} \log^2 n}{\beta_2^6} \cdot q_\alpha(n),$$
$$\frac{\delta_{\text{out}}}{\delta_{\text{in}}} \geq \frac{\beta_2^3}{16},$$
$$\frac{\rho_{\text{out}}}{\rho_{\text{in}}} \geq (1-\beta_2)(1-\beta(n)).$$

Proof. Let r be a parameter we set later on. Define the code C' to be the code C_{in} though with the following decoder. To decode C', apply the decoder of C_{in} for r times and return the symbol according to plurality. Clearly, the rate and distance remain intact. By a simple application of the Chernoff bound, one can show that the error of C' is $2^{-\Omega(r)}$. The query complexity of C' is then rq_{in}. Thus, by taking $r = c \log n$ for a sufficiently large constant c, we can get a code with error $1/n^2$. The query complexity is then increased by a multiplicative $O(\log n)$ factor. The proof then follows by applying Claim 80 to C'. ◀

With Corollary 81 we are ready to prove Corollary 79.

Proof of Corollary 79. The construction of the asserted code is obtained by devising a sequence of LDC $C'_0, C'_1, C'_2, \ldots$ where C'_0 is taken to be a code over Σ with block length

$$n_0 = 2\left(\frac{16 c_{\mathsf{samp}}}{\beta(\log n)^6}\right)^{8/\alpha(n^{1.01})}, \tag{6.7}$$

having rate $\rho_0 = 1 - \beta(\log n)$ and distance $\beta(\log n)^3$. A code with such parameters exists, over any alphabet, by the Gilbert-Varshamov bound. Clearly, as an LDC, this code has error $\varepsilon_0 = 0$ and query complexity n_0. For $t > 0$, the code C'_t is obtained by applying Corollary 81 with the code C'_{t-1} as C_{in} in the notations of the corollary and using $\beta_2 = \beta(\log n)$. Denote the message length and block length of C'_t by k_t and n_t, respectively. By construction, for every integer $t \geq 1$ such that $n_t \leq n^{1.01}$ we have that

$$k_{t-1} \leq \frac{8 c_{\mathsf{samp}} n_t^{1-\alpha(n_t)}}{\beta(\log n)^6} \leq \frac{8 c_{\mathsf{samp}} n_t^{1-\alpha(n^{1.01})}}{\beta(\log n)^6}, \tag{6.8}$$

where we used the fact that $\alpha(n)$ is non-increasing. By Corollary 81,

$$\rho_t = \frac{k_t}{n_t} \geq (1-\beta(\log n))^2 \rho_{t-1},$$

and so

$$\rho_t \geq (1-\beta(\log n))^{2t}\rho_0 = (1-\beta(\log n))^{2t+1}.$$

In particular, for every $t \leq \frac{1}{4\beta(\log n)}$ we get

$$\rho_t \geq (1-\beta(\log n))^{1+\frac{1}{2\beta(\log n)}} \geq \frac{1}{2}.$$

The the last inequality follows since the function $(1-x)^{1+\frac{1}{2x}} \geq \frac{1}{2}$ for all $x \leq 0.1$ and, recall, we assume that the function β is bounded above by 0.1. By Equation (6.8) we have that for every $t \leq \frac{1}{4\beta(\log n)}$,

$$n_{t-1} \leq 2k_{t-1} \leq \frac{16c_{\mathsf{samp}} n_t^{1-\alpha(n^{1.01})}}{\beta(\log n)^6}.$$

Thus,

$$n_t \geq \left(\frac{n_{t-1}\beta(\log n)^6}{16c_{\mathsf{samp}}} \right)^{\frac{1}{1-\alpha(n^{1.01})}}. \tag{6.9}$$

One can prove the following easy claim by induction.

▷ Claim 82. Let $(n_t)_{t\in\mathbb{N}}$ be a sequence of positive integers such that $n_t \geq (n_{t-1}/a)^b$ for some $a, b > 1$. Then, for every $t \geq 1$ we have that $n_t \geq (n_0/a^{h(b,t)})^{b^t}$, where $h(b,t) = \sum_{i=0}^{t-1} \frac{1}{b^i}$.

With the notation of Claim 82, we have

$$h\left(\frac{1}{1-\alpha(n^{1.01})}, t\right) = \sum_{i=0}^{t-1} (1-\alpha(n^{1.01}))^i \leq \frac{1}{\alpha(n^{1.01})}.$$

By applying Claim 82 with $a = 16c_{\mathsf{samp}}/\beta(\log n)^6$ and $b = \frac{1}{1-\alpha(n^{1.01})}$ we get that for every t such that $n_t \leq n^{1.01}$ it holds

$$n_t \geq \left(\frac{n_0}{\left(\frac{16c_{\mathsf{samp}}}{\beta(\log n)^6}\right)^{1/\alpha(n^{1.01})}} \right)^{\left(\frac{1}{1-\alpha(n^{1.01})}\right)^t} \geq 2^{\left(\frac{1}{1-\alpha(n^{1.01})}\right)^t},$$

where for the last equality we used our of n_0 given in Equation (6.7). We now wish to take t' to be the least integer for which the right hand side is larger or equal than n. However, we must make sure that such t' exists. Indeed, the above analysis only works for t such that both $n_t \leq n^{1.01}$ and $t \leq \frac{1}{4\beta(\log n)}$ holds. So, one must verify that there exists a $t' \leq \frac{1}{4\beta(\log n)}$ for which $n \leq n_{t'} \leq n^{1.01}$. To see this, recall that $k \in [D/2, 2D]$ where D is as given by Equation (6.5). Hence,

$$n_{t-1} \geq k_{t-1} \geq \frac{2c_{\mathsf{samp}} n_t^{1-\alpha(n)}}{\beta_2^6} \geq n_t^{1-\alpha(n^{1.01})},$$

Hence, if $n_{t-1} < n$ then

$$n_t < n^{\frac{1}{1-\alpha(n^{1.01})}} < n^{1.01},$$

where the last inequality follows as $\alpha(n_t) \leq 0.009$. Thus,

$$t' = \Theta\left(\frac{\log\log n}{\log\left(\frac{1}{1-\alpha(n^{1.01})}\right)} \right) = \Theta\left(\frac{\log\log n}{\alpha(n^{1.01})} \right),$$

and we can thus see that $t' \leq \frac{1}{4\beta(\log n)}$ per our assumption that is given by Equation (6.4).

It is easy to verify that the query complexity $q_{t'}$ of and distance $\delta_{t'}$ of $C'_{t'}$ are

$$\begin{aligned} q_{t'} &= \left(\frac{\log n}{\beta(\log n)}\right)^{\Theta(t')}, \\ \delta_{t'} &= \beta(\log n)^{\Theta(t')}. \end{aligned}$$

As for the rate,

$$\rho_{t'} \geq (1-\beta(\log n))^{\Theta(t')} = 1 - O\left(\frac{\beta(\log n)\log\log n}{\alpha(n^{1.01})}\right),$$

where the last equality follows by Equation (6.4). Finally, the error of $C'_{t'}$ can be reduced from $1/4$ to $1/5$ with no asymptotic overhead in query complexity, and so $C'_{t'}$ has all the asserted properties. ◀

6.4.1 Proofs of Corollary 6 and Corollary 7

In this short section prove Corollary 6 and Corollary 7.

Proof of Corollary 6. With the hypothesis of the corollary, we may apply Corollary 79 with $\alpha(n)$ and $\beta(n)$ in the notation of Corollary 79 set to $\alpha(n) = \min(\alpha, 0.009)$ and $\beta(n) = \frac{1}{\log^2 n}$ (and, in fact, taking $\beta(n) = \frac{c}{\log n}$ for sufficiently small constant $c > 0$ will do as well). Note that Equation (6.4) holds with this choice. Corollary 79 then yields a $(q_1, \delta_1, \varepsilon_1 = 1/5)$-LDC, where

$$\begin{aligned} q_1 &= (q_\alpha(n)\cdot\log n)^{O(\log\log n)}, \\ \delta_1 &= 2^{-O(\log\log(n)\log\log\log n)}, \\ \rho_1 &= 1 - O\left(\frac{1}{\log\log n}\right). \end{aligned}$$

Recall that by the Katz-Trevisan bound [23], constant rate LDC with distance δ have query complexity $\Omega(\log(\delta n/\log n))$ (see, e.g., [42]). Thus, $q_\alpha(n) = \Omega(\log n)$ and so, in fact, $q_1 = q_\alpha(n)^{O(\log\log n)}$. The resulted code is obtained by amplifying the distance from δ_1 to constant. Indeed, one can invoke, say, the AEL distance amplification procedure. Since $1/\delta = o(q_1)$, the proof follows. ◀

Proof of Corollary 7. With the hypothesis of the corollary, we may apply Corollary 79 with $\alpha(n) = 1/(\log\log n)^c$ and $\beta(n) = 1/(\log n)^{c+2}$ in the notation of Corollary 79. Note that Equation (6.4) holds with this choice. Corollary 79 then yields a $(q_1, \delta_1, \varepsilon_1 = 1/5)$-LDC, where

$$\begin{aligned} q_1 &= (q_\alpha(n)\cdot\log n)^{O((\log\log n)^{c+1})}, \\ \delta_1 &= 2^{-O((\log\log n)^{c+1}\cdot\log\log\log n)}, \\ \rho_1 &= 1 - O\left(\frac{1}{\log\log n}\right). \end{aligned}$$

By the Katz-Trevisan bound [23], $q_\alpha(n) = \Omega(\log n)$ and so, in fact, $q_1 = q_\alpha(n)^{O((\log\log n)^{c+1})}$. The resulted code is obtained by amplifying the distance from δ_1 to constant. By invoking the AEL distance amplification procedure. ◀

6.5 Proof of Corollary 8

In this section we prove Corollary 8 based on Proposition 65. We start by prove thing following.

▶ **Corollary 83.** *There exists a constant $c \geq 1$ such that the following holds. Let $0 < \alpha < 1$ be an arbitrary constant, and $\beta : \mathbb{N} \to (0,1)$ a monotone non-increasing function that satisfy*

$$2^{-\frac{1}{6}(\log n)^{\alpha}} \leq \beta(n) \leq \frac{c}{\log\log n} \tag{6.10}$$

Assume there exists a family of $(q_{\alpha}(n), 2^{-(\log n)^{\alpha}}, 1/5)$-LDC over alphabet Σ having rate $1-\beta(n)$. Then, for every sufficiently large n there exists a $(q, \delta, 1/5)$-LDC on block length m over Σ, for which $\log m \in [\log n, (\log n)^{1/(1-\alpha)}]$, and

$$q = q_{\alpha}(n)^{O(\log\log\log n)},$$
$$\rho = 1 - O\left(\beta(\log n)\log\log\log n\right),$$
$$\delta = \beta(\log n)^{O(\log\log\log n)}.$$

To prove Corollary 83, we prove the following claim. In its statement we refer to the constant $c_{\mathsf{samp}} \geq 1$ that is given by Theorem 15.

▷ Claim 84. Let $\beta_2 < 1/2$. Assume there exists a $(q_{\text{in}}, \delta_{\text{in}}, \varepsilon_{\text{in}})$-LDC C_{in} over alphabet Σ for every message length $k_{\text{in}} \in [D/2, 2D]$ where

$$D = \frac{4c_{\mathsf{samp}}2^{(\log n)^{\alpha}}}{\beta_2^6}, \tag{6.11}$$

having rate $\rho_{\text{in}} \geq 1/2$. Then, under the hypothesis of Corollary 83, there exists a $(q_{\text{out}}, \delta_{\text{out}}, \varepsilon_{\text{out}})$-LDC over Σ with block length n having rate ρ_{out}, where

$$\frac{q_{\text{out}}}{q_{\text{in}}} \leq \frac{8c_{\mathsf{samp}}(\log n)^{\alpha}}{\beta_2^6} \cdot q_{\alpha}(n),$$
$$\frac{\delta_{\text{out}}}{\delta_{\text{in}}} \geq \frac{\beta_2^3}{16},$$
$$\frac{\rho_{\text{out}}}{\rho_{\text{in}}} \geq (1-\beta_2)\left(1-\beta(n)\right),$$
$$\varepsilon_{\text{out}} \leq \frac{1}{5} + n\varepsilon_{\text{in}}.$$

Proof. Let C_1 be the LDC from the hypothesis of Corollary 83 taken with block length $n_1 = n$. Let C_2 be a code set with message length $k_2 = \frac{4c_{\mathsf{samp}}(\log n)^{\alpha}}{\beta_2^6}$, over Σ having rate $1-\beta_2$ and distance $\delta_2 = \beta_2^3$. A code with such parameters exists, over any alphabet, by the Gilbert-Varshamov bound.

In the distance amplification procedure (Section 6.1), we make use of a $(\delta_2/2, \delta_1)$ sampler $G = ([\ell], [r], E)$ with $\ell = n_1/k_2$ and left-degree $d = n_2$. For the proof of the claim, we will instantiate the distance amplification procedure with the sampler that is given by Theorem 15, and so we must verify that the left-degree is indeed large enough with respect to the parameters of the sampler. As, in our case, the left degree is n_2, we need to verify that

$$n_2 \geq c_{\mathsf{samp}} \cdot \frac{\log\left(1/\delta_1\right)}{(\delta_2/2)^2} = \frac{4c_{\mathsf{samp}}(\log n)^{\alpha}}{\beta_2^6}, \tag{6.12}$$

which indeed holds as the right hand side equals k_2.

As for the degree D_v of any given right vertex v of the sampler, we have by Theorem 15 that $D_v \in [D/2, 2D]$, where

$$D = \frac{\ell d}{r} = \frac{4c_{\mathsf{samp}} n^{1-\alpha(n)}}{\beta_2^6},$$

is as defined in Equation 6.11. Thus, we may use C_{in} as in the hypothesis of the claim. We are therefore in a position to apply Proposition 65, and the proof readily follows. ◁

As in the previous section, it will be convenient to have no error loss in the reduction that is given by Claim 80. This is easily achievable by amplifying the error of the input code before applying the previous claim. We state the following corollary whose proof is similar to the proof of Corollary 81 and so we omit it.

▶ **Corollary 85.** *Let $\beta_2 < 1/2$. Assume there exists a $(q_{\text{in}}, \delta_{\text{in}}, 1/4)$-LDC C_{in} over alphabet Σ for every message length $k_{\text{in}} \in [D/2, 2D]$ where D is as defined in Equation* (6.11)*, having rate $\rho_{\text{in}} \geq 1/2$. Then, under the hypothesis of Corollary 83, there exists a $(q_{\text{out}}, \delta_{\text{out}}, 1/4)$-LDC over Σ with block length n having rate ρ_{out}, where*

$$\frac{q_{\text{out}}}{q_{\text{in}}} \leq \frac{\log^2 n}{\beta_2^6} \cdot q_\alpha(n),$$
$$\frac{\delta_{\text{out}}}{\delta_{\text{in}}} \geq \frac{\beta_2^3}{16},$$
$$\frac{\rho_{\text{out}}}{\rho_{\text{in}}} \geq (1-\beta_2)(1-\beta(n)).$$

With Corollary 85 we are ready to prove Corollary 83.

Proof of Corollary 83. The construction of the asserted code starts by devising a sequence of LDC $C'_0, C'_1, C'_2, \ldots$ where C'_0 is taken to be a code over Σ with block length $n_0 = \log n$, having rate $1 - \beta(\log n)$ and distance $\beta(\log n)^3$. We obtain such code using Lemma 19. Clearly, as an LDC, this code has error $\varepsilon_0 = 0$ and query complexity n_0. For $t > 0$, the code C'_t is obtained by applying Corollary 85 with the code C'_{t-1} as C_{in} in the notations of the corollary and using $\beta_2 = \beta(\log n)$. Denote the message length and block length of C'_t by k_t and n_t, respectively. By construction, for every integer $t \geq 1$ such that $n_t \leq 2^{(\log n)^{1/(1-\alpha)}}$ we have that

$$k_{t-1} \leq \frac{8c_{\mathsf{samp}} 2^{(\log n_t)^\alpha}}{\beta_2^6}$$

By Corollary 81,

$$\rho_t = \frac{k_t}{n_t} \geq (1-\beta(\log n))^2 \rho_{t-1},$$

and so

$$\rho_t \geq (1-\beta(\log n))^{2t} \rho_0 = (1-\beta(\log n))^{2t+1}.$$

In particular, for every $t \leq \frac{1}{4\beta(\log n)}$ we get

$$\rho_t \geq (1-\beta(\log n))^{1+\frac{1}{2\beta(\log n)}} \geq \frac{1}{2}.$$

The the last inequality follows since the function $(1-x)^{1+\frac{1}{2x}} \geq \frac{1}{2}$ for all $x \leq 0.1$. Note that, indeed, by our assumption on β if follows that for a large enough n, $\beta(n)$ is bounded above by 0.1. Therefore,

$$n_{t-1} \leq 2k_{t-1} \leq \frac{8c_{\mathsf{samp}} 2^{(\log n_t)^{\alpha}}}{\beta_2^6}.$$

Now, per our assumption that is given by Equation (6.10), we have that

$$\beta_2 = \beta(\log n) \geq 2^{-\frac{1}{6}(\log\log n)^{\alpha}} \geq 2^{-\frac{1}{6}(\log n_t)^{\alpha}},$$

where the last inequality follows as $n_0 = \log n$. Thus, we get

$$n_{t-1} \leq 8c_{\mathsf{samp}} 2^{2(\log n_t)^{\alpha}} \leq 8^{(\log n_t)^{\alpha}}.$$

Thus, $\log n_t \geq \left(\frac{\log n_{t-1}}{3}\right)^{1/\alpha}$. By Claim 82, we get

$$\log n_t \geq \left(\frac{\log n_0}{3^{\frac{1}{1-\alpha}}}\right)^{\frac{1}{\alpha^t}} \geq 2^{\frac{1}{\alpha^t}}.$$

We now take t' to be the least integer for which the right hand side is larger or equal than $\log n$. Note that $t' = \Theta(\log\log\log n)$. However, the above analysis only holds only for $t \leq \frac{1}{4\beta(\log n)}$ and so one must verify that $t' \leq \frac{1}{4\beta(\log n)}$ which does indeed hold per our assumption that is given by Equation (6.10).

By the above, we get that $C'_{t'}$ is a $(q', \delta', 1/4)$-LDC having rho ρ' where

$$q' = (q_\alpha(n)\log n)^{O(\log\log\log n)},$$
$$\rho' = 1 - O\left(\beta(\log n)\log\log\log n\right),$$
$$\delta' = \beta(\log n)^{O(\log\log\log n)}.$$

By [23], $q_\alpha(n) = \Omega(\log n)$ and so, in fact, $q' = q_\alpha(n)^{O(\log\log\log n)}$. The final code is obtained by amplifying the distance from δ' to constant. By invoking, say, the AEL distance amplification procedure. ◀

6.6 Explicit reduction to LDC with polynomially-small distance

In this section we show a result similar to the one proven in Section 6.4, but with an explicit reduction that yields an explicit code. Throughout this section we assume $\Sigma = \mathbb{F}_p$ for some prime power p (this is needed for the existence of explicit base codes). We prove the following corollary of Proposition 76

▶ **Corollary 86.** *Let $\alpha > 0$ be a constant. Let $\beta : \mathbb{N} \to (0,1)$ be a monotone non-increasing function that satisfies*

$$\frac{1}{n} \leq \beta(n) \leq \frac{\log(1/\alpha)}{24\log n}. \tag{6.13}$$

Assume there exists a family of explicit $(q_\alpha(n), n^{-\alpha'}, 1/5)$-LDC over alphabet Σ having rate $1-\beta(n)$ for block-length n. Then, for every sufficiently large n there exists an explicit $(q, \delta, 1/5)$-LDC on block length poly(n) *over Σ, where*

$$q = (q_\alpha(n)\log n)^{O(\log\log n)},$$
$$\rho = 1 - O\left(\beta(\log n)\log\log n\right),$$
$$\delta = \beta(\log n)^{O(\log\log n)}.$$

Note that the distance δ above can then be further amplified to a constant, at the expense of lowering the rate from $1-o(1)$ to some constant, without asymptotic cost in query complexity. Indeed, in the above corollary, $1/\delta = \mathrm{poly}(q)$ per our assumption that $\beta(\log n) \geq 1/\log n$.

To prove Corollary 86, we prove the following claim. In what follows, we refer to $c = c(\Delta)$ – the function that appears in the statement of Theorem 17.

▷ Claim 87. There exists a universal constant $\beta_0 \leq \frac{1}{2}$ such that the following holds. Let n be an integer, and $\beta_2 \in (\frac{1}{\log n}, \beta_0)$. Assume there exists an explicit $(q_{\mathrm{in}}, \delta_{\mathrm{in}}, \varepsilon_{\mathrm{in}})$-LDC C_{in} over alphabet Σ for every message length $k_{\mathrm{in}} \in [D'/2, 4D']$ where $D' = D'(1/\sqrt{\alpha}, \delta_2/8, \delta_1)$ is as defined in Equation (3.2), having rate $\rho_{\mathrm{in}} \geq 1/2$. Then, under the hypothesis of Corollary 86, there exists an explicit $(q_{\mathrm{out}}, \delta_{\mathrm{out}}, \varepsilon_{\mathrm{out}})$-LDC over Σ with block-length n having rate ρ_{out}, where

$$\frac{q_{\mathrm{out}}}{q_{\mathrm{in}}} \leq (\log n)^{10c(1/\sqrt{\alpha})} \cdot q_\alpha(n),$$
$$\frac{\delta_{\mathrm{out}}}{\delta_{\mathrm{in}}} \geq \frac{\beta_2^3}{16},$$
$$\frac{\rho_{\mathrm{out}}}{\rho_{\mathrm{in}}} \geq (1-\beta_2)\,(1-\beta(n))\,,$$
$$\varepsilon_{\mathrm{out}} \leq \frac{1}{5} + n\varepsilon_{\mathrm{in}}.$$

Proof. Let C_1 be the LDC from the hypothesis of Corollary 86 taken with block length $n_1 = n$. Set $\delta_2 = \beta_2^3$. By Theorem 17, invoked with $\Delta = 1/\sqrt{\alpha}$, there exists an explicit $(\delta_2/8, \delta_1)$-sampler with $z = n/(1-\beta_2)$ edges. By Theorem 17, G has left-degree

$$d = \left(\frac{8}{\delta_2}\log\frac{1}{\delta_1}\right)^c = \left(\frac{8}{\beta_2^3}\alpha \log n\right)^c,$$

where $c = c(\Delta) = c(1/\sqrt{\alpha})$ is the constant as defined in Theorem 17. Note that since $\beta_2 \geq 1/\log n$ we have that $d \leq (\log n)^{10c}$. We also have that the average right-degree D is in $[D', 2D']$, where

$$D' = \frac{d}{2} \cdot \left(\frac{2}{\delta_1}\right)^{\Delta+1} \leq n^{2\sqrt{\alpha}},$$

where the inequality holds for all sufficiently large n.

Let C_2 be an explicit code set with message length $k_2 = (1-\beta_2)d$ over Σ having rate $1-\beta_2$ and distance $\delta_2 = \beta_2^3$. An explicit code with such parameters exists, by Lemma 19, as we can choose β_0 to be smaller than the least β for which the lemma holds.

We now want to instantiate the distance amplification procedure with C_1, C_2, the sampler G, and the code family C_{in} as C_3. Note that since the right degrees of the sampler G are not necessarily bounded, we use the relaxed distance amplification of Section 6.3. Recall that it is a prerequisite of the distance amplification procedure that the sampler has n_1/k_2 left vertices, and that $n_2 = d$, the degree of the sampler. Both of these hold, as indeed, the block length of C_2 is $\frac{1}{1-\beta_2}(1-\beta_2)d = d$, and the number of left vertices of the sampler is $\frac{z}{d} = \frac{n}{d(1-\beta_2)} = n_1/k_2$. Further note that the distance amplification procedure requires that the family C_3 contains a code with message length k_3 for every $k_3 \in [D/2, 2D]$, and this is indeed satisfied by the assumption regarding the message lengths of the code family C_{in}, of the hypothesis of the claim.

With C_1, C_2, G and C_{in} at hand, we can now apply Proposition 76 of the distance amplification procedure. The assertions regarding the query complexity, distance and rate readily follow by Proposition 65. That the error is bounded as stated readily follows by noting that $\varepsilon_2 = 0$. ◁

As in the previous sections, it will be convenient to have no error loss in the reduction that is given by Claim 87. This is easily achievable by amplifying the error of the input code before applying the previous claim. We state the following corollary whose proof is similar to the proof of Corollary 81 and so we omit it.

▶ **Corollary 88.** *There exists a universal constant $\beta_0 \le \frac{1}{2}$ for which the following holds. Let $\beta_2 \in (\frac{1}{\log n}, \beta_0)$. Assume there exists an explicit $(q_{\text{in}}, \delta_{\text{in}}, 1/4)$-LDC C_{in} over alphabet Σ for every message length $k_{\text{in}} \in [D'/2, 4D']$ where $D' = D'(1/\sqrt{\alpha}, \delta_2/8, \delta_1)$ is as defined in Equation (3.2), having rate $\rho_{\text{in}} \ge 1/2$. Then, under the hypothesis of Corollary 86, there exists an explicit $(q_{\text{out}}, \delta_{\text{out}}, 1/4)$-LDC over Σ with block-length n having rate ρ_{out}, where*

$$\frac{q_{\text{out}}}{q_{\text{in}}} \le (\log n)^{10c(1/\sqrt{\alpha})} \cdot q_\alpha(n),$$
$$\frac{\delta_{\text{out}}}{\delta_{\text{in}}} \ge \frac{\beta_2^3}{16},$$
$$\frac{\rho_{\text{out}}}{\rho_{\text{in}}} \ge (1-\beta_2)\,(1-\beta(n))\,.$$

With Corollary 88 we are ready to prove Corollary 86.

Proof of Corollary 86. The construction of the asserted code is obtained by devising a sequence of LDC $C'_0, C'_1, C'_2, \ldots$ where C'_0 is taken to be a code over Σ with block length $n_0 = \log n$ having rate $\rho_0 = 1 - \beta(\log n)$ and distance $\beta(\log n)^3$. By Lemma 19 such an explicit code exists, for every large enough n (the lemma holds for every small enough β, and indeed by Equation (6.13), $\beta(n)$ is decreasing). Clearly, as an LDC, this code has error $\varepsilon_0 = 0$ and query complexity n_0. For $t > 0$, the code C'_t is obtained by applying Corollary 88 with the code C'_{t-1} as C_{in} in the notations of the corollary and using $\beta_2 = \beta(\log n)$. Note that per our assumption given by Equation (6.13), this choice satisfies $\beta_2 \ge \frac{1}{\log n}$, and for large enough n, $\beta(n) \le \beta_0$, and so we can apply the corollary. Denote the message length and block length of C'_t by k_t and n_t, respectively. By construction, for every integer $t \ge 1$ we have that

$$k_{t-1} \le n_t^{2\sqrt{\alpha}} \le n_t^{\alpha^{1/4}}, \tag{6.14}$$

where the last inequality holds for all large enough n. By Corollary 88,

$$\rho_t = \frac{k_t}{n_t} \ge (1-\beta(\log n))^2 \rho_{t-1},$$

and so

$$\rho_t \ge (1-\beta(\log n))^{2t}\rho_0 = (1-\beta(\log n))^{2t+1}.$$

In particular, for every $t \le \frac{1}{4\beta(\log n)}$ we get

$$\rho_t \ge (1-\beta(\log n))^{1+\frac{1}{2\beta(\log n)}} \ge \frac{1}{2}.$$

The last inequality follows since the function $(1-x)^{1+\frac{1}{2x}} \ge \frac{1}{2}$ for all $x \le 0.1$, and for every large enough n, $\beta(n) \le 0.1$. By Equation (6.14) we have that for every $t \le \frac{1}{4\beta(\log n)}$,

$$n_{t-1} \le 2k_{t-1} \le 2n_t^{\alpha^{1/4}} \le n_t^{\alpha^{1/5}}.$$

Thus,

$$n_t \geq n_0^{\frac{1}{\alpha^{t/5}}}. \tag{6.15}$$

It follows that by taking $t' = \lceil \frac{5 \log \log n}{\log(1/\alpha)} \rceil$ we get that $n_{t'} \geq n$. However we need to verify that this choice satisfies $t' \leq \frac{1}{4\beta(\log n)}$ for the above analysis to hold. Indeed per our assumption given by Equation (6.13), it holds that $\frac{6 \log \log n}{\log(1/\alpha)} \leq \frac{1}{4\beta(\log n)}$.

It is easy to verify that the query complexity $q_{t'}$ of and distance $\delta_{t'}$ of $C'_{t'}$ are

$$q_{t'} = ((\log n) q_\alpha(n))^{\Theta(t')},$$
$$\delta_{t'} = \beta(\log n)^{\Theta(t')}.$$

As for the rate,

$$\rho_{t'} \geq (1 - \beta(\log n))^{\Theta(t')} = 1 - O\left(\beta(\log n) \log \log n\right).$$

Finally, the error of $C'_{t'}$ can be reduced from 1/4 to 1/5 with no asymptotic overhead in query complexity, and so $C'_{t'}$ has all the asserted properties. ◀

References

1 Noga Alon, Jeff Edmonds, and Michael Luby. Linear time erasure codes with nearly optimal recovery. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 512–519. IEEE, 1995.

2 Noga Alon and Michael Luby. A linear time erasure-resilient code with nearly optimal recovery. *IEEE Transactions on Information Theory*, 42(6):1732–1736, 1996.

3 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

4 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM (JACM)*, 45(1):70–122, 1998.

5 László Babai, Lance Fortnow, Leonid A Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 21–32, 1991.

6 Laszlo Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.

7 Omer Barkol, Yuval Ishai, and Ronny Roth. *Locally decodable codes and their applications*. PhD thesis, Computer Science Department, Technion, 2008.

8 Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 37–48. Springer, 1990.

9 Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 1994*, pages 276–287. IEEE, 1994.

10 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 73–83, 1990.

11 Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 41–50. IEEE, 1995.

12 Zeev Dvir. On matrix rigidity and locally self-correctable codes. *computational complexity*, 20(2):367–388, 2011.

13 Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM Journal on Computing*, 40(4):1154–1178, 2011.

14 Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM Journal on Computing*, 41(6):1694–1703, 2012.

15 Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *STOC*, volume 91, pages 32–42. Citeseer, 1991.

16 Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Information processing letters*, 43(4):169–174, 1992.

17 Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography*, pages 302–332. Springer, 2011. `doi:10.1007/978-3-642-22670-0_24`.

18 Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM (JACM)*, 53(4):558–655, 2006.

19 Sivakanth Gopi, Swastik Kopparty, Rafael Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally testable and locally correctable codes approaching the gilbert-varshamov bound. *IEEE Transactions on Information Theory*, 64(8):5813–5831, 2018.

20 Alan Guo, Swastik Kopparty, and Madhu Sudan. New affine-invariant codes from lifting. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 529–540. ACM, 2013.

21 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. *Draft available at* `http://www.cse.buffalo.edu/~atri/courses/coding-theory/book`, 2012.

22 Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander codes. *Information and Computation*, 243:178–190, 2015.

23 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 80–86, 2000.

24 Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 590–600. IEEE, 2007.

25 Kiran S Kedlaya and Sergey Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of mersenne numbers. *SIAM Journal on Computing*, 38(5):1952–1969, 2009.

26 Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *Journal of the ACM (JACM)*, 64(2):11, 2017.

27 Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *Journal of the ACM (JACM)*, 61(5):28, 2014.

28 Ray Li and Mary Wootters. Lifted multiplicity codes and the disjoint repair group property. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

29 Richard J. Lipton. Efficient checking of computations. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 207–215. Springer, 1990.

30 Irving S Reed. A class of multiple-error-correcting codes and the decoding scheme. Technical report, Massachusetts inst of tech Lexington Lincoln lab, 1953.

31 Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Electronic Colloquium on Computational Complexity (ECCC)*, page 18, 2001. URL: `https://eccc.weizmann.ac.il/report/2001/018/`.

32 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

33 C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001. Originally appeared in *Bell System Tech. J.* 27:379–423, 623–656, 1948.

34 Carl Siegel. Über die classenzahl quadratischer zahlkörper. *Acta Arithmetica*, 1(1):83–86, 1935.

35 Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

36 Luca Trevisan. List-decoding using the XOR lemma. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 126–135. IEEE, 2003.

37 Arnold Walfisz. Zur additiven zahlentheorie. ii. *Mathematische Zeitschrift*, 40(1):592–607, 1936.

38 David Woodruff. New lower bounds for general locally decodable codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, 2007.

39 S. Yekhanin. Locally decodable codes. In *International Computer Science Symposium in Russia*, pages 289–290. Springer, 2011.

40 Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1–16, 2008.

41 Kalina Petrova Zeev Dvir. Lecture 1: Introduction. Lecture notes: `https://www.cs.princeton.edu/~zdvir/LDCnotes/LDC1.pdf`, year = 2016.

42 Kalina Petrova Zeev Dvir. Lecture 4: Lower bounds for r-query LDCs. Lecture notes: `https://www.cs.princeton.edu/~zdvir/LDCnotes/LDC4.pdf`, 2016.

43 Victor Vasilievich Zyablov. An estimate of the complexity of constructing binary linear cascade codes. *Problemy Peredachi Informatsii*, 7(1):5–13, 1971.

An Improved Protocol for the Exactly-N Problem[*]

Nati Linial ✉
Hebrew University of Jerusalem, Israel

Adi Shraibman ✉
The Academic College of Tel-Aviv-Yaffo, Israel

Abstract

In the 3-players exactly-N problem the players need to decide whether $x + y + z = N$ for inputs x, y, z and fixed N. This is the first problem considered in the multiplayer Number On the Forehead (NOF) model. Even though this is such a basic problem, no progress has been made on it throughout the years. Only recently have explicit protocols been found for the first time, yet no improvement in complexity has been achieved to date. The present paper offers the first improved protocol for the exactly-N problem. This improved protocol has also interesting consequences in additive combinatorics. As we explain below, it yields a higher lower bound on the possible density of corner-free sets in $[N] \times [N]$.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Communication complexity, Number-On-the-Forehead, Corner-free sets

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.2

Funding *Nati Linial*: Supported in part by Grant 659/18 “High dimensional combinatorics” of the Israel Science Foundation.

Acknowledgements We thank Noga Alon, Aya Bernstine and Alex Samorodnitsky for insightful discussions.

1 Introduction

The multiplayer Number On the Forehead (NOF) model of communication complexity was introduced by Chandra, Furst and Lipton [9]. Given a function $f : [N]^k \to \{0, 1\}$, the k players in this scenario should jointly find out $f(x_1, \ldots, x_k)$. We think of x_i as being placed on player i's forehead, so that each player sees the whole input bar one argument. Players communicate by writing bits on a shared blackboard according to an agreed-upon protocol. This model is intimately connected to several key problems in complexity theory. E.g., lower bounds on the size of ACC^0 circuits for a natural function in P [23, 12], branching programs, time-space tradeoffs for Turing machines [13], and proof complexity [5]. In addition, progress in the NOF model, even for a specific problem and for $k = 3$, would have profound implications in graph theory and combinatorics [14, 3].

Much of Chandra, Furst and Lipton's seminal paper [9] is dedicated to the exactly-N function $f : [N]^k \to \{0, 1\}$, where $f(x_1, \ldots, x_k) = 1$ iff $\sum x_i = N$. They discovered a connection between the communication complexity of this function and well-known problems in additive combinatorics and Ramsey theory. They used Ramsey's theory to prove a (rather weak) lower bound on the NOF communication complexity of this function. Using the connection to additive number theory, they showed that a $O(\sqrt{\log N})$ protocol exists, although they have not made this protocol explicit.

[*] Our companion paper “Larger Corner-Free Sets from Better NOF Exactly-$\boldsymbol{N}$ Protocols” presents the same results, emphasizing the combinatorial perspective.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 2; pp. 2:1–2:8
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

There are several reasons why it is highly significant to determine the communication complexity of the exactly-N function, aside of the very fundamental nature of the problem:

- Our poor understanding of this question is manifested by the huge gap between the upper and lower bounds that we currently have on the communication complexity of this problem. This gap is double exponential for three players, and is even worse for $k > 3$ players.
- Despite the significance of the NOF model, we still know very little about it. The rich web of mathematical and computational concepts surrounding the exactly-N function suggests that it may open the gate to progress in understanding numerous other NOF functions.
- The k-player exactly-N function is a *graph function* [4]. For most functions in this class the deterministic and randomized communication complexity differ substantially, but no explicit function with a significant gap is presently known.
- This problem is *equivalent* to corner theorems in additive combinatorics (e.g., [2]), and is closely related to other important problems such as constructing Ruzsa-Szemerédi graphs and the triangle removal lemma [14, 3].

Nevertheless, progress on the complexity of the NOF exactly-N problem has mostly been made on the additive combinatorics side and includes several breakthrough results such as Szemerédi's regularity lemma [21], and its extension to hypergraphs [11, 17, 16]. Translated back to the realm of NOF communication complexity, these advances bear on lower bounds in communication complexity, yet there is essentially nothing concerning upper bounds. We believe that the more promising line of attack is for advances in communication complexity to shed light on questions in additive combinatorics by exploiting the power of new algorithmic ideas.

As mentioned, the existence of a protocol for the exactly-N problem has already been known since [9]. However, this was just an existential statement and no actual protocol was provided. This lacuna was recently remedied with two protocols [14, 3] of the exact same complexity as the one whose existence was proven in [9], namely of complexity[1]

$$2\sqrt{2}\sqrt{\log N} + o(\sqrt{\log N}) = 2.828...\sqrt{\log N} + o(\sqrt{\log N}). \tag{1}$$

Here we give the first improved protocol for the exactly-N problem, and prove

▶ **Theorem 1.** *There is an explicit protocol for NOF exactly-N of complexity*

$$2\sqrt{\log e}\sqrt{\log N} + o(\sqrt{\log N}) = 2.4022...\sqrt{\log N} + o(\sqrt{\log N}). \tag{2}$$

Due to the connection between NOF complexity and additive combinatorics, our improved protocol has interesting implications in that area that we briefly mention now. More details are given in Section 3. Let $\rho_3(N)$ be the largest density of a subset of $[N]$ that contains no 3-term arithmetic progression. As Roth [18] showed, $\rho_3(N) = o(1)$. However, we still do not know the rate at which $\rho_3(N)$ tends to 0. The upper bounds have gradually improved over the years and the current "world record" found in 2020 by Bloom and Sisask [8] is

$$\rho_3(N) \leq (\log N)^{-1-c} \text{ for some absolute constant } c > 0.$$

[1] All logarithms in this paper are in base 2.
Also, unless otherwise specified, all asymptotic statements are taken with $N \to \infty$.

Much less has happened with the lower bound. Behrend's construction [6] yields

$$\rho_3(N) \geq 2^{-2\sqrt{2}\sqrt{\log N}+o(\sqrt{\log N})} = 2^{-2.828\ldots\sqrt{\log N}+o(\sqrt{\log N})}.$$

But in the ensuing 75 years, only the little-oh term saw an improvement (Elkin [10]).

A *corner* in $\mathbb{N}^2$ is a triple of points $(x,y),(x+\delta,y),(x,y+\delta)$ for some $\delta \neq 0$. Let $\rho_3^{\angle}(N)$ be the largest density of a subset of $[N]\times[N]$ that contains no corner. Ajtai and Szemerédi's *corner theorem* [2] shows that $\rho_3^{\angle}(N) = o(1)$. This readily implies Roth's theorem that $\rho_3(N) = o(1)$.

The best previously known lower bound on $\rho_3^{\angle}(N)$ again comes from Behrend's construction:

$$\rho_3^{\angle}(N) \geq 2^{-2\sqrt{2}\sqrt{\log N}+o(\sqrt{\log N})} = 2^{-2.828\ldots\sqrt{\log N}+o(\sqrt{\log N})}.$$

Our work gives the first improvement in decades, showing (Theorem 3)

$$\rho_3^{\angle}(N) \geq 2^{-2\sqrt{\log e}\sqrt{\log N}+o(\sqrt{\log N})} = 2^{-2.4022\ldots\sqrt{\log N}+o(\sqrt{\log N})}.$$

2 Proof of Theorem 1

The three players in our protocol are called P_x, P_y and P_z. The inputs that they get to see are $(y,z),(x,z)$ and (x,y) respectively.

Here is a similar problem in the realm of vector addition. Given integers $q,d>1$, define $g = g_{q,d}(\alpha,\beta,\gamma)$ to be 1 if $\alpha+\beta=\gamma$ and 0 otherwise. Here $\alpha,\beta \in [q]^d$, $\gamma \in [2q]^d$ and addition is vector addition in $\mathbb{R}^d$. The following one-round protocol [3] for g is correct because the inequality $\|2\alpha-\gamma\|^2 + \|2\beta-\gamma\|^2 \geq 2\|\alpha-\beta\|^2$ holds always and is an equality iff $\gamma = \alpha+\beta$.

▶ **Protocol 1.** A protocol for $g_{q,d}$

1. P_z *computes* $\|\alpha-\beta\|_2^2$*, and writes the result on the board.*
2. P_y *writes* 1 *iff* $\|\alpha-\beta\|_2^2 = \|2\alpha-\gamma\|_2^2$.
3. P_x *writes* 1 *iff* $\|\alpha-\beta\|_2^2 = \|2\beta-\gamma\|_2^2$.

The cost of this protocol is $2+\log dq^2$.

The above is an efficient method to decide high-dimensional vector addition, but our objective is to decide the integer addition relation $X+Y+Z=N$. We let $x=X, y=Y$ and $z=N-Z$, so the relation we need to consider is $x+y=z$.

Our protocol to decide whether $x+y=z$ builds on the protocol for $g_{q,d}$. It is the issue of carry bits in integer addition that makes this decision problem harder. The integers $q,d>1$ are chosen so that

$$2qN > q^d \geq 2N. \tag{3}$$

the specific choice is made below so as to minimize the cost of the protocol.

We denote by w_q the vector that corresponds to the base q representation of the integer w.

As usual, e_i is the d-dimensional vector with 1 in the i-th coordinate and zeros elsewhere. Let $C(x,y)\in\{0,1\}^d$ be the carry vector when x and y are added in base q. The relation $x+y=z$ among integers is equivalent to the vector relation

$$x_q + y_q = \zeta,$$

where the i-th coordinate of ζ is

$$\zeta_i = z_i + q\cdot C(x,y)_i - C(x,y)_{i-1}$$

(Here $C(x,y)_0 = 0$).

The protocol from [3] now suggests itself: P_z posts $C(x,y)$, and Protocol 1 is used to decide the relation $x_q + y_q = \zeta$. This yields again the estimate (1).

The alternative approach that we adopt here considers instead the equivalent vector relation

$$x_q + \eta = z_q$$

where

$$\eta = (x+y)_q - x_q.$$

Concretely, the i-th coordinate of η is:

$$\eta_i = y_i - q \cdot C(x,y)_i + C(x,y)_{i-1}.$$

In order to run Protocol 1, P_z needs to know η and x_q, which he does. The situation with P_y is even simpler, since he needs to know x_q and z_q which are his inputs. The only difficulty is with P_x who needs to know z_q (which he does) and η. The latter is not part of his input and P_z fills in the missing information for him.

The obvious solution is for P_z to reveal C to P_x using d bits of information. However, we can save communication by exploiting the fact that P_x and P_z share some information, i.e., they both know y for every $y \neq 0$.

By a standard argument in this area which we detail below (Proposition 2), a protocol that works for *typical* pairs x, y can be easily modified to work in *all* cases. So, let us pick x and y uniformly at random from among the d-digit numbers in base q and think of C, the vector of carry bits as a random variable on this probability space. The number of bits that P_z needs to post so that P_x gets to know C, and therefore know η, is $H(C|y)$, the entropy of C given y. The gain is clear, since $H(C) \geq H(C|y)$.

It remains to estimate $H(C|y)$. Fix some integers $s \geq t \geq 0$, and let X be the random variable that is a uniformly sampled subset of $[s]$ of cardinality $\geq t$. It is easily verified that $H(X) = (1 + o_s(1)) \cdot s \cdot h(t/s)$, where $h(\cdot)$ is the univariate entropy function. The entropy of X is the same also if we sample subsets of $[s]$ of cardinality $< t$. Let r be an integer in the range $d \gg r \gg 1$, e.g., $r \approx \sqrt{d}$. For $j = 1, \ldots, r$, let

$$S_j = \{i \mid \frac{qj}{r} > y_i \geq \frac{q(j-1)}{r}\},$$

where $q > y_i \geq 0$ is the i-th digit of y. A carry occurs in digit $i \in S_j$ only if $x_i > \frac{q(r-j)}{r}$, where x_i is the i-th digit of x. Then

$$H(C|y) \leq (1 + o_r(1)) \sum_{j=1}^{r} \frac{|S_j|}{d} h(\frac{j}{r}).$$

Since y is chosen at random, $|S_j| \leq (1 + o_r(1))\frac{d}{r}$, and so

$$H(C|y) \leq (1 + o_r(1)) \sum_{j=1}^{r} \frac{1}{r} h(\frac{j}{r}).$$

The limit of this expression as $r \to \infty$ is

$$\lambda = \int_0^1 h(u)du = \frac{\log e}{2} = 0.721...$$

It is left to optimize on q and d. The complexity of our protocol is

$$\lambda d + \log dq^2 + 2,$$

where recall that $2qN > q^d \geq 2N$. It is not hard to verify that choosing

$$d = \sqrt{\frac{2}{\lambda} \log 2N} \qquad q = 2^{\sqrt{\frac{\lambda}{2} \log 2N}}, \tag{4}$$

we get a protocol with complexity

$$2\sqrt{2\lambda \log N} + o(\sqrt{\log N}),$$

and this is asymptotically optimal in our setting.

To sum up, here is the protocol which proves Theorem 1:

▶ **Protocol 2.** A protocol for exactly-N, for typical pairs x, y

For d, q as in Equation (4)

1. *P_z publishes the vector $\eta = (x+y)_q - x_q$ in a way that P_x can read.*
2. *The players run protocol 1 for $g_{q,d}$ on inputs x_q, η, z_q. That is:*

   a. *P_z writes $\|\eta - x_q\|_2^2$ on the board*
   b. *P_y writes 1 iff $\|\eta - x_q\|_2^2 = \|2x_q - z_q\|_2^2$.*
   c. *P_x writes 1 iff $\|\eta - x_q\|_2^2 = \|2\eta - z_q\|_2^2$.*

▶ **Proposition 2.** *Let $\mathcal{P}$ be an NOF protocol for the exactly-N that works correctly for an $\Omega(1)$-fraction of the input pairs x, y (and every z) with communication complexity $\Phi(N)$. Then there is an NOF protocol that works for all inputs with communication complexity $\Phi(N) + O(\log \log N)$.*

Proof. Let $S \subseteq [N] \times [N]$ be the set of input pairs x, y on which $\mathcal{P}$ succeeds. We claim that there is a collection F of $O(\log N)$ vectors $\Delta \in [N] \times [N]$ such that

$$\cup_{\Delta \in F} (S + \Delta) \supseteq [N] \times [N].$$

In the modified protocol, P_z sees x, y and announces the index of some $\Delta = (\Delta_1, \Delta_2) \in F$ for which $(x - \Delta_1, y - \Delta_2) \in S$. Then the players run Protocol 2 with inputs $(x - \Delta_1, y - \Delta_2, z - \Delta_1 - \Delta_2)$.

The construction of F uses a standard fact about the set-cover problem. For a family of finite sets $\mathcal{X} \subseteq 2^{\Omega}$ we denote by $c(\mathcal{X})$ the least number of members in $\mathcal{X}$ whose union is Ω. Also $c^*(\mathcal{X})$ is the minimum cost of a fractional cover. Namely,

$$c^*(\mathcal{X}) = \min \sum_{\mathcal{X}} \omega_{\mathrm{X}}, \text{ where } \omega_{\mathrm{X}} \geq 0 \text{ for every } \mathrm{X} \in \mathcal{X} \text{ and } \sum_{\mathrm{x} \in \mathrm{X}} \omega_{\mathrm{X}} \geq 1 \text{ for every } \mathrm{x} \in \Omega.$$

Then

$$c(\mathcal{X}) \leq \log(|\Omega|) \cdot c^*(\mathcal{X})$$

(e.g., Lovász [15]) and actually the greedy algorithm yields a set cover that meets this bound.

In our case $\Omega = [N] \times [N]$, and

$$\mathcal{X} = \{(S + \Delta) \cap ([N] \times [N]) \mid \Delta \in [-N, N] \times [-N, N]\}.$$

It is easily verified that the weights $\omega_x = \frac{10}{N^2}$ constitute a fractional cover, so that $c^*(\mathcal{X}) \leq 40$ and hence $c(\mathcal{X}) \leq 80 \log N$, as claimed. ◀

3 Applications in additive combinatorics

In this section we briefly explain the connections and implications in additive combinatorics.

Van der Waerden's well known theorem [22] states that for every r, k and every large enough N, if the elements of $[N] := \{1, \dots, N\}$ are colored by r colors, then there must exist a length-k monochromatic arithmetic progression. Erdős and Turán introduced the density version of this theorem. Let $\rho_k(N)$ be the largest density of a subset of $[N]$ without an arithmetic progression of length k. Szemerédi's famous theorem [21] shows that $\rho_k(N) = o(1)$ for every $k \geq 3$.

Extending van der Waerden's theorem, Gallai proved that in every finite coloring of $\mathbb{Z}^2$ some color contains arbitrarily large monochromatic square subarrays. In search of a density version of Gallai's theorem, Erdős and Graham asked about the largest density of a subset of the integer grid $[N] \times [N]$ without a *corner*, i.e., a triple $(x, y), (x + \delta, y), (x, y + \delta)$ for some $\delta \neq 0$. Denote this quantity by $\rho_3^{\angle}(N)$.

Ajtai and Szemerédi [2] proved the first *corners theorem*, showing that $\rho_3^{\angle}(N) = o(1)$. Namely, for every $\varepsilon > 0$ and large enough N, every subset of $[N] \times [N]$ of cardinality εN^2 must contain a corner. This theorem easily yields that $\rho_3(N) = o(1)$, namely, the $k = 3$ case of Szemerédi's theorem (a result of Roth [18], proved two decades before Szemerédi's theorem). Later on, Solymosi [20] showed how to derive Ajtai and Szemerédi's corners Theorem from the Triangle Removal Lemma [19].

The quantitative aspects of all these results: Szemerédi's theorem, the corner theorem, and the triangle removal lemma remain unfortunately poorly understood. In particular, we know very little concerning the lower bounds in these problems. Behrend [6] has famously constructed a large subset of $[N]$ without a 3-term arithmetic progression. This construction implies that

$$\rho_3(N) \geq 2^{-2\sqrt{2}\sqrt{\log N}+o(\sqrt{\log N})}.$$

Using similar tools Elkin [10] improved Behrend's construction. However, his construction only improves the little-o term. Behrend's construction also yields the previously best known lower bounds on $\rho_3^{\angle}(N)$, viz.

$$\rho_3^{\angle}(N) \geq 2^{-2\sqrt{2}\sqrt{\log N}+o(\sqrt{\log N})} = 2^{-2.828\ldots\sqrt{\log N}+o(\sqrt{\log N})}. \tag{5}$$

As mentioned in the introduction, the NOF communication complexity of f is closely related to corners theorems. Our Theorem 1 immediately implies,

▶ **Theorem 3.**

$$\rho_3^{\angle}(N) \geq 2^{-2\sqrt{\log e}\sqrt{\log N}+o(\sqrt{\log N})} = 2^{-2.4022\ldots\sqrt{\log N}+o(\sqrt{\log N})}.$$

There is an explicit corner-free subset of $[N] \times [N]$ *of size*

$$N^2/2^{2\sqrt{\log e}\sqrt{\log N}+o(\sqrt{\log N})}.$$

The derivation of Theorem 3 from Theorem 1 is an easy consequence of the following claim.

▷ Claim 4 ([9], implicit).

1. There is an optimal one-round protocol for the addition problem.
2. Let $T = \mathbb{T}(x, y)$ be the message that the P_z sends on inputs (x, y) in a one-round protocol for the addition problem. Then the set

$$S(T) = \{(x, y) : \mathbb{T}(x, y) = T\}$$

is corner-free.

See [9, 7, 1, 14, 3] for more details about the above claim and the relation between communication complexity and additive combinatorics. The same comments and corollaries above apply also verbatim to the $(6, 3)$ Theorem (e.g., [19]) and to the quantitative version of the triangle removal lemma.

4 Discussion

The strong relation between the exactly-N problem in the NOF model and questions in additive combinatorics has been discovered decades ago, in the seminal paper of Chundra, Furst and Lipton [9]. However, this subject remains under-developed. We believe that there is a lot to be done here, and many interesting avenues of research that this study can take. One obvious candidate for improvement is the addition problem. We conjecture:

▶ **Conjecture 5.** *The NOF communication complexity of exactly-N is $o(\sqrt{\log N})$. Possibly it is much smaller, even as small as $(\log \log N)^{O(1)}$.*

In the realm of additive combinatorics these conjectures translate to:

▶ **Conjecture 6.**

$$\rho_3^{\angle}(N) \geq 2^{-o(\sqrt{\log N})}.$$

and possibly even

$$\rho_3^{\angle}(N) \geq 2^{-(\log\log N)^{O(1)}}.$$

References

1 A. Ada, A. Chattopadhyay, O. Fawzi, and P. Nguyen. The NOF multiparty communication complexity of composed functions. *computational complexity*, 24(3):645–694, 2015.

2 M. Ajtai and E. Szemerédi. Sets of lattice points that form no squares. *Stud. Sci. Math. Hungar*, 9(1975):9–11, 1974.

3 N. Alon and A. Shraibman. Number on the forehead protocols yielding dense ruzsa–szemerédi graphs and hypergraphs. *Acta Mathematica Hungarica*, 161(2):488–506, 2020.

4 P. Beame, M. David, T. Pitassi, and P. Woelfel. Separating deterministic from randomized nof multiparty communication complexity. In *Proceedings of the 34th International Colloquium On Automata, Languages and Programming*, Lecture Notes in Computer Science. Springer-Verlag, 2007.

5 P. Beame, T. Pitassi, and N. Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity. *SIAM Journal on Computing*, 37(3):845–869, 2006.

6 F. A. Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, 1946.

7 R. Beigel, W. Gasarch, and J. Glenn. The multiparty communication complexity of Exact-T: Improved bounds and new problems. In *International Symposium on Mathematical Foundations of Computer Science*, pages 146–156. Springer, 2006.

8 T. F. Bloom and O. Sisask. Breaking the logarithmic barrier in Roth's theorem on arithmetic progressions. *arXiv preprint*, 2020. `arXiv:2007.03528`.

9 A. Chandra, M. Furst, and R. Lipton. Multi-party protocols. In *Proceedings of the 15th ACM Symposium on the Theory of Computing*, pages 94–99. ACM, 1983.

10 M. Elkin. An improved construction of progression-free sets. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 886–905. Society for Industrial and Applied Mathematics, 2010.

11 W. T. Gowers. Hypergraph regularity and the multidimensional szemerédi theorem. *Annals of Mathematics*, pages 897–946, 2007.

12 J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

13 E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

14 N. Linial, T. Pitassi, and A. Shraibman. On the communication complexity of high-dimensional permutations. In *10th Innovations in Theoretical Computer Science Conference, ITCS San Diego, California, USA*, volume 124, pages 54:1–54:20, 2019.

15 L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

16 B. Nagle, V. Rödl, and M. Schacht. The counting lemma for regular k-uniform hypergraphs. *Random Structures & Algorithms*, 28(2):113–179, 2006.

17 V. Rödl and J. Skokan. Regularity lemma for k-uniform hypergraphs. *Random Structures & Algorithms*, 25(1):1–42, 2004.

18 K. F. Roth. On certain sets of integers. *Journal of the London Mathematical Society*, 1(1):104–109, 1953.

19 I. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1978.

20 J. Solymosi. Note on a generalization of Roth's theorem. *Discrete and Computational Geometry: The Goodman-Pollack Festschrift*, pages 825–827, 2003.

21 E. Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arith*, 27(199-245):2, 1975.

22 B. L. van der Waerden. Beweis einer Baudetschen Vermutung. *Nieuw Arch. Wiskunde*, 15:212–216, 1927.

23 A. Yao. On ACC and threshold circuits. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science*, pages 619–627. IEEE, 1990.

Proof Complexity of Natural Formulas via Communication Arguments

Dmitry Itsykson
St. Petersburg Department of Steklov Mathematical Institute of
Russian Academy of Sciences, Russia

Artur Riazanov
St. Petersburg Department of Steklov Mathematical Institute of
Russian Academy of Sciences, Russia

Abstract

A canonical communication problem Search (φ) is defined for every unsatisfiable CNF φ: an assignment to the variables of φ is partitioned among the communicating parties, they are to find a clause of φ falsified by this assignment. Lower bounds on the randomized k-party communication complexity of Search (φ) in the number-on-forehead (NOF) model imply tree-size lower bounds, rank lower bounds, and size-space tradeoffs for the formula φ in the semantic proof system $\mathrm{T}^{\mathrm{cc}}(k, c)$ that operates with proof lines that can be computed by k-party randomized communication protocol using at most c bits of communication [9]. All known lower bounds on Search (φ) (e.g. [1, 9, 13]) are realized on ad-hoc formulas φ (i.e. they were introduced specifically for these lower bounds). We introduce a new communication complexity approach that allows establishing proof complexity lower bounds for natural formulas.

First, we demonstrate our approach for two-party communication and apply it to the proof system Res($\oplus$) that operates with disjunctions of linear equalities over $\mathbb{F}_2$ [14]. Let a formula PM_G encode that a graph G has a perfect matching. If G has an odd number of vertices, then PM_G has a tree-like Res($\oplus$)-refutation of a polynomial-size [14]. It was unknown whether this is the case for graphs with an even number of vertices. Using our approach we resolve this question and show a lower bound $2^{\Omega(n)}$ on size of tree-like Res($\oplus$)-refutations of $\mathrm{PM}_{K_{n+2,n}}$.

Then we apply our approach for k-party communication complexity in the NOF model and obtain a $\Omega\left(\frac{1}{k}2^{n/2k-3k/2}\right)$ lower bound on the randomized k-party communication complexity of Search $\left(\mathrm{BPHP}^{M}_{2^n}\right)$ w.r.t. to some natural partition of the variables, where $\mathrm{BPHP}^{M}_{2^n}$ is the bit pigeonhole principle and $M = 2^n + 2^{n(1-1/k)}$. In particular, our result implies that the bit pigeonhole requires exponential tree-like $\mathrm{Th}(k)$ proofs, where $\mathrm{Th}(k)$ is the semantic proof system operating with polynomial inequalities of degree at most k and $k = \mathcal{O}(\log^{1-\epsilon} n)$ for some $\epsilon > 0$. We also show that $\mathrm{BPHP}^{2^n+1}_{2^n}$ superpolynomially separates tree-like $\mathrm{Th}(\log^{1-\epsilon} m)$ from tree-like $\mathrm{Th}(\log m)$, where m is the number of variables in the refuted formula.

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Communication complexity

Keywords and phrases bit pigeonhole principle, disjointness, multiparty communication complexity, perfect matching, proof complexity, randomized communication complexity, Resolution over linear equations, tree-like proofs

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.3

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2020/184/`

Funding The research presented in Sections 3 and 4 is supported by Russian Science Foundation (project 18-71-10042).

Acknowledgements The authors are grateful to Anastasia Sofronova, Svyatoslav Gryaznov, Danil Sagunov, Petr Smirnov, Dmitry Sokolov, and Jakob Nordström for fruitful discussions and useful comments. We also thank the anonymous referees for useful comments and corrections. Dmitry Itsykson is a Young Russian Mathematics award winner and would like to thank sponsors and jury of the contest.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 3; pp. 3:1–3:34
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Propositional proof complexity studies proof systems that allow proving the unsatisfiability of Boolean CNF formulas. The main line of research in proof complexity is focused on refutation size lower bounds for different proof systems. This research activity is motivated by NP vs coNP question [3] as well as by studying properties of SAT-solvers. This paper develops the communication complexity approach to proof complexity lower bounds.

1.1 Communication complexity of search problems

In the classical communication settings, several participants collaborate to compute a function using a broadcast communication channel; each participant knows only a part of the input and the goal is to compute the function with the minimum number of transmitted bits. In the case of search problems, participants compute a relation $R \subseteq X \times Y$ instead of a function in the following sense: an input $x \in X$ is partitioned among the participants and they have to find $y \in Y$ such that $(x, y) \in R$. Analyzing the communication complexity of search problems is usually much harder than analyzing the communication complexity of functions. Unrestricted and monotone circuit depth of a Boolean function can be characterized in terms of the communication complexity of an appropriate search problem [17].

Every unsatisfiable CNF-formula φ defines a search problem $\text{Search}(\varphi)$: the values of the variables of φ are partitioned between the parties of the protocol in some way, the participants are to find a clause of φ that is falsified by the values of the variables. This problem plays an important role in proof complexity.

One of the promising approaches for obtaining proof complexity lower bounds is the investigation of *dag-like* communication protocols [20, 33]. This approach allows proving lower bounds for proof systems operating with proof lines having small communication complexity in the appropriate communication model. Every refutation of a formula φ of size S can be translated to a dag-like communication protocol for $\text{Search}(\varphi)$ of complexity $S \cdot C$, where C depends on the upper bound on the communication complexity of proof lines. Thus, lower bounds on the complexity of dag-like communication protocols imply lower bounds on the size of refutations. Nontrivial lower bounds on the size of dag-like protocols are currently known only for two-party deterministic and two-party real communication models. There are two known approaches for obtaining these lower bounds. The first is based on the correspondence between dag-like protocols and monotone Boolean/real circuits [20, 33, 11]. The second approach is lifting from the resolution width [7]. The mentioned lower bounds on dag-like communication imply lower bounds for Resolution [20], OBDD-based proof systems [21] (via deterministic protocols), and Cutting Planes [28, 10, 6, 7] (via real protocols).

Proving a superpolynomial lower bound for any of the models of dag-like communication protocols listed in the left column of Table 1 seems to be a very challenging open question. Such lower bounds would imply currently unknown superpolynomial lower bounds on the corresponding proof systems in the right column of the table.

In this paper, we deal with classical (tree-like) communication protocols. A lower bound on (tree-like) communication complexity of the problem $\text{Search}(\varphi)$ in the model from the left column of Table 1 implies a lower bound on the size of tree-like refutations of φ in the corresponding proof system from the right column as well as a lower bound on the size of dag-like refutation of φ using small space (a size-space tradeoff [9, 12]). The usual strategy for obtaining lower bounds on the proof size via communication complexity is the following: by a tree-like refutation of φ of size S (or by a realization of a dag-like refutation of φ in size S within small space), one constructs a communication protocol for $\text{Search}(\varphi)$ with

Table 1 Correspondence between communication models and proof systems.

Communication model	Proof systems
Randomized two-party protocols	Res($\oplus$) [15]. Proof lines in Res($\oplus$) are disjunctions of linear equations over $\mathbb{F}_2$.
Real k-party protocols in the number-on-forehead (NOF) model	Semantic Th$(k-1)$ [1]. Proof lines in Th$(k-1)$ are inequalities of the form $f(x_1, x_2, \ldots, x_n) \geq 0$, where f is a polynomial of degree at most $k-1$ with integer coefficients and Boolean variables.
Randomized k-party protocols in the NOF model	Semantic $\mathrm{T}^{\mathrm{cc}}(k, c)$. Proof lines in $\mathrm{T}^{\mathrm{cc}}(k, c)$ are arbitrary predicates that can be computed with k-party randomized communication cost at most c in the NOF model. $\mathrm{T}^{\mathrm{cc}}(k, c)$ for small c simulates Th$(k-1)$ and Res(PC_{k-1}). Proof lines in Res(PC_d) [22, 19] are disjunctions of polynomial equalities of the form $p(x_1, x_2, \ldots, x_n) = 0$, where p is a polynomial over $\mathbb{F}_2$ of degree at most d. Notice that Res(PC_1) coincides with Res($\oplus$).

communication complexity $\mathcal{O}(\log S \log\log S \cdot c)$[1] for an arbitrary partition of the variables of φ between the parties, where c is an upper bound for communication complexity of a proof line in the proof system in question. One then proceeds to prove a lower bound on the communication complexity of Search(φ) for some fixed partition of variables between the parties.

Proving lower bounds for the communication complexity of Search(φ) is not trivial since a lower bound on Search(φ) in the two-party deterministic communication model implies a lower bound on the monotone circuit depth for the corresponding monotone Boolean function [9, 29]. However, in the tree-like case good enough lower bounds are known for all models listed in the left column of Table 1. We discuss the strongest model, k-party randomized communication. Typically lower bounds on the communication complexity of Search(φ) are shown for artificial formulas φ that are constructed as follows: take a standard formula ψ and replace each of its variables with a function $g(y_1, y_2, \ldots, y_m)$ (also known as a gadget), where $y_1, y_2, \ldots, y_m$ are fresh variables; the result of this substitution is denoted by $\psi \circ g$. The variables of every gadget are partitioned among k parties. Beame, Pitassi and Segerlind [1] have shown a lower bound on the randomized k-party communication complexity of Search$(T(G) \circ \wedge_k)$, where $T(G)$ is an unsatisfiable Tseitin formula based on a special expander G and $\wedge_k$ is the conjunction of k variables, and the ith party has the ith argument of each instance of $\wedge_k$ written on their forehead.

Huynh and Nordström [12] have introduced a method to obtain a two-party randomized communication complexity lower bound for a search problem via lifting from search problems with large critical block sensitivity. Göös and Pitassi [9] have simplified and generalized this result to multiparty communication complexity and shown that if Search(φ) has large critical block sensitivity and a gadget g has a *versatile* property, then Search$(\varphi \circ g)$ has large randomized communication complexity. Although the construction of versatile functions is somewhat tricky, the proof of the lower bound is much simpler than the proofs from [1, 12].

[1] sometimes it can be improved to $\mathcal{O}(\log S \cdot c)$

There is an established stereotype that lower bounds on the randomized communication complexity of search problems are rather complicated and the resulting lower bounds for proof systems hold only for artificial formulas. In this paper, we break this stereotype and suggest an approach that allows obtaining lower bounds for natural families of formulas by reduction from randomized communication complexity. Moreover, our proofs are elementary.

In the first part of the paper, we demonstrate our method by proving an exponential lower bound on the size of tree-like $\mathrm{Res}(\oplus)$-refutations of the perfect matching principle, while the known lower bound techniques for tree-like $\mathrm{Res}(\oplus)$ do not work for this formula. This lower bound is based on two-party communication complexity. In the second part of the paper, we apply our method to k-party communication complexity and prove a lower bound for communication complexity of $\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{2^n+2^{n(1-1/k)}}\right)$, where $\mathrm{BPHP}_{2^n}^{M}$ denotes the bit pigeonhole principle stating that there are M distinct n-bit strings $s_1, \ldots, s_M$, every string s_i for $i \in [M]$ is partitioned into k almost equal sequential parts and the jth part of every string is written on the forehead of the jth party. In particular, the latter result implies that the bit pigeonhole principle is hard for tree-like $\mathrm{Th}(k)$, so it is the first natural hard instance.

1.2 Search problem $\oplus_k\mathrm{Search}\,(\varphi)$

To achieve our results we use the parity gadget, one of the simplest and the most natural gadgets. We then show how to get rid of this gadget using either properties of a proof system or properties of a family of formulas.

For an unsatisfiable CNF formula φ we define a k-party communication problem $\oplus_k\mathrm{Search}\,(\varphi)$ (usually denoted as $\mathrm{Search}\,(\varphi) \circ \oplus_k$) as follows: for every $i \in [k]$, the ith party has an assignment $\alpha_i \in \mathbb{F}_2^n$ written on the forehead, where n is the number of variables of φ. They are to find a clause of φ that is falsified by the assignment $\sum_{i=1}^{k} \alpha_i$.

It is easy to see that the communication complexity of $\mathrm{Search}\,(\varphi \circ \oplus_k)$ is at least the communication complexity of $\oplus_k\mathrm{Search}\,(\varphi)$, where $\oplus_k$ is the parity of the sum of k bits. However, the formula $\varphi \circ \oplus_k$ may have exponential size if φ contains a wide clause.

In Section 3 we observe the following lemma.

► **Lemma 1.** *If an unsatisfiable CNF-formula φ has a tree-like $\mathrm{Res}\,(\mathrm{PC}_d)$ refutation of size S, then there exists a bounded-error randomized communication protocol for $\oplus_{d+1}\mathrm{Search}\,(\varphi)$ that transmits $\mathcal{O}(d \log S)$ bits.*

1.3 Perfect matching principle in tree-like $\mathrm{Res}(\oplus)$

One of the most important open questions in proof complexity is obtaining a superpolynomial lower bound for bounded-depth Frege with parity gates. $\mathrm{Res}(\oplus)$ is a special case of this system and there are still no known superpolynomial lower bounds for its dag-like version. The first exponential lower bounds for tree-like $\mathrm{Res}(\oplus)$ were proved by Itsykson and Sokolov [14, 15]. Itsykson and Sokolov have shown a lower bound $2^{\Omega(n)}$ on size of tree-like $\mathrm{Res}(\oplus)$ refutations of Pigeonhole Principle (PHP_n^m) for arbitrary $m > n$ using generalized Prover-Delayer games. Oparin in [26] has shown a tight upper bound $2^{\mathcal{O}(n)}$ for such refutations. A lower bound $2^{\Omega(n)}$ for functional pigeonhole principle (FPHP_n^m) for $m = \mathcal{O}(n)$ can be shown using a connection between the size of tree-like $\mathrm{Res}(\oplus)$ refutations and the degree of polynomial calculus refutations (over $\mathbb{F}_2$), observed by Garlik and Kolodziejczyk (see Section 7 of [8]; this method is described in details in [27]; an alternative explanation can be found in [19]). It is also worth mentioning the result of Krajicek (Theorem 18.6.4 from [23]) that formulas encoding Hall's theorem about matchings in bipartite graphs require exponential-size tree-like $\mathrm{Res}(\oplus)$ refutations.

Let PM_G for a graph G encode the existence of a perfect matching in G. Itsykson and Sokolov [14, 15] have shown that for graphs with an odd number of vertices, PM_G has a polynomial-size tree-like $\mathrm{Res}(\oplus)$ refutation. The question about graphs with an even number of vertices remained open; we resolve it in this paper.

Let $K_{m,n}$ be the complete bipartite graphs with parts of size m and n respectively. In Section 4 we prove the following theorem.

▶ **Theorem 2.** *The size of a tree-like* $\mathrm{Res}(\oplus)$ *refutation of* $\mathrm{PM}_{K_{n+2,n}}$ *is* $2^{\Omega(n)}$*.*

Notice that since PHP^m_n is a weakening of $\mathrm{PM}_{K_{m,n}}$, Oparin's upper bound for PHP^m_n [26] implies that the obtained lower bound is tight up to a constant in the exponent.

The formula $\mathrm{PM}_{K_{n+2,n}}$ (however, in a different encoding) has a constant-degree derivation in Nullstellensatz over $\mathbb{F}_2$ [2]. $\mathrm{PM}_{K_{n+2,n}}$ may be refuted as follows: compute the number of edges in the matching modulo 4 in two different ways, on the one hand it is n mod 4 and on the other hand it is $(n+2)$ mod 4. This yields a low-degree Nullstellensatz refutation since the function MOD_4 has a representation as a polynomial of degree 3, see Lemma 8.7 of [2] for details. Thus, Theorem 2 can not be proved via the same reduction to the Polynomial Calculus degree as it can be done for FPHP^m_n.

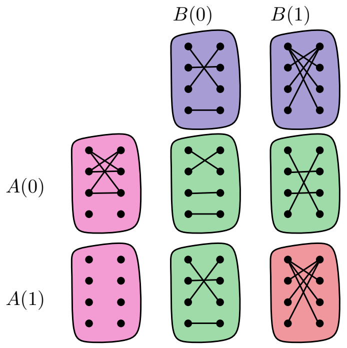
Since $\mathrm{PM}_{K_{n+2,n}}$ has a tree-like Cutting Planes refutation of polynomial size and with polynomial coefficients, the problem $\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)$ has communication complexity $\mathcal{O}(\log n)$ for any partition and thus can not yield a superpolynomial lower bound on size of tree-like $\mathrm{Res}(\oplus)$ refutations. *Therefore the methods previously used to establish tree-like* $\mathrm{Res}(\oplus)$ *lower bounds fail for* $\mathrm{PM}_{K_{n+2,n}}$*.*

To establish this lower bound we employ an idea similar to the one used in [30] to show monotone circuit depth lower bound for matching.

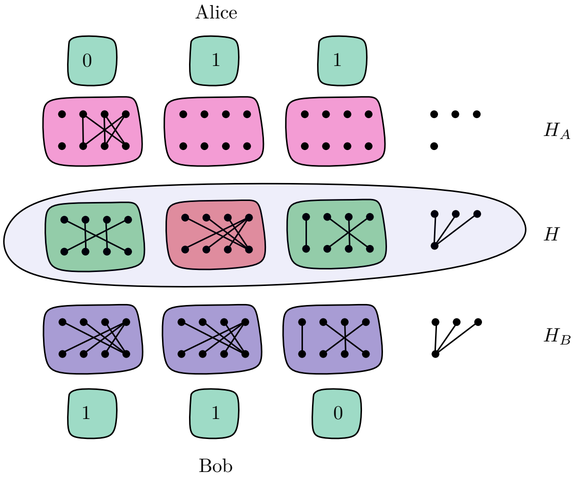
Proof sketch of Theorem 2. By Lemma 1 it is sufficient to show a lower bound $\Omega(n)$ on the two-party bounded-error randomized communication complexity of $\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)$. We show this lower bound via probabilistic reduction from the set disjointness problem. Recall that in the set disjointness problem DISJ_n Alice and Bob have strings $x, y \in \{0,1\}^n$ respectively and they want to verify that there are no $i \in [n]$ such that $x_i = y_i = 1$. It is known that two-party bounded-error randomized communication complexity of DISJ_n is $\Omega(n)$ [16]. Let $G_0(V, E_1)$ and $G_1(V, E_1)$ be graphs on the same set of vertices V; we define $G_0 \oplus G_1$ as a graph on V with edges $E_1 \oplus E_2$, where $\oplus$ denotes the symmetric difference.

We now describe the reduction from DISJ_n to $\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)$. Before starting the communication, each of the parties constructs two graphs: Alice constructs $A(0)$ and $A(1)$, Bob constructs $B(0)$ and $B(1)$ that are shown in Figure 1. These four graphs are bipartite graphs on 8 vertices, 4 vertices in each part and the parts coincide for all the graphs. These graphs have the following property: for $a, b \in \{0,1\}$ the graph $A(a) \oplus B(b)$ is a perfect matching iff at least one of a and b is zero. The graph $A(1) \oplus B(1)$ has two connected components, the first component consists of a single vertex from the first part connected with three vertices from the second part, the second connected component consists of a single vertex from the second part connected with three vertices from the first part.

For each $i \in [n]$ Alice and Bob create new 8 vertices; Alice builds the graph $A(x_i)$ on these vertices and Bob builds the graph $B(y_i)$ on these vertices. Thus, Alice and Bob construct two bipartite graphs G_A and G_B with $4n$ vertices in each part such that $G_A \oplus G_B$ is a perfect matching iff $\mathrm{DISJ}_n(x, y) = 1$. Additionally, Alice and Bob add three vertices to the first part and one vertex to the second part of $G_A \oplus G_B$ connecting the latter with the three vertices added to the first part. Let us denote the resulting graph by H. Let $H = H_A \oplus H_B$, where H_A is known to Alice and H_B is known to Bob. An example of the



■ **Figure 1** The graphs $A(0), A(1), B(0)$, and $B(1)$ and their pairwise symmetric differences. Only $A(1) \oplus B(1)$ is not a matching.



■ **Figure 2** The construction of the graphs H_A, H_B and H for $x = (0,1,1)$; $y = (1,1,0)$.

resulting graphs is shown in Figure 2. Alice and Bob shuffle the vertices in each part of their graphs according to a permutation generated using public random bits and get graphs H'_A and H'_B. As a result, in the shuffled graph $H' = H'_A \oplus H'_B$ the violation of the perfect matching principle artificially added by Alice and Bob is indistinguishable from a violation that appears because of $\mathrm{DISJ}_n(x,y) = 0$. After that Alice and Bob run the communication protocol for $\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{4n+3,4n+1}}\right)$. If the protocol returns a clause corresponding to the artificially added contradiction, Alice and Bob return 1; otherwise, they return 0. By repeating the whole protocol multiple times one can reduce the error probability. ◀

1.4 Bit pigeonhole principle

1.4.1 Bit pigeonhole principle with ⊕-gadget

In Section 5 we apply our lower bound technique for k-party communication in the number-on-forehead model. We consider the bit pigeonhole principle $\mathrm{BPHP}^m_{2^\ell}$ that encodes in CNF that there are m pairwise distinct strings from $\{0,1\}^\ell$. This formula is unsatisfiable for $m > 2^\ell$.

▶ **Theorem 3.** *Let ℓ and k be natural numbers such that $2 \le k \le \ell - 7$. Then the randomized communication complexity of $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}^{2^\ell+2^k}_{2^\ell}\right)$ in the k-party NOF model is $\Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right)$. For $k = 2$ the stronger bound $\Omega\left(2^\ell\right)$ holds.*

Proof idea. The proof follows the same plan as the communication complexity lower bound in Theorem 2. In Subsection 5.1 we consider a decision problem $\mathrm{Distinct}_{k,\ell}$ that is similar to the search problem $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}^{2^\ell}_{2^\ell}\right)$. Let each of k parties have a $2^\ell \times \ell$ matrix over $\mathbb{F}_2$ on the forehead. The goal is to determine whether the rows of the sum of these matrices are distinct. Recall that the unique disjointness $\mathrm{UDISJ}_{k,n}$ is the promise version of the k-party set disjointness: the ith of k parties has a string $x^{(i)}$ from $\{0,1\}^n$ on the forehead, they are to verify that there is no $j \in [n]$ such that $x^{(i)}_j = 1$ for all $i \in [k]$ under the promise that there is at most one such index j. We describe a randomized reduction from $\mathrm{UDISJ}_{k,2^{\ell-k}+1}$ to $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}^{2^\ell}_{2^\ell}\right)$ and then use the known lower bound on the communication complexity of the former problem [32]. First, we reduce $\mathrm{UDISJ}_{k,2^{\ell-k}}$ to the problem $\mathrm{Distinct}_{k,\ell}$: the ith of the parties of the UDISJ protocol generates a matrix D_i of size $2^\ell \times \ell$ such that the matrix $\sum_{i=1}^{k} D_i$ contains a pair of equal rows iff $\mathrm{UDISJ}_{k,2^{\ell-k}}$ evaluates to 0. Moreover, the matrix $\sum_{i=1}^{k} D_i$ has additional properties:

- each of the $2^{\ell-k}$ bits of UDISJ correspond to a block of 2^k rows of the matrix $\sum_{i=1}^{k} D_i$ such that any two rows from different blocks are distinct;
- if the common 1-bit of the inputs of UDISJ has the index $j \in [2^{\ell-k}]$, then the block corresponding to the bit j contains each of its rows exactly twice (all the other blocks have distinct rows).

In Subsections 5.2 and 5.3 we adapt this reduction for $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)$. We add an additional (fake) block to each of the matrices D_i such that the matrix $\sum_{i=1}^{k} D_i$ has the following property: every row of this new block appears in it exactly twice and does not appear anywhere else. Using randomization we make sure that the new artificially added row collisions from the fake block are indistinguishable from the collisions coming from the initial (genuine) blocks corresponding to the bits of UDISJ. Finally, if UDISJ evaluates to 1 then all the collisions are artificially added; if UDISJ evaluates to 0, then with a significant probability the protocol solving $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)$ finds a pair of equal rows coming from a genuine block. ◀

Theorem 3 and Lemma 1 immediately imply the lower bound $\exp\left(\Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right)\right)$ on the size of tree-like $\mathrm{Res}\left(\mathrm{PC}_{k-1}\right)$ refutations of $\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}$ (for $k = 2$ the stronger lower bound $\Omega(2^\ell)$ holds).

1.4.2 Bit pigeonhole without ⊕-gadget

In Section 6 we present a pretty simple and nice reduction from $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^m\right)$ to $\mathrm{Search}\left(\mathrm{BPHP}_{2^{kn}}^{m\cdot 2^{(k-1)n}}\right)$. Here we describe this reduction for $k = 2$. For a larger k the proof is essentially the same. Let us reduce $\oplus_2\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^m\right)$ to $\mathrm{Search}\left(\mathrm{BPHP}_{2^{2n}}^{2^n\cdot m}\right)$. We denote the input of Alice in $\oplus_2\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^m\right)$ as $a_1, \dots, a_m \in \mathbb{F}_2^n$ and the input of Bob as $b_1, \dots, b_m \in \mathbb{F}_2^n$. Their goal is to find a clause of $\mathrm{BPHP}_{2^n}^m$ falsified by the assignment $a_1 + b_1, \dots, a_m + b_m$. Observe that given $i \neq j \in [m]$ such that $a_i + b_i = a_j + b_j$ they can find a falsified clause transmitting additional $\mathcal{O}(n)$ bits. For each $i \in [m]$, Alice and Bob generate 2^n strings from $\mathbb{F}_2^n$: Alice generates $a_i + z$ for each $z \in \mathbb{F}_2^n$ and Bob generates $b_i + z$ for each $z \in \mathbb{F}_2^n$. For each pair of strings $a_i + z$ and $b_i + z$ their sum coincides with $a_i + b_i$. Alice and Bob run the protocol for $\mathrm{Search}\left(\mathrm{BPHP}_{2^{2n}}^{2^n\cdot m}\right)$ on an input where each line has the form $(a_i + z, b_i + z)$ for each $i \in [m]$ and $z \in \mathbb{F}_2^n$. Given a falsified clause of $\mathrm{BPHP}_{2^{2n}}^{2^n\cdot m}$ on this input they determine the lines $(a_i + z, b_i + z)$ and $(a_j + z', b_j + z')$ that are equal to each other. Then $a_i + b_i = a_j + b_j$ and $i \neq j$ since each pair $(i, z) \in [m] \times \mathbb{F}_2^n$ is used by Alice and Bob exactly once.

Together with Theorem 3 this yields the following theorem.

▶ **Theorem 4.** *For $n \geq k(k + 7)$ the randomized k-party communication complexity of* $\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{2^n+2^{n+k-\lfloor n/k\rfloor}}\right)$ *is $\Omega\left(\frac{1}{k}2^{n/2k-3k/2}\right)$, where every string of* BPHP *is partitioned into k almost equal contiguous parts such that jth party has the jth part of every string on its forehead. For $k = 2$ the bound can be improved up to $\Omega\left(2^{n/2}\right)$.*

In particular, Theorem 4 implies the lower bound $\exp\left(2^{\Omega(n/k)}\right)$ on the size of tree-like $\mathrm{T}^{\mathrm{cc}}(k, c)$ (and $\mathrm{Th}(k - 1)$) refutations of $\mathrm{BPHP}_{2^n}^{2^n+2^{n+k-\lfloor n/k\rfloor}}$.

Hrubes and Pudlák [10] proved a lower bound on the complexity of dag-like two-party real communication protocols for $\mathrm{Search}\left(\mathrm{BPHP}^{m}_{2^\ell}\right)$ with the same variable partition, where $m > 2^\ell$ is arbitrary. Formally their and our results are incomparable. On the one hand, the result of Hrubes and Pudlák holds for dag-like protocols and arbitrary weak bit pigeonhole principle, on the other hand, we use a stronger (randomized) model and the statement holds for the multiparty communication as well.

In addition, we show an upper bound on the communication complexity of $\mathrm{Search}\left(\mathrm{BPHP}^{m}_{2^\ell}\right)$. The gap between the upper and the lower bound for $k > 2$ is quadratic. For $k = 2$ the bounds coincide up to a logarithmic factor.

▶ **Proposition 5.** *For $M > 2^n$ and $k \in \{2, 3, \dots, n\}$ there exists a* deterministic *NOF communication protocol for* $\mathrm{Search}\left(\mathrm{BPHP}^{M}_{2^n}\right)$ *with variables partitioned as in Theorem 4 transmitting $\mathcal{O}\left(2^{\lceil n/k \rceil} \cdot \log M\right)$ bits.*

Our lower bound on the k-party communication complexity of $\mathrm{Search}\left(\mathrm{BPHP}^{m}_{n}\right)$ is non-trivial for $k \le \log^{1-\varepsilon} n$ for $\varepsilon > 0$. This lower bound implies a superpolynomial lower bound on the size of tree-like $\mathrm{Th}(k)$-refutations of BPHP^{m}_{n} for such k. We show that there exists a short tree-like $\mathrm{Th}(\log n)$ refutation:

▶ **Proposition 6.** *For $m > 2^\ell$ there exists a tree-like $\mathrm{Th}(\ell)$ refutation of $\mathrm{BPHP}^{m}_{2^\ell}$ of size $\mathcal{O}(m^2 \cdot 2^\ell)$.*

Proposition 6 and the result of Hrubes and Pudlák [10] imply that tree-like $\mathrm{Th}(\log n)$, where n is the number of variables of the refuted formula can not be simulated by *dag-like* $\mathrm{Th}(1)$. Theorem 4 and Proposition 6 imply that the bit pigeonhole principle $\mathrm{BPHP}^{2^\ell+1}_{2^\ell}$ separates [2] tree-like $\mathrm{Th}(\log n)$ from tree-like $\mathrm{Th}(k)$ for $k \le \log^{1-\varepsilon} n$.

1.5 Open questions

1. Is it possible to prove lower bounds on the randomized communication complexity of $\oplus_2\mathrm{Search}\left(\mathrm{PM}_G\right)$ for *constant-degree* graphs G? An $\Omega(n)$ lower bound would improve the best known $\Omega(n/\log n)$ lower bound on the two-party communication complexity of a $\mathrm{Search}\left(\varphi\right)$ problem, where n is the number of variables.
2. Is it true that our results extend to $\mathrm{Res}\left(\mathrm{PC}_d\right)$ over arbitrary finite fields?
3. Is our lower bound for tree-like $\mathrm{Th}(k)$ refutation of $\mathrm{BPHP}^{m}_{2^n}$ tight? Such upper bound would imply a superpolynomial separation between *tree-like* $\mathrm{Th}(k)$ and *dag-like* cutting planes due to the lower bound by [10] as well as separations between tree-like $\mathrm{Th}(k)$ for different values of k.
4. Can we show a lower bound on the communication complexity of the search problem for weaker versions of $\mathrm{BPHP}^{M}_{2^n}$, for example with $M = 2^{n+1}$?

[2] The formula $\mathrm{BPHP}^{2^\ell+1}_{2^\ell}$ uses $n = (2^\ell + 1)\ell$ variables. By Proposition 6, there is a tree-like $\mathrm{Th}(\log n)$ refutation of size $\mathrm{poly}(n)$. By Theorem 4, the size of any tree-like $\mathrm{Th}(\log^{1-\epsilon} n)$ refutation is at least $\exp(\exp(\Omega(\log^{\epsilon} n)))$; the latter grows superpolynomially in n.

2 Preliminaries

Notations

We use the following notation: $[n] = \{1, 2, \ldots, n\}$. Let $S^{n \times m}$ denote the set of matrices of size $n \times m$ with elements from S. We denote by $\mathbf{0}_{n\times m}$ the zero matrix of size $n \times m$ and by $\mathbf{1}_{n\times m}$ the matrix of the same size containing only ones. For square matrices $A_1, \ldots, A_k$ we denote a diagonal block matrix with blocks $A_1, \ldots, A_k$ by $\mathrm{diag}(A_1, \ldots, A_k)$. For $x \in \{0, 1, \ldots, 2^k - 1\}$ we denote a vector $(a_0, \ldots, a_{k-1}) \in \{0, 1\}^k$ such that $x = \sum_{i=0}^{k-1} a_i 2^i$ by $\mathtt{bin}_k(x)$, i.e. $(a_0, \ldots, a_{k-1})$ is the *reversed* binary representation of x. For vectors $v_1, \ldots, v_n$ from a vector space over a field $\mathbb{F}$ we denote their linear span by $\mathrm{Span}(v_1, \ldots, v_n)$.We use coordinate-wise comparison of strings from $\{0, 1\}^n$, i.e. for $x, y \in \{0, 1\}^n$ we write $x \le y$ iff $x_i \le y_i$ for each $i \in [n]$. We denote the set of variables of a CNF-formula φ by $\mathrm{Vars}(\varphi)$.

Communication complexity

We briefly recall some notions of communication complexity. For formal definition and details we refer to [24].

In the classic two-party randomized communication protocol with public randomness, Alice and Bob cooperate to compute a relation $Q \subseteq X \times Y \times Z$: Alice has an input $x \in X$ and Bob has an input $y \in Y$, their goal is to compute $z \in Z$ such that $(x, y, z) \in Q$. We assume that Alice and Bob have access to an arbitrary large random string of bits that is common for Alice and Bob. Let for every $x \in X$ and $y \in Y$, $R^{\delta}_{pub}(Q, x, y)$ denote the minimal number of bits Alice and Bob need to transmit between each other so they both find a $z \in Z$ such that $(x, y, z) \in Q$ with probability at least $1 - \delta$ taken over the values of the common random string. And $R^{\delta}_{pub}(Q) := \max_{x\in X, y\in Y} R^{\delta}_{pub}(Q, x, y)$.

We also consider multiparty communication protocols in the number on forehead (NOF) model that extends two-party protocols for an arbitrary number of parties. In this setting k parties cooperate to compute a relation $Q \subseteq X_1 \times X_2 \times \ldots \times X_k \times Y$. The ith party has $x_i \in X_i$ written on their forehead so they know all x_j for $j \ne i$, their goal is to compute $y \in Y$ such that $(x_1, x_2, \ldots, x_k, y) \in Q$. The parties communicate by taking turns broadcasting messages to all other parties until all parties learn the value of $y \in Y$ such that $(x_1, \ldots, x_k, y) \in Q$. In this model we also assume that all parties have access to a common random string of bits. Let $R^{\delta}_{pub}(Q, x_1, \ldots, x_k)$ for $x_1 \in X_1, \ldots, x_k \in X_k$ denote the minimal total number of bits transmitted until each party learns $y \in Y$ such that $(x_1, \ldots, x_k, y) \in Q$ with probability at least $1 - \delta$ taken over the set of values of the random string of bits. Also, let $R^{\delta}_{pub}(Q) := \max_{x_1\in X_1,\ldots,x_k\in X_k} R^{\delta}_{pub}(Q, x_1, \ldots, x_k)$.

Let f be a function from $X_1 \times X_2 \times \ldots \times X_k \to Y$. Then $R^{\delta}_{pub}(f)$ denotes $R^{\delta}_{pub}(Q_f)$, where $Q_f = \{(x_1, x_2, \ldots, x_k, y) \mid f(x_1, \ldots, x_k) = y\}$.

We prove communication complexity lower bounds by reduction from different versions of the set disjointness problem. $\mathrm{DISJ}_{k,n}$ is a function $\{0, 1\}^{kn} \to \{0, 1\}$ such that for every $x_1, \ldots, x_k \in \{0, 1\}^n$ the following holds: $\mathrm{DISJ}_{k,n}(x_1, \ldots, x_k) = \bigwedge_{j=1}^{n} \underbrace{\neg \left(\bigwedge_{i=1}^{k} (x_i)_j \right)}_{NAND}$.

Let us define the communication promise problem $\mathrm{UDISJ}_{k,n}$ in the k-party NOF model. For each $i \in [k]$ the string x_i is written on the forehead of the ith party, it is guaranteed that there exists at most one index $j \in [n]$ such that for every $i \in [k]$, $(x_i)_j = 1$. The goal is to compute $\mathrm{DISJ}_{k,n}(x_1, \ldots, x_k)$.

▶ **Theorem 7** ([31, 32]). $R^{1/3}_{pub}(\mathrm{UDISJ}_{k,n}) = \Omega\left(\frac{\sqrt{n}}{2^k k}\right)$.

For $k = 2$ we omit the first index: $\mathrm{DISJ}_n = \mathrm{DISJ}_{2,n}$; in this case Theorem 7 may be improved.

▶ **Theorem 8** ([16]). $R^{1/3}_{pub}(\mathrm{DISJ}_n) \geq R^{1/3}_{pub}(\mathrm{UDISJ}_{2,n}) = \Omega(n)$.

Proof complexity

We consider refutational proof systems for the language of unsatisfiable CNF-formulas UNSAT. A refutation of $\varphi \in$ UNSAT in a proof system Π is a sequence of Boolean functions (proof lines) such that each proof line either represents a clause of φ or derived from previous proof lines in the sequence via some sound inference rules. The last line of the proof is identically zero function. A proof system Π is defined by a representation of proof lines and by a set of admissible inference rules. It is required that the inference rules are polynomially verifiable i.e. there exists an algorithm that checks whether it is legitimate to derive a line L_0 from the lines $L_1, \dots, L_k$.

For example, in the Resolution proof lines are represented by clauses and the only inference rule is the resolution rule that allows deriving a clause $A \lor B$ from the clauses $A \lor x$ and $A \lor \neg x$.

The *size* of a proof is the total size of all representations of proof lines in the proof. The *length* of a proof is the number of proof lines in it.

A tree-like proof is such a proof that every its line can be used as a premise of a rule at most once. For each proof system, we can also consider its tree-like version where all proofs are constrained to be tree-like.

We also consider *semantic* refutational proof systems, where we drop the requirement for polynomial verification of inference rules i.e. we allow to derive any sound consequence from the premises. For such systems it is crucial to bound fan-in i.e. the number of the premises from which each proof line can be derived, otherwise, it would be possible to derive a contradiction from the clauses of the initial formula immediately. For example, it is well-known that Resolution is polynomially equivalent to a semantic proof system with fan-in 2 operating with clauses.

A lower bound on the proof size in a semantic proof system implies a lower bound on the proof size in its syntactic counterpart because a syntactic proof is always a semantic proof that operates with the same class of proof lines.

We define semantic $\mathrm{Res}(\oplus)$ as a semantic proof system with fan-in 2 that operates with linear clauses. A linear clause is a disjunction of linear equations over $\mathbb{F}_2$: $\bigvee_{i=1}^{k}(f_i = a_i)$, where f_i is a linear form over $\mathbb{F}_2$ and $a_i \in \mathbb{F}_2$. Notice that an ordinary clause $\bigvee_{i \in P} x_i \lor \bigvee_{j \in N} \neg x_j$ can be represented by the linear clause $\bigvee_{i \in P}(x_i = 1) \lor \bigvee_{j \in N}(x_j = 0)$. For definition of syntactic version of $\mathrm{Res}(\oplus)$ we refer to [15]; it is also proved there that syntactic and semantic $\mathrm{Res}(\oplus)$ are polynomially equivalent.

We define semantic $\mathrm{Res}\,(\mathrm{PC}_d)$ as a semantic proof system with fan-in 2 that operates with disjunctions of equations of type $f = 0$, where f is a degree-d polynomial over $\mathbb{F}_2$. Notice that semantic $\mathrm{Res}\,(\mathrm{PC}_1)$ is exactly semantic $\mathrm{Res}(\oplus)$. For the definition of the syntactic version of $\mathrm{Res}\,(\mathrm{PC}_d)$ we refer to [19].

Following [1] we define $\mathrm{Th}(k)$ as a semantic proof system with fan-in 2 that operates with polynomial inequalities $g \geq 0$, where g is a polynomial of degree at most k with integer coefficients and Boolean variables. A clause $\bigvee_{i \in P} x_i \lor \bigvee_{j \in N} \neg x_j$ can be represented by an inequality $\sum_{i \in P} x_i + \sum_{j \in N}(1 - x_j) - 1 \geq 0$.

Proof complexity and communication complexity

For an unsatisfiable CNF-formula φ we define the communication problem $\mathrm{Search}(\varphi)$. $\mathrm{Search}(\varphi)$ is the following problem: given an assignment of the variables of the unsatisfiable CNF φ, find a clause that is falsified by this assignment. It is assumed that variables of φ are somehow partitioned between the parties.

Following the paper [9] we consider a semantic proof system $\mathrm{T}^{\mathrm{cc}}(k, c)$ that models many interesting syntactic and semantic proof systems. The proof lines in $\mathrm{T}^{\mathrm{cc}}(k, c)$ can be arbitrary Boolean functions having the following property: for every proof line C and every partition of variables of C between k parties, the NOF k-party randomized communication complexity of C is at most c w.r.t. this partition. We also define a semantic proof system $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(k, c)$ that is a subsystem of $\mathrm{T}^{\mathrm{cc}}(k, c)$ with the restriction that a communication protocol for proof lines must have a one-sided error: if the value of a proof line is zero, then the protocol should return zero with probability 1.

For example, $\mathrm{T}^{\mathrm{cc}}(2, 2)$ simulates Resolution; $\mathrm{T}^{\mathrm{cc}}(2, \mathcal{O}(1))$ simulates $\mathrm{Res}(\oplus)$ [22, 15]; $\mathrm{T}^{\mathrm{cc}}(k, \mathcal{O}(k^3 \log^2 n))$, where n is the number of variables in a refuted formula, simulates $\mathrm{Th}(k-1)$ [9]. In Section 3 we show that $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(d+1, \mathcal{O}(1))$ simulates $\mathrm{Res}(\mathrm{PC}_d)$.

The following connection between the communication complexity of $\mathrm{Search}(\varphi)$ and *tree-like* proof complexity of φ is known.

▶ **Lemma 9** ([1, 9]). *If a CNF formula φ has a tree-like $\mathrm{T}^{\mathrm{cc}}(k, c)$ refutation of length ℓ then, over any k-partition of the variables, there is a randomized bounded-error k-party NOF protocol for $\mathrm{Search}(\varphi)$ with communication cost $\mathcal{O}(c \cdot \log \ell \log\log \ell)$.*

In Section 3 we show that for $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(k, c)$ the bound can be improved, see Remark 14.

Basic formulas

A CNF formula PHP^m_n encodes the pigeonhole principle; PHP^m_n states that it is possible to put m pigeons into n holes such that every pigeon flies to at least one hole and at most one pigeon flies to each hole. PHP^m_n depends on variables $p_{i,j}$ for $i \in [m]$ and $j \in [n]$ and $p_{i,j} = 1$ iff the i-th pigeon flies to the j-th hole. PHP^m_n is the conjunction of $\frac{m(m-1)n}{2}$ hole axioms and m pigeons axioms. For every $i \in [m]$ PHP^m_n contains a pigeon axiom $(p_{i,1} \vee p_{i,1} \vee \cdots \vee p_{i,n})$. And for every $j \in [n]$ and every $k \neq \ell \in [n]$, PHP^m_n contains a hole axiom $(\neg p_{k,j} \vee \neg p_{\ell,j})$. PHP^m_n is unsatisfiable iff $m > n$.

For an undirected graph $G(V, E)$, the formula PM_G encodes in CNF that G has a perfect matching. The formula PM_G has $|E|$ variables, each of them corresponds to an edge of G, x_e is the variable corresponding to $e \in E$.

$$\mathrm{PM}_G = \bigwedge_{v \in V} \left(\left(\bigvee_{e \text{ is incident to } v} x_e \right) \wedge \bigwedge_{e_1 \neq e_2 \text{ are incident to } v} (\neg x_{e_1} \vee \neg x_{e_2}) \right).$$

PM_G is unsatisfiable iff G does not have a perfect matching.

▶ **Theorem 10** ([26]). *Let G be a graph with n vertices, which has no perfect matching. Then the formula PM_G has a tree-like $\mathrm{Res}(\oplus)$ refutation of size $2^{\mathcal{O}(n)}$.*

▶ **Proposition 11** ([14]). *Let G be a graph with an odd number of vertices. Then the formula PM_G has a tree-like $\mathrm{Res}(\oplus)$ refutation of size $\mathrm{poly}(n)$.*

The binary pigeonhole principle $\mathrm{BPHP}^m_{2^\ell}$ states that there are m different ℓ-bit binary strings $s_1, s_2, \ldots, s_m$. $\mathrm{BPHP}^m_{2^\ell}$ has $m\ell$ variables corresponding to the bits of s_i for $i \in [m]$. Then $\mathrm{BPHP}^m_{2^\ell} = \bigwedge_{i \neq j \in [m]} s_i \neq s_j$, where the predicate $s_i \neq s_j$ is encoded as a 2ℓ-CNF formula of size 2^ℓ as follows: $\bigwedge_{\alpha\in\{0,1\}^\ell}(s_i \neq \alpha \vee s_j \neq \alpha)$; notice that the predicate $(s_i \neq \alpha \vee s_j \neq \alpha)$ can be represented by a clause with 2ℓ literals. If $m > 2^\ell$, then the formula $\mathrm{BPHP}^m_{2^\ell}$ is unsatisfiable.

Let φ be a CNF formula with n variables, and $g : \{0,1\}^k \to \{0,1\}$ be a Boolean function. Then $\varphi \circ g$ denotes a CNF formula on kn variables that represents $\varphi(g(\overrightarrow{x_1}), g(\overrightarrow{x_2}), \dots, g(\overrightarrow{x_n}))$, where $\overrightarrow{x_i}$ denotes a vector of k new variables. $\varphi \circ g$ is constructed by applying the substitution to every clause C of φ and converting the resulting function $C \circ g$ to CNF in some fixed way.

3 Communication protocols from tree-like $\mathrm{Res}\,(\mathrm{PC}_d)$ proofs

Let φ be an unsatisfiable CNF formula with n variables. Let us define the communication problem $\oplus_k\mathrm{Search}\,(\varphi)$ with k parties as follows. Assume that the ith party has an assignment $\alpha_i \in \{0,1\}^n$ written on the forehead. They aim to find a clause of φ falsified by the assignment $\sum_{i=1}^k \alpha_i$ (all sums of boolean vectors are computed modulo 2).

▶ **Lemma 1.** *Let φ be an unsatisfiable CNF formula. If there exists a tree-like* $\mathrm{Res}\,(\mathrm{PC}_d)$ *proof of φ of length m, then $R^{1/3}_{pub}(\oplus_{d+1}\mathrm{Search}\,(\varphi)) = \mathcal{O}(d \cdot \log m)$.*

A slightly weaker version of the following lemma was implicitly proved in [15]:

▶ **Lemma 12** (see proof of Theorem 3.11 from [15])**.** *Let T be a binary tree with m vertices such that the ith vertex is labeled with $a_i \in \{0,1\}$ with the* hereditary property*: for each inner vertex i with direct descendants c_1 and c_2, if $a_i = 1$, then $a_{c_1} = 1$ or $a_{c_2} = 1$. We also assume that if r is the root of T, then $a_r = 1$. Assume that we have a one-sided bounded error oracle access to a_i i.e. if we request a value of a_i and $a_i = 0$ we get 1 with probability at most $\frac{1}{2}$ and 0 with probability at least $\frac{1}{2}$; if $a_i = 1$ we get 1 with probability 1. Then there exists an algorithm $\mathcal{A}$ that with probability at least $\frac{2}{3}$ returns a leaf ℓ of T with $a_\ell = 1$ and makes $\mathcal{O}(\log m)$ oracle queries to $a_1, \dots, a_m$.*

Proof. See Appendix A. ◀

Proof of Lemma 1. Let $F_1, \dots, F_m$ be a tree-like $\mathrm{Res}\,(\mathrm{PC}_d)$-refutation of φ with the underlying tree T, where vertices of T are identified with $[m]$. Then the leaves of T correspond to the clauses of φ and m is the root of T.

Let $\alpha_1, \dots, \alpha_{d+1}$ be the assignments written on the foreheads of $d+1$ parties. Let $\alpha = \sum_{i=1}^{d+1} \alpha_i$. Let $a_i = 1$ iff α falsifies F_i for $i \in [m]$. Then $a_m = 1$ since F_m is identically false. For any inner node v of T, if $a_v = 1$ then for the direct descendants of v, c_1 and c_2 either $a_{c_1} = 1$ or $a_{c_2} = 1$. In the next paragraphs we show that for any $i \in [m]$ there exists a NOF $(d+1)$-party protocol that computes a_i given that for each $j \in [d+1]$ the jth party has α_j written on their forehead such that

- the protocol transmits $\mathcal{O}(d)$ bits;
- the protocol has one-sided bounded error: if $a_i = 1$ then the protocol returns 1 with probability 1 and if $a_i = 0$ the protocol returns 0 with probability at least $\frac{1}{2}$.

Then we use this protocol to compute a_i as an oracle in the algorithm given by Lemma 12 and thus show that there is a NOF $(d+1)$-party protocol computing $\oplus_{d+1}\mathrm{Search}\,(\varphi)$ with communication cost $\mathcal{O}(d \log m)$.

Now we show that for every $\ell \in [m]$, $F_\ell(\alpha)$ can be computed by a $(d+1)$-party NOF protocol with one-sided error using $\mathcal{O}(d)$ bits of communication. Let $F_\ell = \bigvee_{j=1}^t (f_j = 1)$, where $f_1, \dots, f_t$ are polynomials over $\mathbb{F}_2$ of degree at most d. Let $z_1, \dots, z_n$ be the variables of φ. Let us introduce new variables $y_{1,1}, \dots, y_{1,n}, \dots, y_{d+1,1}, \dots, y_{d+1,n}$ and assume that for each $i \in [d{+}1]$ the ith party has the value of variables $y_{i,1}, y_{i,2}, \ldots, y_{i,n}$ written on the forehead or in other words α_i assigns values of $y_{i,1}, y_{i,2}, \ldots, y_{i,n}$. Let $\bar{f}_j$ denote f_j after substitution $z_\ell := y_{1,\ell} + y_{2,\ell} + \ldots + y_{d+1,\ell}$ for $\ell \in [n]$; $j \in [t]$. Since for all $j \in [t]$, $\deg \bar{f}_j = \deg f_j \le d$, we can represent $\bar{f}_j = \bar{f}_j^{(1)} + \ldots + \bar{f}_j^{(d+1)}$ such that $\bar{f}_j^{(s)}$ does not contain variables $y_{s,1}, \ldots, y_{s,n}$ for each $s \in [d+1]$. Then the ith party can compute $\bar{f}_1^{(i)}(\alpha_1, \ldots, \alpha_{d+1}), \ldots, \bar{f}_t^{(i)}(\alpha_1, \ldots, \alpha_{d+1})$. Notice that $F_\ell = \neg\left(\bigwedge_{j=1}^t (f_j = 0)\right)$.

The final step of the protocol exploits the idea used to construct a short randomized communication protocol for equality. Take a random uniformly distributed vector $(e_1, \ldots, e_t) \in \mathbb{F}_2^t$. Then all parties compute $\sum_{j=1}^t e_j f_j(\alpha) = \sum_{i=1}^{d+1} \underbrace{\sum_{j=1}^{t} e_j \bar{f}_j^{(i)}}_{i\text{th party}}$ with $\mathcal{O}(d)$ bits of communication and the protocol halts.

To bound the error probability we use the following well-known statement:

▶ **Proposition 13** (Random subsum principle). *For any $x \in \mathbb{F}_2^k \setminus \{0^k\}$,*

$$\Pr_{y \leftarrow \mathcal{U}(\mathbb{F}_2^k)}\left[\sum_{i=1}^k y_i x_i = 1\right] = \frac{1}{2}.$$

If $F_\ell(\alpha) = 1$ then $\Pr\left[\sum_{j=1}^t e_j f_j(\alpha) \neq 0\right] = \frac{1}{2}$ by the random subsum principle. If $F_\ell(\alpha) = 0$, then $\Pr\left[\sum_{j=1}^t e_j f_j(\alpha) = 0\right] = 1$. ◀

▶ Remark 14. Similarly to the proof of Lemma 1 one can prove that if an unsatisfiable CNF formula φ has a tree-like $\mathrm{T}^{\mathrm{cc}}_{\mathrm{os}}(k, c)$ refutation of length ℓ, then for any k-partition of the variables, there is a randomized bounded-error k-party NOF protocol for Search (φ) with communication cost $\mathcal{O}(c \log \ell)$. Thus, the bound from Lemma 9 can be slightly improved in the case of one-sided error.

4 Perfect matching

In this section we prove the following theorem:

▶ **Theorem 2.** *The size of any tree-like semantic* $\mathrm{Res}(\oplus)$ *refutation of the formula* $\mathrm{PM}_{K_{n+2,n}}$ *is* $2^{\Omega(n)}$.

By Lemma 1, to prove Theorem 2 it is sufficient to show that $R^{1/3}_{pub}\left(\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)\right) = \Omega(n)$.

Consider the communication problem $\oplus\mathrm{PM}^m_n$ that is defined as follows: Alice and Bob have matrices X and Y over $\mathbb{F}_2$ respectively, each of the matrices has size $m \times n$, where $m \neq n$. Their goal is to find an all-zero row or column or two 1-cells in the same row or column in the matrix $X + Y$.

▶ **Proposition 15.** $R^{1/3}_{pub}\left(\oplus_2\mathrm{Search}\left(\mathrm{PM}_{K_{n+2,n}}\right)\right) \ge R^{1/3}_{pub}(\oplus\mathrm{PM}^{n+2}_n)$.

Proof. A Boolean matrix of size $(n+2) \times n$ naturally corresponds to a subset of edges of $K_{n+2,n}$. A falsified clause encoding that a vertex must be covered by a matching corresponds to an all-zero row or column of the matrix; a falsified clause, encoding that a vertex can not be covered by a matching twice, corresponds to two ones in the same row or column. ◀

Theorem 2 follows from Proposition 15 and the following theorem.

▶ **Theorem 16.** $R^{1/3}_{pub}(\oplus\mathrm{PM}^{n+2}_n) = \Omega(n)$.

Proof. We assume that $n = 4m + 1$, where m is a non-negative integer. If the theorem is true for all n with the residue 1 modulo 4, then it also holds for all other n. Indeed, the protocol for $\oplus\mathrm{PM}^{n+3}_{n+1}$ can be used for $\oplus\mathrm{PM}^{n+2}_n$ by adding to Alice's matrix an extra column and a row with exactly one 1-cell on their intersection and to Bob's matrix an extra column and a row with all zeros.

Let $\mathcal{P}_0$ be a protocol for $\oplus\mathrm{PM}^{n+2}_n$ transmitting at most k bits. We are going to apply $\mathcal{P}_0(X, Y)$ only to the instances where the matrix $X + Y$ does not contain all-zero rows or columns. Thus, we assume that with probability at least 2/3 $\mathcal{P}_0$ returns a tuple $(r_1, c_1, r_2, c_2) \in ([n + 2] \times [n])^2$ such that $(X + Y)_{r_1,c_1} = (X + Y)_{r_2,c_2} = 1$ and either $\begin{cases} r_1 = r_2 \\ c_1 \neq c_2 \end{cases}$ or $\begin{cases} r_1 \neq r_2 \\ c_1 = c_2 \end{cases}$. With $\mathcal{O}(1)$ bits of communication Alice and Bob can verify whether the answer of $\mathcal{P}_0$ is correct and return $\perp$ (failure) if it is not. Also, we can reduce the failure probability by the repetition of the protocol. Let $\mathcal{P}$ be a protocol for $\oplus\mathrm{PM}^{n+2}_n$ under the promise that $X + Y$ does not contain all-zero rows and columns that uses $\mathcal{O}(k)$ bits of communication and returns a correct answer with probability at least $\frac{99}{100}$ and $\perp$ otherwise.

We are going to construct a protocol for DISJ_m transmitting $\mathcal{O}(k)$ bits, where $m = \frac{n-1}{4}$. Since by Theorem 8 any protocol for DISJ_m transmits $\Omega(m)$ bits, we conclude that $k = \Omega(m)$. Let Alice's input for DISJ_m be $a_1, \dots, a_m$ and Bob's input be $b_1, \dots, b_m$.

▶ **Lemma 17.** *There exist matrices $A(0), A(1), B(0), B(1) \in \mathbb{F}_2^{4\times 4}$ such that $A(x) + B(y)$ is a permutation matrix iff $x \wedge y$ is 0 and*

$$A(1) + B(1) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{1}$$

Proof. We simply present matrices that satisfy the conditions:

$$A(0) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \; A(1) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

$$B(0) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \; B(1) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$ ◀

Notice that Lemma 17 immediately allows to reduce DISJ_m to the problem of *checking* whether the sum of Alices and Bobs matrices is a permutation matrix. In order to achieve that, Alice builds a matrix $\mathcal{A} = \mathrm{diag}(A(a_1), \dots, A(a_m))$, Bob builds a matrix $\mathcal{B} = \mathrm{diag}(B(b_1), \dots, B(b_m))$. It is easy to see that $\mathcal{A} + \mathcal{B}$ is a permutation matrix iff $\mathrm{DISJ}_m(a, b) = 1$.

Let us describe the reduction of DISJ_m to $\oplus\mathrm{PM}_n^{n+2}$. Alice and Bob first construct matrices X_0 and Y_0 of the following form:

$$X_0 = \begin{pmatrix} \mathcal{A} & \mathbf{0}_{(n-1)\times 1} \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \end{pmatrix}; \quad Y_0 = \begin{pmatrix} \mathcal{B} & \mathbf{0}_{(n-1)\times 1} \\ \mathbf{0}_{1\times(n-1)} & 0 \\ \mathbf{0}_{1\times(n-1)} & 0 \\ \mathbf{0}_{1\times(n-1)} & 0 \end{pmatrix},$$

then

$$X_0 + Y_0 = \begin{pmatrix} \mathcal{A}+\mathcal{B} & \mathbf{0}_{(n-1)\times 1} \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \\ \mathbf{0}_{1\times(n-1)} & 1 \end{pmatrix},$$

where $\mathcal{A}+\mathcal{B}$ is a permutation matrix iff $\mathrm{DISJ}_m(a,b) = 1$. Then if $\mathcal{P}(X_0, Y_0)$ returns two cells that do not belong to the column n we may conclude that $\mathrm{DISJ}_m(a,b) = 0$. If $\mathcal{P}(X_0, Y_0)$ returns two cells from the nth column, then the value of $\mathrm{DISJ}_m(a,b)$ can not be uniquely determined. Notice that for X_0 and Y_0 constructed as above the protocol always returning $(n+1, n, n+2, n)$ solves $\oplus\mathrm{PM}_n^{n+2}$.

If $\mathrm{DISJ}_m(a,b) = 0$, then the matrix $X_0 + Y_0$ contains at least two columns with three ones and these columns are indistinguishable from each over. To make use of that, we randomly shuffle rows and columns.

We are going to construct a protocol $\mathcal{T}$ for DISJ_m as follows: Alice and Bob choose permutations $\pi \in S_n$, $\tau \in S_{n+2}$ and a matrix $\Delta \in \mathbb{F}_2^{(n+2)\times n}$ uniformly at random. We define matrices $X_0^{\tau,\pi}$ and $Y_0^{\tau,\pi}$ from $\mathbb{F}_2^{n+2\times n}$ such that for each $i \in [n+2]$ and $j \in [n]$, $(X_0^{\tau,\pi})_{i,j} = (X_0)_{\tau(i),\pi(j)}$ and $(Y_0^{\tau,\pi})_{i,j} = (Y_0)_{\tau(i),\pi(j)}$. Alice and Bob run the protocol $\mathcal{P}$ for inputs $X = X_0^{\tau,\pi} + \Delta$, $Y = Y_0^{\tau,\pi} + \Delta$. Notice that $X + Y = X_0^{\tau,\pi} + Y_0^{\tau,\pi}$, thus $X + Y$ can be obtained from $X_0 + Y_0$ by shuffling rows and columns. If $\mathcal{P}(X,Y)$ returns two cells from the column $\pi(n)$, Alice and Bob return 1, if $\mathcal{P}(X,Y)$ returns two cells from other column or row, Alice and Bob return 0. If $\mathcal{P}(X,Y)$ returns $\bot$, then Alice and Bob return $\bot$.

First notice that if $\mathrm{DISJ}_m(a,b) = 1$, then $\mathcal{T}$ returns a correct answer or $\bot$ with probability 1 (and the probability of $\bot$ is at most $\frac{1}{100}$), since in that case $X + Y$ has exactly one column with three 1-cells, each of the other columns and rows contains exactly one 1-cell. Let us fix $a, b \in \{0,1\}^m$ such that $\mathrm{DISJ}_m(a,b) = 0$. We denote $p := \Pr[\mathcal{T}(a,b) = 0]$, we will show that $p \geq \frac{99}{200}$. We can then increase this probability to $2/3$ by repeating the protocol twice (if $\mathcal{T}(a,b)$ returns 0 at least once, we return 0, if $\mathcal{T}(a,b)$ always return $\bot$, we return $\bot$, otherwise we return 1).

Let us describe random bits used by the constructed protocol $\mathcal{T}$. First, we use random bits r to run the protocol $\mathcal{P}$. Second, we use random bits to generate π, τ, and Δ. Since $\mathrm{DISJ}_m(a,b) = 0$, we can fix $i \in [m]$ such that $a_i = b_i = 1$. In that case the submatrix of $X_0 + Y_0$ formed by rows and columns with the indices $4(i-1)+1, 4(i-1)+2, 4(i-1)+3, 4(i-1)+4$ coincides with the matrix (1). Let us denote by $\mathrm{col}(j)$ for $j \in [n]$ the set of all tuples $(x, j, y, j) \in ([n+2] \times [n])^2$.

$$\begin{aligned} p &= \Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y) \notin \mathrm{col}(\pi(n))] - \overbrace{\Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y) = \bot]}^{=:p_\bot} \\ &= 1 - \Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y) \in \mathrm{col}(\pi(n))] - p_\bot \\ &= 1 - \sum_{\pi_0,\tau_0} \Pr_{r,\Delta}[\mathcal{P}_r(X_0^{\tau_0,\pi_0} + \Delta, Y_0^{\tau_0,\pi_0} + \Delta) \in \mathrm{col}(\pi_0(n))] \Pr_{\pi,\tau}[\pi = \pi_0, \tau = \tau_0] - p_\bot \end{aligned}$$

Observe that for fixed π_0 and τ_0 the random variable $(X_0^{\tau_0,\pi_0}+\Delta, Y_0^{\tau_0,\pi_0}+\Delta)$ is uniformly distributed over the pairs of matrices with the sum $X_0^{\tau_0,\pi_0}+Y_0^{\tau_0,\pi_0}$. Let $\alpha \in S_n$ be the transposition swapping n and $4(i-1)+1$. Let $\beta \in S_{n+2}$ be the permutation swapping n and $4(i-1)+2$, $n+1$ and $4(i-1)+3$, $n+2$ and $4(i-1)+3$ (i.e. β is a product of three transpositions). By the construction of α and β, $(X_0+Y_0) = (X_0^{\beta,\alpha}+Y_0^{\beta,\alpha})$, thus $(X_0^{\tau,\pi}+Y_0^{\tau,\pi}) = (X_0^{\tau\circ\beta,\pi\circ\alpha}+Y_0^{\tau\circ\beta,\pi\circ\alpha})$ for every π,τ. Thus the random variable $(X_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta, Y_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta)$ has the same distribution with $(X_0^{\tau_0,\pi_0}+\Delta, Y_0^{\tau_0,\pi_0}+\Delta)$, thus we can continue the sequence as follows:

$$\begin{aligned}
p &= 1 - \sum_{\pi_0,\tau_0} \Pr_{r,\Delta}[\mathcal{P}_r(X_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta, Y_0^{\tau_0\circ\beta,\pi_0\circ\alpha}+\Delta) \in \mathrm{col}(\pi_0(n))] \Pr_{\pi,\tau}[\pi=\pi_0,\tau=\tau_0] - p_\perp \\
&= 1 - \sum_{\pi_0,\tau_0} \Pr_{r,\Delta}[\mathcal{P}_r(X_0^{\tau_0,\pi_0}+\Delta, Y_0^{\tau_0,\pi_0}+\Delta) \in \mathrm{col}(\pi_0\circ\alpha^{-1}(n))] \Pr_{\pi,\tau}[\pi=\pi_0,\tau=\tau_0] - p_\perp \\
&= 1 - \Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y) \in \mathrm{col}((\pi\circ\alpha^{-1})(n))] - p_\perp \\
&\geq 1 - \Pr_{\pi,\tau,\Delta,r}[\mathcal{P}_r(X,Y) \notin \mathrm{col}(\pi(n))] - p_\perp = 1 - p - p_\perp
\end{aligned}$$

Thus, $p \geq 1 - p - p_\perp$ and $p \geq \frac{1-p_\perp}{2} = \frac{99}{200}$. ◀

5 Bit pigeonhole principle with parity gadget

In this section, we prove the following theorem.

▶ **Theorem 3.** *Let ℓ and k be natural numbers such that $2 \leq k \leq \ell - 7$. Then*

$$R_{pub}^{1/3}\left(\oplus_k \mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right)\right) = \Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right).$$

For $k=2$ the stronger bound holds: $R_{pub}^{1/3}\left(\oplus_2 \mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^{2^\ell+4}\right)\right) = \Omega\left(2^\ell\right)$.

We consider a combinatorial analogue of the communication problem $\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^m\right)$. Assume that each of k parties gets m binary strings from $\{0,1\}^\ell$, where $m > 2^\ell$. The ith party has numbers $a_{i,1},\dots,a_{i,m} \in \{0,1\}^\ell$ on their forehead. Based on these strings we form the following set of m vectors from $\mathbb{F}_2^\ell$: $x_1, x_2, \dots, x_m$, where $x_j = \sum_{i=1}^k a_{i,j}$. The goal of the parties is to find a pair of different indices $t,s \in [m]$ such that $x_t = x_s$. We denote this problem by $\oplus_k\mathrm{BPHP}_{2^\ell}^m$. It is straightforward that $R_{pub}^{1/3}\left(\oplus_k\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell}^m\right)\right) \geq R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^m\right)$, hence it is sufficient to prove a lower bound on $R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^m\right)$.

▶ **Theorem 18.** *Let ℓ and k be natural numbers such that $2 \leq k \leq \ell - 7$. Then*

$$R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right) = \Omega\left(R_{pub}^{1/3}\left(\mathrm{UDISJ}_{k,2^{\ell-k}-1}\right) - \ell\right).$$

▶ **Corollary 19.** $R_{pub}^{1/3}\left(\oplus_k\mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}\right) = \Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right)$*. For $k=2$ the stronger bound holds:* $R_{pub}^{1/3}\left(\oplus_2\mathrm{BPHP}_{2^\ell}^{2^\ell+4}\right) = \Omega\left(2^\ell\right)$.

Proof of Corollary 19 . Follows from Theorem 18 and Theorem 7; for $k=2$ we should apply Theorem 8. ◀

Theorem 3 immediately follows from Corollary 19.

5.1 Warm-up example

We start with the simpler statement that, nonetheless, demonstrates the main idea of Theorem 18. Consider the following communication problem $\mathrm{Distinct}_{k,\ell}$: let each of k parties have a matrix from $\mathbb{F}_2^{2^\ell \times \ell}$ on their forehead. The goal is to determine whether all rows of the sum of all these matrices are distinct. A version of this problem without the xor-gadget is referred to as *Element Distinctness* (ED) in the literature [25].

▶ **Proposition 20.** $R^{1/3}_{pub}\left(\mathrm{Distinct}_{k,\ell}\right) \geq R^{1/3}_{pub}\left(\mathrm{UDISJ}_{k,2^{\ell-k}}\right)$.

Let $\mathbb{S}_k$ denote the set of matrices from $\{0,1\}^{2^k \times k}$ with all distinct rows. Let $K_k \in \{0,1\}^{2^k \times k}$ be a matrix such that its ith row equals $\mathtt{bin}_k(i-1-((i-1) \bmod 2))$, i.e. the rows of K_k are $\mathtt{bin}_k(0), \mathtt{bin}_k(0), \mathtt{bin}_k(2), \mathtt{bin}_k(2), \dots, \mathtt{bin}_k(2^{k-1}-2), \mathtt{bin}_k(2^{k-1}-2)$. Notice that every row of K_k starts with zero and appears exactly twice.

In the proof of Proposition 20 as well as in the proof of Theorem 18 we will use the following combinatorial lemma that we prove in Subsection 5.4.

▶ **Lemma 21.** *There exist matrices $A_1(0), A_1(1), \dots, A_k(0), A_k(1) \in \mathbb{F}_2^{2^k \times k}$ such that $\sum_{i=1}^k A_i(1) = K_k$ and for all $b_1, b_2 \dots, b_k \in \{0,1\}$, if $\bigwedge_{i=1}^k b_i = 0$, then $\sum_{i=1}^k A_i(b_i) \in \mathbb{S}_k$.*

Proof of Proposition 20. Let $(x_{i,1}, \dots, x_{i,2^{\ell-k}})$ be an input of the ith party of the problem $\mathrm{UDISJ}_{k,2^{\ell-k}}$. For all $i \in [k]$ we construct a matrix D_i of size $2^\ell \times \ell$ and put it on the forehead of the ith party. Let $A_i(b)$ for $i \in [k]$, $b \in \{0,1\}$ be matrices of size $2^k \times k$ from Lemma 21. Let J_t for $t \in [1, \dots, 2^{\ell-k}]$ be a matrix of size $2^k \times (\ell - k)$ such that all its rows are equal to $\mathtt{bin}_{\ell-k}(t-1)$.

Let us define

$$D_1 := \begin{pmatrix} J_1 & A_1(x_{1,1}) \\ \vdots & \vdots \\ J_j & A_1(x_{1,j}) \\ \vdots & \vdots \\ J_{2^{\ell-k}} & A_1(x_{1,2^{\ell-k}}) \end{pmatrix}; \quad D_i := \begin{pmatrix} & \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,1}) \\ & \vdots & \vdots \\ & \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,j}) \\ \vdots & \vdots & \\ & \mathbf{0}_{2^k \times (\ell-k)} & A_i(x_{i,2^{\ell-k}}) \end{pmatrix} \text{ for } i \in \{2, \dots, k\}.$$

By Lemma 21, the matrix $D_1 + D_2 + \dots + D_k$ has the following property: for all $j \in [2^{\ell-k}]$, its submatrix formed by the rows with numbers from $[2^k \cdot (j-1)+1, 2^k \cdot j]$ has two equal rows if and only if $x_{1,j} = x_{2,j} = \ldots = x_{k,j} = 1$. Thus, the communication complexity of $\mathrm{UDISJ}_{k,2^{\ell-k}}$ is at most the communication complexity of $\mathrm{Distinct}_{k,\ell}$. ◀

5.2 Proof of Theorem 18

In order to prove Theorem 18 we modify the proof of Proposition 20 in order to reduce $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$ to $\oplus_k \mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}$ by adding "fake" rows (such rows do not correspond to the input of the unique disjointness) to matrices $D_1, D_2, \dots, D_k$. We also use some randomization in order to hide "fake" rows among other rows.

Proof of Theorem 18. Let $N > 2^\ell$, consider a k-party communication problem $\mathrm{ROW} \oplus_k \mathrm{BPHP}_{2^\ell}^N$, where ith party has a matrix $M_i \in \mathbb{F}_2^{N \times \ell}$ on their forehead and their goal is to find the value of a row of $M_1 + \dots + M_k$ that appears in this matrix at least twice. The difference with the problem $\oplus_k \mathrm{BPHP}_{2^\ell}^N$ is that we are looking for values of a repeated row rather than numbers of equal rows.

▷ Claim 22. If $R_{1/3}\left(\oplus_k \mathrm{BPHP}^N_{2^\ell}\right) \leq t$, then there exists a communication protocol $\mathcal{P}$ for $\mathrm{ROW} \oplus_k \mathrm{BPHP}^N_{2^\ell}$ using $\mathcal{O}(t+\ell)$ bits of communication such that $\mathcal{P}$ either returns the correct answer or $\perp$ (failure) and $\Pr[\mathcal{P}(M_1, \dots, M_k) = \perp] \leq \frac{1}{100}$ for all input matrices M_i, $i \in [k]$.

Proof. $\mathcal{P}$ executes a randomized protocol for $\oplus_k \mathrm{BPHP}^N_{2^\ell}$ and verifies its answer by transferring additional $\mathcal{O}(\ell)$ bits. The probability of failure can be reduced by repetition. ◁

Let us describe a protocol for the problem $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$ that uses a protocol $\mathcal{P}$ for $\mathrm{ROW} \oplus_k \mathrm{BPHP}^{2^\ell+2^k}_{2^\ell}$ from Claim 22.

Let $x_1, \dots, x_k \in \{0,1\}^{2^{\ell-k}-1}$ be inputs of the communication problem $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$. Let $x_{i,j}$ denote the jth bit of x_i for $i \in [k], j \in [2^{\ell-k}-1]$. Let $\overrightarrow{x} = (x_1, x_2, \dots, x_k)$.

Important matrices

Let γ be a bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$, we define k matrices $D_1(x_1, \gamma)$ and $D_2(x_2), D_3(x_3), \dots, D_k(x_k)$ of size $(2^\ell + 2^k) \times \ell$ similar to Proposition 20.

Let $A_i(b)$ for $i \in [k]$, $b \in \{0,1\}$ be matrices of size $2^k \times k$ from Lemma 21. Let for every $t \in \{0,1\}^{\ell-k}$, J_t be a matrix of size $2^k \times (\ell - k)$ such that all its rows are equal to t. Let W be some fixed matrix from $\mathbb{S}_k$.

We define

$$D_1(x_1,\gamma) := \begin{pmatrix} J_{\gamma(1)} & A_1(x_{1,1}) \\ \vdots & \vdots \\ J_{\gamma(j)} & A_1(x_{1,j}) \\ \vdots & \vdots \\ J_{\gamma(2^{\ell-k}-1)} & A_1(x_{1,2^{\ell-k}-1}) \\ J_{\gamma(*)} & W \\ J_{\gamma(*)} & W \end{pmatrix};$$

and for $i \in [k] \setminus \{1\}$

$$D_i(x_i) := \begin{pmatrix} \mathbf{0}_{2^k\times(\ell-k)} & A_i(x_{i,1}) \\ \vdots & \vdots \\ \mathbf{0}_{2^k\times(\ell-k)} & A_i(x_{i,j}) \\ \vdots & \vdots \\ \mathbf{0}_{2^k\times(\ell-k)} & A_i(x_{i,2^{\ell-k}-1}) \\ \mathbf{0}_{2^k\times(\ell-k)} & \mathbf{0}_{2^k\times k} \\ \mathbf{0}_{2^k\times(\ell-k)} & \mathbf{0}_{2^k\times k} \end{pmatrix}.$$

Notice that the submatrix of $D_1(x_1,\gamma)$ formed by the last 2^{k+1} rows of the matrix $D_1(x_1,\gamma)$ contains every its row exactly two times.

We define $H_{\overrightarrow{x}}(\gamma) := D_1(x_1,\gamma) + D_2(x_2) + \dots + D(x_k)$. By Lemma 21 the matrix $H_{\overrightarrow{x}}(\gamma)$ satisfies the following *key* property w.r.t. $(\gamma, \overrightarrow{x})$ in the standard basis:

▶ **Definition 23.** *Let M be a matrix from $\mathbb{F}_2^{(2^k+2^\ell)\times\ell}$, γ be a bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ and $e_1, e_2, \dots, e_\ell$ be a basis in $\mathbb{F}_\ell$.*

We say that M satisfies the key *property w.r.t $(\gamma, \overrightarrow{x})$ in the basis $(e_1, e_2, \dots, e_\ell)$ if the following properties hold:*

- *If s is a row among the last 2^{k+1} rows of M, then*
  - *the first $\ell - k$ coordinates of s in the basis $(e_1, e_2, \dots, e_\ell)$ are $\gamma(*)_1, \dots \gamma(*)_{\ell-k}$;*
  - *s appears in M exactly twice.*
- *If s is a row of M among the rows with numbers $[2^k(i-1)+1; 2^k i]$ for $i \in [2^{\ell-k}-1]$, then*
  - *the first $\ell - k$ coordinates of s in the basis $(e_1, e_2, \dots, e_\ell)$ are $\gamma(i)_1, \dots, \gamma(i)_{\ell-k}$;*
  - *if $\bigwedge_{j=1}^k x_{i,j} = 0$, then s appears in M exactly once.*
  - *if $\bigwedge_{j=1}^k x_{i,j} = 1$, then s appears in M exactly twice and $(\ell - k + 1)$th coordinate of s in the basis $(e_1, e_2, \dots, e_\ell)$ is 0.*

Consider an invertible matrix $E \in \mathbb{F}_2^{\ell \times \ell}$. Let $e_1, e_2, \dots, e_\ell$ be the rows of E. Since E is invertible, $e_1, e_2, \dots, e_\ell$ form a basis. Let us define $C_{\overrightarrow{x}}(\gamma, E) := H(\overrightarrow{x}, \gamma)E$. Rows of $C_{\overrightarrow{x}}(\gamma, E)$ can be viewed as vectors with coordinates in the basis $e_1, e_2, \dots, e_\ell$ corresponding to the rows of $H(\overrightarrow{x}, \gamma)$. Hence, $C_{\overrightarrow{x}}(\gamma, E)$ satisfies the key property w.r.t. $(\gamma, \overrightarrow{x})$ in the basis $(e_1, e_2, \dots, e_\ell)$.

For a bijection γ from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ and an invertible matrix $E \in \mathbb{F}_2^{\ell \times \ell}$ we define a set $\mathrm{Fake}(\gamma, E) \subseteq \mathbb{F}_2^\ell$ as a set of the last 2^{k+1} rows of the matrix $C_{\overrightarrow{x}}(\gamma, E)$. Notice that by the construction this set does not depend on $\overrightarrow{x}$. By the key property rows from $\mathrm{Fake}(\gamma, E)$ appear exactly twice in $C_{\overrightarrow{x}}(\gamma, E)$.

Random variables

Our protocol uses the following public random variables. In order to distinguish random variables from their values, we highlight random variables in bold.

- $\boldsymbol{\gamma}$ is a random bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ distributed uniformly among all such bijections.
- $\boldsymbol{E}$ is a random invertible matrix from $\mathbb{F}_2^{\ell \times \ell}$ distributed uniformly among all such matrices.
- $\boldsymbol{\pi}$ is a random permutation of the set $[2^\ell + 2^k]$ and $M_{\boldsymbol{\pi}}$ is a permutation matrix of size $(2^\ell + 2^k) \times (2^\ell + 2^k)$ corresponding to the permutation $\boldsymbol{\pi}$ (i.e. $(M_{\boldsymbol{\pi}})_{i,j} = 1 \iff \boldsymbol{\pi}(i) = j$).
- $\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \dots, \boldsymbol{\Delta}_k$ are random matrices from $\mathbb{F}_2^{(2^\ell+2^k)\times \ell}$ distributed uniformly on the set of all matrices $\Delta_1, \Delta_2, \dots, \Delta_k$ such that $\Delta_1 + \Delta_2 + \ldots + \Delta_k$ is the zero matrix.

We define random matrices $\boldsymbol{P}_1, \boldsymbol{P}_2, \dots, \boldsymbol{P}_k$ as follows: $\boldsymbol{P}_i = M_{\boldsymbol{\pi}} \cdot D_i(x_i) \cdot \boldsymbol{E} + \boldsymbol{\Delta}_i$ for $i \geq 2$ and $\boldsymbol{P}_1 = M_{\boldsymbol{\pi}} \cdot D_1(x_1, \boldsymbol{\gamma}) \cdot \boldsymbol{E} + \boldsymbol{\Delta}_1$.

- The addition of $\boldsymbol{\Delta}_i$ makes $\boldsymbol{P}_i$ indistinguishable from the random matrix for every $i \in [k]$.
- $\sum_{i=1}^k \boldsymbol{P}_i = M_{\boldsymbol{\pi}} C_{\overrightarrow{x}}(\boldsymbol{\gamma}, \boldsymbol{E})$ and this matrix is obtained from $C_{\overrightarrow{x}}(\boldsymbol{\gamma}, \boldsymbol{E})$ by the permutation $\boldsymbol{\pi}$ applied to its rows.

Recall that $\mathcal{P}$ is the protocol for $\mathrm{ROW} \oplus_k \mathrm{BPHP}_{2^\ell}^{2^\ell+2^k}$ from Claim 22. Let N be a constant to be chosen later. The protocol $\mathcal{T}$ solving $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$ is described by Algorithm 1.

Protocol analysis

Let us analyze the protocol $\mathcal{T}$. Since it executes the protocol $\mathcal{P}$ a constant number of times, $\mathcal{T}$ transmits $\mathcal{O}(t+\ell)$ bits. Assume that $x_1, x_2, \dots, x_k$ is a 1-instance of $\mathrm{UDISJ}_{k,2^\ell+2^k}$. Then by the key property of $C_{\overrightarrow{x}}(\boldsymbol{\gamma}, \boldsymbol{E})$ all repeated rows of $\sum_{i=1}^k \boldsymbol{P}_i$ are in $\mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})$, hence the protocol $\mathcal{T}$ returns either $\perp$ or the correct answer. Since $\mathcal{P}$ is executed N times independently, the probability that $Z = \{\perp\}$ is at most $\frac{1}{100^N}$, hence $\mathcal{T}$ returns 1 with probability at least $1 - \frac{1}{100^N}$.

The rest of the proof is devoted to the analysis of the case, where $x_1, x_2, \dots, x_k$ is a 0-instance of $\mathrm{UDISJ}_{k,2^\ell+2^k}$. This is the most technically involved part of the proof. So it is a good point to give a **large scale overview of the further proof strategy**. Our goal

Algorithm 1 Protocol $\mathcal{T}$ solving $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$.

```
Input $x_1, x_2, \dots, x_k \in \{0,1\}^{2^{\ell-k}-1}$; $x_i$ is written on the forehead of the $i$th party for every $i \in [k]$.
  $Z := \varnothing$
  loop repeat $N$ times
    Sample $\pi \leftarrow \boldsymbol{\pi}$, $E \leftarrow \boldsymbol{E}$, $\gamma \leftarrow \boldsymbol{\gamma}$, $\vec{\Delta} \leftarrow \vec{\boldsymbol{\Delta}}$     ▷ Use fresh public random bits
    $P_1 := M_\pi \cdot D_1(x_1, \gamma) \cdot E + \Delta_1$     ▷ Can be computed by parties $2, 3, \dots, k$
    $P_i := M_\pi \cdot D_i(x_i) \cdot E + \Delta_i$ for $i \geq 2$     ▷ Can be computed by all parties except the $i$th
    $z := \mathcal{P}(P_1, \dots, P_k)$     ▷ Use fresh random bits for $\mathcal{P}$ and assume that $P_i$ is written on the $i$th party's forehead.
    $Z := Z \cup \{z\}$
  if $Z = \{\bot\}$ then return $\bot$
  else if $Z \setminus \{\bot\} \subseteq \mathrm{Fake}(\gamma, E)$ then return 1     ▷ Intuitively this step means that most likely there are no more repeated rows in $C_{\vec{x}}(\gamma, E)$ except $\mathrm{Fake}(\gamma, E)$ and, hence, $\mathrm{DISJ}(x_1, x_2, \dots, x_k) = 1$ by the key property of $C_{\vec{x}}(\gamma, E)$.
  return 0
```

is to show that if $x_1, x_2, \dots, x_k$ is a 0-instance of $\mathrm{UDISJ}_{k,2^\ell+2^k}$, then the probability that $\mathcal{P}(\boldsymbol{P}_1, \dots, \boldsymbol{P}_k)$ returns a value from $\mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})$ is bounded by some constant less than 1. The random variable $\mathcal{P}(\boldsymbol{P}_1, \dots, \boldsymbol{P}_k)$ depends on random bits used by the protocol $\mathcal{P}$ and on random bits needed for sampling $\boldsymbol{P}_1, \dots, \boldsymbol{P}_k$. Let R denote the set of all random strings used by the protocol $\mathcal{P}$ (i.e. we assume that $\mathcal{P}$ sample a random string from R and use it as public randomness) and S denote the set of all random strings used for sampling $\boldsymbol{P}_1, \dots, \boldsymbol{P}_k$. We would like to construct two bijections α and β on the set S such that for every $s \in S$ the following two properties hold.

1. The three values of random variable $(\boldsymbol{P}_1, \dots, \boldsymbol{P}_k)$ sampled using three strings $s, \alpha(s)$ and $\beta(s)$ as a random source, are the same.
2. Let $(\gamma, E), (\gamma_\alpha, E_\alpha)$ and (γ_β, E_β) be values of the random variable $(\boldsymbol{\gamma}, \boldsymbol{E})$ that is sampled using three strings $s, \alpha(s)$ and $\beta(s)$ as a random source. Then $\mathrm{Fake}(\gamma, E) \cap \mathrm{Fake}(\gamma_\alpha, E_\alpha) \cap \mathrm{Fake}(\gamma_\beta, E_\beta) = \varnothing$.

Consider arbitrary strings $r \in R$ and $s \in S$. The first property implies that for random variables sampled using strings (r, s), $(r, \alpha(s))$ and $(r, \beta(s))$ as a random source values of $\mathcal{P}(\boldsymbol{P_1}, \dots, \boldsymbol{P_k})$ are the same. The second property implies that for at least one of this cases this value does not belong to $\mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})$. Then, using that α and β are bijections, we get $\Pr[\mathcal{P}(\boldsymbol{P_1}, \dots, \boldsymbol{P_k}) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})] \leq \frac{2}{3}$.

Since we have many random variables, it is a tedious task to construct such α and β. In order to simplify this task we slightly relax the properties. We will define bijections α and β not on all strings S but only on the part of bits corresponding to sampling of $\boldsymbol{\gamma}$ and $\boldsymbol{E}$. More precisely we will define two bijections α and β on the set of values of the random variable $(\boldsymbol{\gamma}, \boldsymbol{E})$. We relax the first property as follows:

1'. For every γ and E the three conditional distributions of the random variable $(\boldsymbol{P_1}, \dots, \boldsymbol{P_k})$ under the following three conditions coincide:

a. $(\boldsymbol{\gamma}, \boldsymbol{E}) = (\gamma, E)$,

b. $(\boldsymbol{\gamma}, \boldsymbol{E}) = \alpha(\gamma, E)$ and

c. $(\boldsymbol{\gamma}, \boldsymbol{E}) = \beta(\gamma, E)$.

Unfortunately, we were not able to construct such bijections on the set of all pairs (γ, E). Thus we take a set Ξ consisting $1-\delta$ fraction of all values of $(\boldsymbol{\gamma}, \boldsymbol{E})$ and we will claim that α and β are bijections on Ξ. Such relaxation will weaken the bound of the probability up to $\frac{2}{3}+\delta$. We formalize the requirements to Ξ, α and β in Definition 24. Then we verify in Claim 25 that these requirements are sufficient to bound $\Pr[\mathcal{P}(\boldsymbol{P_1}, \dots, \boldsymbol{P_k}) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})]$. The construction of Ξ, α and β is given in Subsection 5.3.

▶ **Definition 24.** *Let $x_1, \dots, x_k$ be a 0-instance of $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$ and $1 > \delta \geq 0$ be an arbitrary constant. Let Ξ be a set consisting of pairs (γ, E), where γ is a bijection from $[2^{\ell-k}-1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$, E is an invertible matrix from $\mathbb{F}_2^{\ell\times\ell}$. Let α and β be bijections from Ξ to Ξ. We say that (Ξ, α, β) forms a $(1-\delta)$-symmetry randomness space for $\overrightarrow{x}$ if the following conditions hold:*

- *(Largeness)* $\Pr[(\boldsymbol{\gamma}, \boldsymbol{E}) \in \Xi] \geq 1-\delta$.
- *(Difference) For all $(\gamma, E) \in \Xi$,* $\mathrm{Fake}(\gamma, E) \cap \mathrm{Fake}(\alpha(\gamma, E)) \cap \mathrm{Fake}(\beta(\gamma, E)) = \varnothing$.
- *(Symmetry) For all $(\gamma, E) \in \Xi$ the matrices $C_{\overrightarrow{x}}(\gamma, E)$, $C_{\overrightarrow{x}}(\alpha(\gamma, E))$ and $C_{\overrightarrow{x}}(\beta(\gamma, E))$ differ only by a permutation of rows.*

▷ Claim 25. Assume that $x_1, \dots, x_k$ is a 0-instance of $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$, $1 > \delta \geq 0$ is a constant. Let (Ξ, α, β) form a $(1-\delta)$-symmetry randomness space for $\overrightarrow{x}$

Then

$$\Pr\left[\mathcal{P}\left(\boldsymbol{P}_1, \boldsymbol{P}_2, \dots, \boldsymbol{P}_k\right) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})\right] \leq \frac{2}{3} + \delta.$$

Proof. Let us denote $\overrightarrow{\boldsymbol{P}} = (\boldsymbol{P}_1, \boldsymbol{P}_2, \dots, \boldsymbol{P}_k)$, $\overrightarrow{\boldsymbol{\Delta}} = (\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2, \dots, \boldsymbol{\Delta}_k)$ and $\overrightarrow{D}(\overrightarrow{x}, \gamma) = (D_1(x_1, \gamma), D_2(x_2), \dots, D_k(x_k))$.

$\overrightarrow{\boldsymbol{P}} = (\boldsymbol{\Delta}_1 + M_{\boldsymbol{\pi}} D_1(x_1, \boldsymbol{\gamma})\boldsymbol{E}, \boldsymbol{\Delta}_2 + M_{\boldsymbol{\pi}} D_2(x_2)\boldsymbol{E}, \dots, \boldsymbol{\Delta}_k + M_{\boldsymbol{\pi}} D_k(x_k)\boldsymbol{E})$, for brevity we use the vector notation $\overrightarrow{\boldsymbol{P}} = \overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}(\overrightarrow{D}(\overrightarrow{x}, \boldsymbol{\gamma})\boldsymbol{E})$.

Let $p := \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right) \in \mathrm{Fake}(\boldsymbol{\gamma}, \boldsymbol{E})\right]$.

$$p = \sum_{\gamma, E} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\left(\overrightarrow{D}(\overrightarrow{x}, \gamma) \cdot E\right)\right) \in \mathrm{Fake}(\gamma, E)\right] \cdot \Pr[\boldsymbol{\gamma} = \gamma, \boldsymbol{E} = E]$$

$$\overset{\text{(Largeness)}}{\leq} \sum_{(\gamma, E) \in \Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}} \cdot \left(\overrightarrow{D}(\overrightarrow{x}, \gamma) \cdot E\right)\right) \in \mathrm{Fake}(\gamma, E)\right] \cdot \Pr[\boldsymbol{\gamma} = \gamma, \boldsymbol{E} = E] + \delta$$

Notice that for fixed γ, E the random variable $\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}} \cdot \left(\overrightarrow{D}(\overrightarrow{x}, \gamma) \cdot E\right)$ is distributed uniformly on the set of tuples $(L_1, \dots, L_k)$ of k matrices from $\mathbb{F}_2^{(2^\ell+2^k)\times\ell}$ such that $\sum_{i=1}^k L_i$ differs from $C_{\overrightarrow{x}}(\gamma, E)$ only by a permutation of rows. Let $(\gamma_{\alpha^{-1}}, E_{\alpha^{-1}}) = \alpha^{-1}(\gamma, E)$. By the symmetry condition, matrices $C_{\overrightarrow{x}}(\gamma, E)$ and $C_{\overrightarrow{x}}(\gamma_{\alpha^{-1}}, E_{\alpha^{-1}})$ differ only by permutation of rows. Thus, for every set A the probability $\Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}} \cdot \left(\overrightarrow{D}(\overrightarrow{x}, \gamma) \cdot E\right)\right) \in A\right] = \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}} \cdot \left(\overrightarrow{D}(\overrightarrow{x}, \gamma_{\alpha^{-1}}) \cdot E_{\alpha^{-1}}\right)\right) \in A\right]$. Hence,

$$
\begin{aligned}
p &\le \sum_{(\gamma,E)\in\Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\cdot\left(\overrightarrow{D}(\overrightarrow{x},\gamma_{\alpha^{-1}})\cdot E_{\alpha^{-1}}\right)\right)\in \mathrm{Fake}(\gamma,E)\right]\cdot \Pr[\boldsymbol{\gamma}=\gamma,\boldsymbol{E}=E]+\delta\\
&= \sum_{(\gamma,E)\in\Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\cdot\left(\overrightarrow{D}(\overrightarrow{x},\gamma)\cdot E\right)\right)\in \mathrm{Fake}(\alpha(\gamma,E))\right]\cdot \Pr[(\boldsymbol{\gamma},\boldsymbol{E})=\alpha(\gamma,E)]+\delta\\
&= \sum_{(\gamma,E)\in\Xi} \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{\Delta}} + M_{\boldsymbol{\pi}}\cdot\left(\overrightarrow{D}(\overrightarrow{x},\gamma)\cdot E\right)\right)\in \mathrm{Fake}(\alpha(\gamma,E))\right]\cdot \Pr[(\boldsymbol{\gamma},\boldsymbol{E})=(\gamma,E)]+\delta\\
&= \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right)\in \mathrm{Fake}(\alpha(\boldsymbol{\gamma},\boldsymbol{E})),(\boldsymbol{\gamma},\boldsymbol{E})\in\Xi\right]+\delta.
\end{aligned}
$$

Analogously, $p \le \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right)\in \mathrm{Fake}(\beta(\boldsymbol{\gamma},\boldsymbol{E})),(\boldsymbol{\gamma},\boldsymbol{E})\in\Xi\right]+\delta$. Also the inequality $p \le \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right)\in \mathrm{Fake}(\boldsymbol{\gamma},\boldsymbol{E}),(\boldsymbol{\gamma},\boldsymbol{E})\in\Xi\right]+\delta$ follows by the largeness condition. Then,

$$
\begin{aligned}
3(1-p) \ge{}& \Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right)\notin \mathrm{Fake}(\beta(\boldsymbol{\gamma},\boldsymbol{E}))\ \vee\ (\boldsymbol{\gamma},\boldsymbol{E})\notin\Xi\right]\\
&+\Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right)\notin \mathrm{Fake}(\alpha(\boldsymbol{\gamma},\boldsymbol{E}))\ \vee\ (\boldsymbol{\gamma},\boldsymbol{E})\notin\Xi\right]\\
&+\Pr\left[\mathcal{P}\left(\overrightarrow{\boldsymbol{P}}\right)\notin \mathrm{Fake}(\boldsymbol{\gamma},\boldsymbol{E})\ \vee\ (\boldsymbol{\gamma},\boldsymbol{E})\notin\Xi\right]-3\delta\\
\ge{}& \Pr\left[\mathcal{P}\left(\overrightarrow{P}\right)\notin \mathrm{Fake}(\boldsymbol{\gamma},\boldsymbol{E})\cap\mathrm{Fake}(\alpha(\boldsymbol{\gamma},\boldsymbol{E}))\cap\mathrm{Fake}(\beta(\boldsymbol{\gamma},\boldsymbol{E}))\ \vee\ (\boldsymbol{\gamma},\boldsymbol{E})\notin\Xi\right]-3\delta\\
={}&1-3\delta.
\end{aligned}
$$

The last equality follows by the difference condition. Hence, $3(1-p)\ge 1-3\delta$, thus $p\le\frac{2}{3}+\delta$. ◁

We prove the following lemma in Subsection 5.3

▶ **Lemma 26.** *Let $x_1,\dots,x_k$ be a 0-instance of $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$. Then for some $\delta<\frac{1}{3}-\frac{1}{100}$ there exists a $(1-\delta)$-symmetry randomness space for $\overrightarrow{x}$.*

Lemma 26 and Claim 25 imply that there is a constant $\varepsilon>0$ such that

$$\Pr\left[\mathcal{P}\left(\boldsymbol{P}_1,\boldsymbol{P}_2,\dots,\boldsymbol{P}_k\right)\notin\mathrm{Fake}(\boldsymbol{\gamma},\boldsymbol{E})\right]\ge\varepsilon+\frac{1}{100}.$$

Thus,

$$\Pr\left[\mathcal{P}\left(\boldsymbol{P}_1,\boldsymbol{P}_2,\dots,\boldsymbol{P}_k\right)\notin\mathrm{Fake}(\boldsymbol{\gamma},\boldsymbol{E})\cup\{\perp\}\right]\ge\varepsilon.$$

Then, for $N=\mathcal{O}\left(\log\frac{1}{\varepsilon}\right)$, $\mathcal{T}$ gives a correct answer for every 0-instance with probability at least $\frac{2}{3}$. ◀

5.3 Constructions of Ξ,α and β

Proof of Lemma 26. Assume that $x_1,\dots,x_k$ is a 0-instance $\mathrm{UDISJ}_{k,2^{\ell-k}-1}$. Let $i_0\in[2^{\ell-k}-1]$ be such that $x_{1,i_0}=x_{2,i_0}=\ldots=x_{k,i_0}=1$.

Hereinafter γ denotes a bijection from $[2^{\ell-k}-1]\cup\{*\}$ to $\{0,1\}^{\ell-k}$, E denotes an invertible matrix from $\mathbb{F}_2^{\ell\times\ell}$ and $e_1,e_2,\dots,e_\ell$ denote rows of E.

Before presenting constructions of Ξ,α, and β we explain how we are going to establish symmetry and difference properties from Definition 24.

For every $s\in\{0,1\}^{\ell-k}$ and $b\in\{0,1\}$ we introduce the following notation:

$$R(s,b,E):=\left\{(s,b,z)\cdot E\mid z\in\mathbb{F}_2^{k-1}\right\}.$$

Using the key property of the matrix $C_{\vec{x}}(\gamma, E)$ we can describe rows of $C_{\vec{x}}(\gamma, E)$ in terms of $R(s, b, E)$.

▷ Claim 27.

- The set of the last 2^{k+1} rows of $C_{\vec{x}}(\gamma, E)$ is $R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$ and each of this rows appears exactly twice. Recall that we already denote this set as $\mathrm{Fake}(\gamma, E)$. Hence, $\mathrm{Fake}(\gamma, E) = R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$.
- The set of rows of $C_{\vec{x}}(\gamma, E)$ with indices from $[2^k(i-1)+1; 2^k i]$ for $i \in [2^{\ell-k-1}] \setminus \{i_0\}$ is exactly $R(\gamma(i), 0, E) \cup R(\gamma(i), 1, E)$ and every such row appears exactly once.
- The set of rows of $C_{\vec{x}}(\gamma, E)$ with indices from $[2^k(i_0-1)+1; 2^k i_0]$ is exactly $R(\gamma(i_0), 0, E)$ and every such row appears exactly twice.

▷ Claim 28. $R(s, b, E)$ can be represented as a shift of the linear space $\mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell)$:

$$R(s, b, E) = \left(\sum_{j=1}^{\ell-k} s_j e_j + b \cdot e_{\ell-k+1}\right) + \mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell).$$

Proof.

$$R(s, b, E) = \left\{(s, b, z) \cdot E \mid z \in \mathbb{F}_2^{k-1}\right\} = \left\{(s, b, z) \cdot (e_1, e_2, \dots, e_\ell)^T \mid z \in \mathbb{F}_2^{k-1}\right\} =$$
$$\left\{\sum_{i=j}^{\ell-k} s_j e_j + b \cdot e_{\ell-k+1} + \sum_{i=1}^{k-1} z_i e_{\ell-k+1+i} \mid z \in \mathbb{F}_2^{k-1}\right\} =$$
$$\left(\sum_{j=1}^{\ell-k} s_j e_j + b \cdot e_{\ell-k+1}\right) + \mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell).$$

◁

▷ Claim 29. For every $s \in \{0,1\}^{\ell-k}$ and $b \in \{0,1\}$, $|R(s, b, E)| = 2^{k-1}$.

Proof. By Claim 28, $|R(s, b, E)| = \left|\left(\sum_{j=1}^{\ell-k} s_j e_j + b \cdot e_{\ell-k+1}\right) + \mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell)\right| = |\mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell)| = 2^{k-1}$. ◁

▷ Claim 30. Sets $R(s, b, E)$ for $s \in \{0,1\}^{\ell-k}$ and $b \in \{0,1\}$ are disjoint.

Proof. Consider two vectors $u \in R(s, b, E)$ and $v \in R(s', b', E)$ such that $(s, b) \neq (s', b')$. Then, by Claim 28, u and v have different coordinates in the basis $e_1, e_2, \dots, e_\ell$, hence $u \neq v$. ◁

▷ Claim 31. Assume that γ, γ' are bijections from $[2^{\ell-k} - 1] \cup \{*\}$ to $\{0,1\}^{\ell-k}$ and E and E' are invertible matrices from $\mathbb{F}_2^{\ell \times \ell}$ such that

- $R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E) = R(\gamma'(i_0), 0, E') \cup R(\gamma'(*), 0, E') \cup R(\gamma'(*), 1, E')$;
- $R(\gamma(i_0), 1, E) = R(\gamma'(i_0), 1, E')$.

Then matrices $C_{\vec{x}}(\gamma, E)$ and $C_{\vec{x}}(\gamma', E')$ differ only by a permutation of rows.

Proof. By Claim 27, rows from $R(\gamma(i_0), 1, E)$ do not appear in $C_{\vec{x}}(\gamma, E)$, rows from $R(\gamma(i_0), 0, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$ appear in $C_{\vec{x}}(\gamma, E)$ exactly twice. The matrix $C_{\vec{x}}(\gamma, E)$ has $2^\ell + 2^k$ rows. All rows of $C_{\vec{x}}(\gamma, E)$ that are not in $R(\gamma(i_0), 1, E) \cup R(\gamma(*), 0, E) \cup R(\gamma(*), 1, E)$, by Claim 27, appear in $C_{\vec{x}}(\gamma, E)$ exactly once.

By Claims 29 and 30, $|R(\gamma(i_0),0,E) \cup R(\gamma(*),0,E) \cup R(\gamma(*),1,E)| = 3 \cdot 2^{k-1}$, hence, the number of rows of $C_{\vec{x}}(\gamma,E)$ that are not in $R(\gamma(i_0),1,E) \cup R(\gamma(*),0,E) \cup R(\gamma(*),1,E)$ equals $2^\ell - 2^{k+1}$. By Claims 29 and 30, the number of ℓ-bit strings not from $R(\gamma(i_0),1,E) \cup R(\gamma(i_0),0,E) \cup R(\gamma(*),0,E) \cup R(\gamma(*),1,E)$ is also $2^\ell - 2^{k+1}$. Hence, all rows from $\{0,1\}^\ell \setminus (R(\gamma(i_0),0,E) \cup R(\gamma(*),0,E) \cup (\gamma(*),1,E) \cup R(\gamma(i_0),1,E))$ appear in $C_{\vec{x}}(\gamma,E)$ exactly once. Thus, matrices $C_{\vec{x}}(\gamma,E)$ and $C_{\vec{x}}(\gamma',E')$ have the same set of rows and each row appears the same number of times in each of these matrices. ◁

For $\alpha,\beta : \Xi \to \Xi$ we denote $\alpha(\gamma,E) = (\gamma_\alpha,E_\alpha)$ and $\beta(\gamma,E) = (\gamma_\beta,E_\beta)$. We are going to construct α and β such that for all $(\gamma,E) \in \Xi$ the following equalities are satisfied.

$$\begin{cases} R(\gamma(i_0),1,E) &= R(\gamma_\alpha(i_0),1,E_\alpha) &= R(\gamma_\beta(i_0),1,E_\beta); \\ R(\gamma(i_0),0,E) &= R(\gamma_\alpha(*),0,E_\alpha) &= R(\gamma_\beta(*),0,E_\beta); \\ R(\gamma(*),1,E) &= R(\gamma_\alpha(*),1,E_\alpha) &= R(\gamma_\beta(i_0),0,E_\beta); \\ R(\gamma(*),0,E) &= R(\gamma_\alpha(i_0),0,E_\alpha) &= R(\gamma_\beta(*),1,E_\beta). \end{cases} \tag{2}$$

Notice that by Claim 31, equations (2) imply the symmetry property. Equations (2) also imply the difference property. Indeed,

- $\mathrm{Fake}(\gamma,E) = R(\gamma(*),1,E) \cup R(\gamma(*),0,E)$;
- $\mathrm{Fake}(\gamma_\alpha,E_\alpha) = R(\gamma_\alpha(*),1,E_\alpha) \cup R(\gamma_\alpha(*),0,E_\alpha) = R(\gamma(*),1,E) \cup R(\gamma(i_0),0,E)$;
- $\mathrm{Fake}(\gamma_\beta,E_\beta) = R(\gamma_\beta(*),1,E_\beta) \cup R(\gamma_\beta(*),0,E_\beta) = R(\gamma(*),0,E) \cup R(\gamma(i_0),0,E)$.

Hence, by Claim 30, $\mathrm{Fake}(\gamma,E) \cap \mathrm{Fake}(\gamma_\alpha,E_\alpha) \cap \mathrm{Fake}(\gamma_\beta,E_\beta) = \varnothing$.

In order to complete the proof of the lemma we have to construct Ξ and bijections α,β from Ξ to Ξ such that

- (Largeness) $\Pr[(\boldsymbol{\gamma},\boldsymbol{E}) \in \Xi] > \frac{2}{3} + \frac{1}{100}$;
- and for all $(\gamma,E) \in \Xi$ the equations (2) are satisfied.

Definition of Ξ. A pair (γ,E) is in Ξ iff there exist $m,n \in [\ell-k]$ such that $(\gamma(*))_m = 1, (\gamma(i_0))_m = 0$ and $(\gamma(*))_n = 0, (\gamma(i_0))_n = 1$. In other words, $\gamma(*)$ and $\gamma(i_0)$ are not comparable with respect to coordinate-wise comparison.

Notice that $\boldsymbol{\gamma}(i_0)$ and $\boldsymbol{\gamma}(*)$ are distributed uniformly among non-equal elements of $\{0,1\}^{\ell-k}$. Let $\boldsymbol{S}$ and $\boldsymbol{T}$ are two independent random variables distributed uniformly on the set of all subsets of $[\ell-k]$. Then,

$$\begin{aligned} \Pr[(\boldsymbol{\gamma},\boldsymbol{E}) \in \Xi] =& 1 - \Pr[\boldsymbol{\gamma}(i_0) \le \boldsymbol{\gamma}(*) \vee \boldsymbol{\gamma}(*) \le \boldsymbol{\gamma}(i_0)] \ge 1 - 2\Pr[\boldsymbol{\gamma}(i_0) \le \boldsymbol{\gamma}(*)] \\ =& 1 - 2\Pr[\boldsymbol{S} \subseteq \boldsymbol{T} \mid \boldsymbol{S} \ne \boldsymbol{T}] \ge 1 - 2\Pr[\boldsymbol{S} \subseteq \boldsymbol{T}] \\ =& 1 - 2\prod_{j=1}^{\ell-k}(1 - \Pr[j \in \boldsymbol{S} \wedge j \notin \boldsymbol{T}]) = 1 - 2\left(\frac{3}{4}\right)^{\ell-k} > \frac{2}{3} + \frac{1}{100} \text{ if } \ell - k \ge 7. \end{aligned}$$

Hence, the largeness property is satisfied.

Construction of $\boldsymbol{\alpha}$. Let $(\gamma,E) \in \Xi$, we define $\alpha(\gamma,E) = (\gamma_\alpha,E_\alpha)$, where E_α is a matrix with rows defined by vectors $(e'_1,\dots,e'_\ell) = (e_1,\dots,e_{\ell-k},e_{\ell-k+1} + \sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j, e_{\ell-k+2},\dots,e_\ell)$, and

$$\gamma_\alpha(i) = \begin{cases} \gamma(*) & \text{if } i = i_0 \\ \gamma(i_0) & \text{if } i = * \\ \gamma(i) & \text{otherwise} \end{cases}.$$

▷ Claim 32. α is a bijection from $\Xi \to \Xi$.

Proof. Notice that rows of E' form a basis since $\sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j \in \mathrm{Span}(e_1, \dots, e_{\ell-k})$. The mapping $\gamma \mapsto \gamma_\alpha$ is bijective since it just swaps $\gamma(i_0)$ and $\gamma(*)$. Since the condition on $\gamma(i_0)$ and $\gamma(*)$ does not change after application of α, we get that $\alpha(\Xi) \subseteq \Xi$. Notice that $\sum_{j=1}^{\ell-k}(\gamma(i_0)_j + \gamma(*)_j)e_j = \sum_{j=1}^{\ell-k}(\gamma_\alpha(i_0)_j + \gamma_\alpha(*)_j)e'_j$, hence $\alpha(\gamma_\alpha, E_\alpha) = (\gamma, E)$, hence α is bijective. ◁

▷ Claim 33. For all $(\gamma, E) \in \Xi$ the following equalities hold

1. $R(\gamma_\alpha(i_0), 1, E_\alpha) = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\alpha(i_0), 0, E_\alpha) = R(\gamma(*), 0, E)$;
3. $R(\gamma_\alpha(*), 0, E_\alpha) = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\alpha(*), 1, E_\alpha) = R(\gamma(*), 1, E)$.

Proof. We use Claim 28. Let us denote $S := \mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell) = \mathrm{Span}(e'_{\ell-k+2}, \dots, e'_\ell)$.

1. $R(\gamma_\alpha(i_0), 1, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(i_0)_j e'_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(*)_j e_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(i_0)_j e_j + e_{\ell-k+1}\right) + S = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\alpha(i_0), 0, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(i_0)_j e'_j\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(*)_j e_j\right) + S = R(\gamma(*), 0, E)$;
3. $R(\gamma_\alpha(*), 0, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(*)_j e'_j\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(i_0)_j e_j\right) + S = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\alpha(*), 1, E_\alpha) = \left(\sum_{j=1}^{\ell-k} \gamma_\alpha(*)_j e'_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(i_0)_j e_j + e'_{\ell-k+1}\right) + S = \left(\sum_{j=1}^{\ell-k} \gamma(*)_j e_j + e_{\ell-k+1}\right) + S = R(\gamma(*), 1, E)$. ◁

Construction of β. For $(\gamma, E) \in \Xi$, we define $\beta(\gamma, E) = (\gamma_\beta, E_\beta)$, where $\gamma_\beta = \gamma_\alpha$ and E_β is defined below. Let $j_{\min} = \min\{j \in [\ell - k] \colon (\gamma(*))_j = 1 \wedge (\gamma(i_0))_j = 0\}$; $j_{\min}$ is correctly defined since $(\gamma, E) \in \Xi$. Now we define $E_\beta = (e''_1, \dots, e''_\ell)$:

$$e''_j = \begin{cases} e_j & \text{if } j \notin \{j_{\min}, \ell - k + 1\} \\ \sum_{i=1}^{\ell-k}(\gamma(*)_i + \gamma(i_0)_i)e_i & \text{if } j = \ell - k + 1 \\ e_{j_{\min}} + e_{\ell-k+1} & \text{if } j = j_{\min} \end{cases}.$$

▷ Claim 34. β is a bijection from $\Xi \to \Xi$.

Proof. Let us verify that β is *injective*. Given γ_β we can easily recover γ, hence we can recover $j_{\min}$ as well. Then

$$\sum_{i=1}^{\ell-k}(\gamma(i_0)_i + \gamma(*)_i)e''_i + e''_{\ell-k+1} = \sum_{i\in[\ell-k]\setminus\{j_{\min}\}}(\gamma(i_0)_i + \gamma(*)_i)e_i + \overbrace{e_{j_{\min}} + e_{\ell-k+1}}^{e''_{j_{\min}}} + e''_{\ell-k+1}$$
$$= e_{\ell-k+1} + \underbrace{\sum_{i\in[\ell-k]\setminus\{j_{\min}\}}(\gamma(i_0)_i + \gamma(*)_i)e_i + e_{j_{\min}}}_{e''_{\ell-k+1}} + e''_{\ell-k+1} = e_{\ell-k+1}.$$

Thus, we can uniquely recover $e_{\ell-k+1}$ and, hence, also recover $e_{j_{\min}} = e''_{j_{\min}} + e_{\ell-k+1}$; for $j \in [\ell] \setminus \{j_{\min}, \ell - k + 1\}$, $e_j = e''_j$. Hence, β is injective. Notice that since we represent $e_1, \dots, e_\ell$ as linear combinations of $e''_1, \dots, e''_\ell$, then $e''_1, \dots, e''_\ell$ is a basis, hence the matrix E_β is invertible. Thus, we verify that $\beta(\Xi) \subseteq \Xi$ and β is injective, hence β is bijective. ◁

▷ Claim 35. For all $(\gamma, E) \in \Xi$ the following equalities hold

1. $R(\gamma_\beta(i_0), 1, E_\beta) = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\beta(i_0), 0, E_\beta) = R(\gamma(*), 1, E)$;
3. $R(\gamma_\beta(*), 0, E_\beta) = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\beta(*), 1, E_\beta) = R(\gamma(*), 0, E)$;

Proof. We denote $S := \mathrm{Span}(e_{\ell-k+2}, \dots, e_\ell) = \mathrm{Span}(e''_{\ell-k+2}, \dots, e''_\ell)$. Recall that $\gamma(*)_{j_{\min}} = 1$ and $\gamma(i_0)_{j_{\min}} = 0$.

1. $R(\gamma_\beta(i_0), 1, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(i_0)_i e''_i + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + e_{\ell-k+1} + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + e_{\ell-k+1} + \sum_{i=1}^{\ell-k}(\gamma(*)_i + \gamma(i_0)_i)e_i + S = \sum_{i=1}^{\ell-k} \gamma(i_0)_i e_i + e_{\ell-k+1} + S = R(\gamma(i_0), 1, E)$;
2. $R(\gamma_\beta(i_0), 0, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(i_0)_i e''_i + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + e_{\ell-k+1} + S = R(\gamma(*), 1, E)$;
3. $R(\gamma_\beta(*), 0, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(*)_i e''_i + S = \sum_{i=1}^{\ell-k} \gamma(i_0)_i e_i + S = R(\gamma(i_0), 0, E)$;
4. $R(\gamma_\beta(*), 1, E_\beta) = \sum_{i=1}^{\ell-k} \gamma_\beta(*)_i e''_i + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(i_0)_i e_i + e''_{\ell-k+1} + S = \sum_{i=1}^{\ell-k} \gamma(*)_i e_i + S = R(\gamma(*), 0, E)$. ◁

Claims 33 and 35 imply the equations 2. ◀

5.4 Proof of Lemma 21

To prove Lemma 21 it is sufficient to prove the following:

▶ **Proposition 36.** *There exist matrices $T_1, \dots, T_k \in \mathbb{F}_2^{2^k \times k}$, such that*

- *for $\alpha_1, \dots, \alpha_k \in \{0,1\}$ the matrix $\sum_{i=1}^k \alpha_i T_i$ is zero iff $\alpha_1 = \alpha_2 = \ldots = \alpha_k = 0$, i.e. $T_1, \dots, T_k$ are linearly independent;*
- *For every non-zero matrix $M \in \mathrm{Span}(T_1, \dots, T_k)$, $M + K_k \in \mathbb{S}_k$.*

Proof of Lemma 21. Let for $i \in \{1, \dots, k-1\}$, $A_i(0) = T_i$ and $A_i(1)$ be the zero matrix. Let $A_k(0) = K_k + T_k$, $A_k(1) = K_k$. For each $b_1, \dots, b_k \in \{0,1\}$, $\sum_{i=1}^k A_i(b_i) = \sum_{i=1}^k (1 - b_i)T_i + K_k$. Then $\sum_{i=1}^k A_i(1) = K_k$, and if for at least one $i \in [k]$, $b_i \neq 1$, then by the first condition of Proposition 36, $\sum_{i=1}^k (1 - b_i)T_i$ differs from zero, thus by the second condition of Proposition 36, $\sum_{i=1}^k A_i(b_i) \in \mathbb{S}_k$. ◀

Proof of Proposition 36. Let us prove the proposition by induction on k. We are going to prove a stronger statement: namely, we additionally require that for arbitrary non-zero matrix $M \in \mathrm{Span}(T_1, \dots, T_k)$ the set of even-indexed rows of $M + K_k \in \mathbb{S}_k$ coincide with the set of odd-indexed rows of this matrix with all bits flipped.

The base case $k = 1$. $T_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and $K_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. It is easy to verify that all conditions hold.

Induction step from k to $k+1$. Notice that $K_{k+1} = \begin{pmatrix} K_k & \mathbf{0}_{2^k \times 1} \\ K_k & \mathbf{1}_{2^k \times 1} \end{pmatrix}$. Let $T_1, \dots T_k$ be the matrices from induction hypothesis for k. Then define $T'_i = \begin{pmatrix} T_i & \mathbf{0}_{2^k \times 1} \\ T_i & \mathbf{0}_{2^k \times 1} \end{pmatrix}$ for $i \in [k]$ and $T'_{k+1} = \begin{pmatrix} \mathbf{0}_{2^k \times k} & z_0 \\ \mathbf{1}_{2^k \times k} & z_1 \end{pmatrix}$, where $z_0 = (0,1,0,1,\dots,0,1)^T \in \{0,1\}^{2^k \times 1}$, and $z_1 = (1,0,1,0\dots,1,0)^T \in \{0,1\}^{2^k \times 1}$.

Let us verify that all conditions hold. First we show that the matrices $T'_1, T'_2, \dots, T'_{k+1}$ are linearly independent. Matrices $T'_1, T'_2, \dots, T'_k$ are linearly independent since they contain linearly independent blocks $T_1, \dots, T_k$. The matrix T'_{k+1} does not belong to $\mathrm{Span}(T'_1, \dots, T'_k)$, since the last column of T'_{k+1} is non-zero, but the last columns of all $T'_1, \dots, T'_k$ are zeros.

Let us check that for any non-zero matrix $M \in \mathrm{Span}(T'_1, \dots, T'_k, T'_{k+1})$, the condition $M + K_{k+1} \in \mathbb{S}_{k+1}$ holds and the set of even-indexed rows of $M + K_{k+1}$ coincide with the set of odd-indexed rows of this matrix with all bits flipped. Let us analyze the cases:

1. Let M be a non-zero matrix from $\mathrm{Span}(T'_1, \dots, T'_k)$. Then, M has form $\begin{pmatrix} M' & \mathbf{0}_{2^k\times 1} \\ M' & \mathbf{0}_{2^k\times 1} \end{pmatrix}$, where M' is a non-zero matrix from $\mathrm{Span}(T_1, \dots, T_k)$, thus $M' + K_k \in \mathbb{S}_k$. Then $M + K_{k+1} = \begin{pmatrix} M' + K_k & \mathbf{0}_{2^k\times 1} \\ M' + K_k & \mathbf{1}_{2^k\times 1} \end{pmatrix}$; it follows from the induction hypothesis that all rows of this matrix are distinct, i.e. $M + K_{k+1} \in \mathbb{S}_{k+1}$. In order to verify that the set of even-indexed rows of this matrix coincide with the set of odd-indexed rows with all bits flipped, observe that by induction hypothesis the first 2^{k-1} even-indexed rows of $M + K_{k+1}$ coincide with the last 2^{k-1} odd-indexed rows of $M + K_{k+1}$ with all bits flipped, and the first 2^{k-1} odd-indexed rows of $M + K_{k+1}$ coincide with the last 2^{k-1} even-indexed rows of $M + K_{k+1}$ with flipped bits.
2. $M = T'_{k+1}$, then $M + K_{k+1} = \begin{pmatrix} K_k & z_0 \\ \mathbf{1}_{2^k\times k} + K_k & z_0 \end{pmatrix}$. Let us show that all rows of this matrix are distinct. The first 2^k rows start with 0 and are obtained by appending zeroes and ones to the rows of K_k in the alternating order. Since for every pair of coinciding rows of K_k they are adjacent, the first 2^k rows are distinct. The last 2^k rows start from one, so they differ from the first 2^k rows. The proof that they are distinct is the same as for the first 2^k rows. Observe that the $(2i-1)$th row of the matrix $M + K_{k+1}$ coincide with the $(2^k + 2i)$th row of $M + K_{k+1}$ with flipped bits, and the $(2i)$th row of $M + K_{k+1}$ coincide with the $(2^k + 2i - 1)$th row of $M + K_{k+1}$ with flipped bits for $i \in [2^k]$.
3. $M = R + T'_{k+1}$, where R is a non-zero matrix from $\mathrm{Span}(T'_1, \dots, T'_k)$. Let R have the form $\begin{pmatrix} R' & \mathbf{0}_{2^k\times 1} \\ R' & \mathbf{0}_{2^k\times 1} \end{pmatrix}$, where R' is a non-zero matrix from $\mathrm{Span}(T_1, \dots, T_k)$. Then $M + K_{k+1} = R + T'_{k+1} + K_{k+1} = \begin{pmatrix} R' + K_k & z_0 \\ \mathbf{1}_{2^k\times k} + R' + K_k & z_0 \end{pmatrix}$. By the induction hypothesis, $R' + K_k \in \mathbb{S}_k$ and its even-indexed rows coincide with its odd-indexed rows with flipped bits. Then, all even-indexed rows of $M + K_{k+1}$ end with 0, the first 2^{k-1} of them are even-indexed rows of $R' + K_k$ with appended zero, and the last 2^{k-1} of them are even-indexed rows of $R' + K_k$ with all bits flipped and appended 0. Then, by the induction hypothesis, the set of the former rows does not intersect with the set of the latter rows, therefore they are all distinct. By the same argument, all the rows of $M + K_{k+1}$ that end with 1 are distinct. Thus, $M + K_{k+1} \in \mathbb{S}_{k+1}$.
   Let us verify that the set of even-indexed rows of this matrix coincide with the set of odd-indexed rows of this matrix with all bits flipped. Observe that if the ith row of $R' + K_k$ coincides with the jth row of $R' + K_k$ with flipped bits, then the ith row of $M + K_{k+1}$ coincides with its jth row with flipped bits, and the $(2^k + i)$th row of $M + K_{k+1}$ coincides with its $(2^k + j)$th row with all bits flipped. The required property follows from the induction hypothesis. ◀

5.5 Corollaries

▶ **Corollary 37.** *If $k + 7 \le \ell$, then the size of any semantic* $\mathrm{Res}\,(\mathrm{PC}_{k-1})$ *tree-like refutation of* $\mathrm{BPHP}^{2^\ell + 2^k}_{2^\ell}$ *is at least* $2^{\Omega\left(\frac{2^{\ell/2}}{k 2^{3k/2}}\right)}$. *For $k = 2$, the size of any tree-like semantic* $\mathrm{Res}(\oplus)$ *refutation of* $\mathrm{BPHP}^{2^\ell + 4}_{2^\ell}$ *is at least* $2^{\Omega(2^\ell)}$.

Proof. Follows from Theorem 3 and Lemma 1. ◀

▶ **Corollary 38.** *Let $2 \le k \le \ell - 7$ and S be the minimal size of tree-like refutation of $\varphi = \mathrm{BPHP}_{2^\ell}^{2^\ell+2^k} \circ \oplus_k$ in the semantic proof system $\mathrm{T}^{\mathrm{cc}}(k,c)$. Then $\log S \log\log S \ge c \cdot \Omega\left(\frac{2^{\ell/2}}{k2^{3k/2}}\right)$. For $k = 2$, $\log S \log\log S \ge c \cdot \Omega\left(2^\ell\right)$.*

Proof. By Lemma 9, $R_{pub}^{1/3}(\mathrm{Search}\,(\varphi)) = \mathcal{O}\left(\frac{\log S \log\log S}{c}\right)$. We also know that

$$R_{pub}^{1/3}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^\ell+2^k}^{2^\ell} \circ \oplus_k\right)\right) \ge R_{pub}^{1/3}\left(\oplus_k \mathrm{Search}\left(\mathrm{BPHP}_{2^\ell+2^k}^{2^\ell}\right)\right).$$

Now the statement follows from Theorem 3. ◀

6 Bit pigeonhole principle

6.1 Reduction from BPHP $\circ \oplus_k$ to BPHP

Let $T \subseteq X_1 \times X_2 \times \cdots \times X_k \times Y$ and $S \subseteq Z_1 \times Z_2 \times \cdots \times Z_k \times W$ be two relations. We say that S is many-one reducible to T if there are $k+1$ mappings $f_1 : X_1 \to Z_1$, $f_2 : X_2 \to Z_2, \ldots, f_k : X_k \to Z_k$ and $g : W \to Y$ such that if $(f_1(x_1), \ldots, f_k(x_k), y) \in T$ then $(x_1, \ldots, x_k, g(y)) \in S$.

▶ **Lemma 39.** *If S is many-one reducible to T, then $R_{1/3}^{pub}(S) \le R_{1/3}^{pub}(T)$.*

Proof. The ith party computes $f(x_j)$ for all $j \in [k] \setminus \{i\}$ and then all parties run the optimal protocol for T. As soon as all the parties learn an answer y they compute $g(y)$ without communication. ◀

Recall that $\mathrm{BPHP}_{2^n}^M$ encodes that there exist M different strings $s_1, s_2, \ldots, s_M$ from $\{0,1\}^n$. Let k be a positive integer. Let us define the partition Π_k of the variables of $\mathrm{BPHP}_{2^n}^M$ into k parts. Let $n = \ell k + r$ where $0 \le r < k$. For each $i \in [M]$ the row s_i is partitioned into k parts $s = s_i^{(1)} s_i^{(2)} \cdots s_i^{(k)}$ such that $|s_i^{(t)}| = \ell + 1$ if $t \le r$, and $|s_i^{(t)}| = \ell$ if $t > r$. The partition Π_k of the variables of $\mathrm{BPHP}_{2^n}^M$ into k parts is the following: the tth part consists of the variables $s_1^{(t)}, s_2^{(t)}, \ldots, s_M^{(t)}$.

We consider a search problem $\mathrm{SearchPair}_{2^n}^M$: given the values of the variables of $\mathrm{BPHP}_{2^n}^M$, that are partitioned according to Π_k find a pair of distinct indices $i, j \in [M]$, such that the values of s_i and s_j coincide.

▶ **Proposition 40.** *The relation $\mathrm{SearchPair}_{2^n}^M$ is many-one reducible to $\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^M\right)$ with variables partitioned according to Π_k.*

Proof. The proof is straightforward. ◀

▶ **Theorem 41.** *$\oplus_k \mathrm{BPHP}_{2^\ell}^m$ is many-one reducible to $\mathrm{SearchPair}_{2^{k\ell}}^{m \cdot 2^{(k-1)\ell}}$.*

Proof. Let us denote $M = m \cdot 2^{(k-1)\ell}$. Consider a set $Z = \left\{(y_1, y_2, \ldots, y_k) \in (\mathbb{F}_2^\ell)^k \mid \sum_i y_i = 0\right\}$. It is easy to see that $|Z| = 2^{(k-1)\ell}$. Let φ be a bijection between $[M]$ and $Z \times [m]$.

Let for $i \in [m]$ and $t \in [k]$, $x_i^{(t)}$ denote the ith string of the tth party in the communication problem $\oplus_k \mathrm{BPHP}_{2^\ell}^m$. Let $x_i := (x_i^{(1)}, \ldots, x_i^{(k)})$.

For every $t \in [k]$ we define f_t as follows: $f_t\left(x_1^{(t)}, \ldots, x_m^{(t)}\right)$ is a sequence of rows $r_1^{(t)}, r_2^{(t)}, \ldots, r_M^{(t)}$ such that for all $i \in [M]$, $r_i^{(t)} = z_t + x_j^{(t)}$, where $(z, j) = \varphi(i)$ for all $z \in Z$ and $j \in [m]$ (recall that $z \in Z$ is divided on k parts of equal lengths and z_t denotes the tth part).

Let us construct the function g from the definition of the reduction.

Let $q, w \in [M]$ and $q \ne w$. Assume that $\varphi(q) = (z, j_1)$ and $\varphi(w) = (z, j_2)$. We define $g(q, w) := (j_1, j_2)$.

Let us verify that $f_1, f_2, \dots, f_k$ and g define a reduction. Let $q, w \in M$ be a pair of different numbers such that the assignment $\alpha := \left\{ s_i \leftarrow r_i^{(1)} r_i^{(2)} \dots r_i^{(k)} \mid i \in [M] \right\}$ satisfies $s_q = s_w$. Assume that $g(q, w) = (j_1, j_2)$. We need to verify that $j_1 \neq j_2$ and $\sum_{t=1}^{k} x_{j_1}^{(t)} = \sum_{t=1}^{k} x_{j_2}^{(t)}$.

Notice that under the assignment α the value of s_q is $x_{j_1} + z$ and the value of s_w is $x_{j_2} + y$, where $j_1, j_2 \in [m]$ and $z, y \in Z$ such that $(z, j_1) = \varphi(q)$ and $(y, j_2) = \varphi(w)$. If $j_1 = j_2$, then $x_{j_1} + z = x_{j_2} + y$ implies $z = y$. Since φ is a bijection, we get $q = w$. Thus, $j_1 \neq j_2$.

For each $t \in [k]$, the following equality holds.

$$z_t + x_{j_1}^{(t)} = y_t + x_{j_2}^{(t)} \tag{3}$$

If we sum up equations (3) for all $t \in [k]$ and use that $y, z \in Z$, we get $\sum_{t=1}^{k} x_{j_1}^{(t)} = \sum_{t=1}^{k} x_{j_2}^{(t)}$. Hence, (j_1, j_2) is a correct answer for $\oplus_k \mathrm{BPHP}_{2^\ell}^m$. ◀

The following proposition deals with the case, where the number of bits is not divisible by k.

▶ **Proposition 42.** *Let $n = k\ell + r$, where $0 \le r < k$. Let $M > 2^{k\ell}$. Then* $\mathrm{SearchPair}_{2^{k\ell}}^{M}$ *is many-one reducible to* $\mathrm{SearchPair}_{2^n}^{M2^r}$.

Proof. Let $x_1, x_2, \dots, x_M$ be the input of $\mathrm{SearchPair}_{2^{k\ell}}^{M}$, let $x_j^{(t)}$ be the tth part of the row x_j according to the partition Π_k. Given this input we construct an input for $\mathrm{SearchPair}_{2^n}^{M2^r}$. Let τ be a bijection between $[M] \times \{0,1\}^r$ and $[M2^r]$.

For each $i \in [M]$ we construct 2^r rows $y_{\tau(i,\alpha)}$ one for each $\alpha \in \{0,1\}^r$. Let Π_k partition a row $y_{\tau(i,\alpha)}$ into the following parts: $y_{\tau(i,\alpha)}^{(1)} y_{\tau(i,\alpha)}^{(2)} \cdots y_{\tau(i,\alpha)}^{(k)}$. Let

$$y_{\tau(i,\alpha)}^{(t)} = \begin{cases} x_i^{(t)} & \text{if } t > r \\ x_i^{(t)} \alpha_t & \text{if } 0 \le t \le r \end{cases}.$$

Now we can define the function $f_t(x_1^{(t)}, \dots, x_M^{(t)})$ as $y_{\tau(i,\alpha)}^{(t)}$ for each $i \in M$ and $\alpha \in \{0,1\}^r$ and $t \in [k]$ Observe that for each $i \in [M]$ the rows $y_{i,\alpha}$ for $\alpha \in \{0,1\}^r$ are distinct. That allows us to define the function g as $g(\tau(i_1, \alpha_1), \tau(i_2, \alpha_2)) = (i_1, i_2)$. All the required properties can be easily verified. ◀

▶ **Theorem 4.** *Let $M = 2^n + 2^{k+n-\lfloor n/k \rfloor}$ and $n \ge k(k+7)$. If variables of* $\mathrm{BPHP}_{2^n}^{M}$ *are partitioned according Π_k, then* $R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{M}\right)\right) = \Omega\left(\frac{2^{n/2k-3k/2}}{k}\right)$.

For $k = 2$ a stronger bound holds: $R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{M}\right)\right) = \Omega(2^{n/2})$.

Proof. Let $\ell = \lfloor n/k \rfloor$ and $r = n - \ell k$.

$$\begin{aligned}
R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{M}\right)\right) &= R_{1/3}^{pub}\left(\mathrm{Search}\left(\mathrm{BPHP}_{2^n}^{(2^k+2^\ell)2^{(k-1)\ell+r}}\right)\right) \\
&\overset{\text{(Proposition 40)}}{\ge} R_{1/3}^{pub}\left(\mathrm{SearchPair}_{2^n}^{(2^k+2^\ell)2^{(k-1)\ell+r}}\right) \\
&\overset{\text{(Proposition 42)}}{\ge} R_{1/3}^{pub}\left(\mathrm{SearchPair}_{2^{k\ell}}^{(2^k+2^\ell)2^{(k-1)\ell}}\right) \\
&\overset{\text{(Theorem 41)}}{\ge} R_{1/3}^{pub}\left(\oplus_k \mathrm{BPHP}_{2^\ell}^{2^k+2^\ell}\right) \overset{\text{(Corollary 19)}}{=} \Omega\left(\frac{2^{\ell/2-3k/2}}{k}\right) = \Omega\left(\frac{2^{n/2k-3k/2}}{k}\right).
\end{aligned}$$

The case of $k = 2$ can be treated in the same way, the only difference is in the application of Corollary 19. ◀

6.2 Upper bound for communication complexity of $\mathrm{Search}\left(\mathrm{BPHP}^{m}_{2^n}\right)$

▶ **Proposition 5.** *For $M > 2^n$ and $k \in \{2, 3, \dots, n\}$ there exists a* deterministic *NOF communication protocol for* $\mathrm{Search}\left(\mathrm{BPHP}^{M}_{2^n}\right)$ *w.r.t.* Π_k *transmitting* $\mathcal{O}\left(2^{\lceil n/k \rceil} \cdot \log M\right)$ *bits.*

Proof. The protocol is going to have only two active parties: the second party, which we call Alice, and the first party, which we call Bob. We are going to use that Alice can see the variables $s^{(1)}_1, \dots, s^{(1)}_M$ and that Bob can see all other variables.

Let us denote $\bar{s}^{(1)}_i = s^{(2)}_i s^{(3)}_i \dots s^{(k)}_i \in \{0,1\}^{n - \lceil n/k \rceil}$ the bits Bob sees in the ith line for $i \in [M]$. Bob finds a value $\alpha \in \{0,1\}^{n - \lceil n/k \rceil}$ such that the size of the set $S_\alpha = \left\{ i \in [M] \mid \bar{s}^{(1)}_i = \alpha \right\}$ is larger than $2^{\lceil n/k \rceil}$. Such α exists since $M > 2^n$. Bob then picks an arbitrary subset S' of S_α of size $2^{\lceil n/k \rceil} + 1$ and sends the description of S' to Alice using $\left(2^{\lceil n/k \rceil} + 1\right) \cdot \lceil \log_2 M \rceil$ bits. Then, by the pigeonhole principle there exists $i \neq j \in S'$ such that $s^{(1)}_i = s^{(1)}_j$. Alice and Bob then spend $\mathcal{O}(\log M + n)$ bits transmitting indices i and j and all the values of the ith and jth lines to each other. Both of them then find the falsified clause of $\mathrm{BPHP}^{M}_{2^n}$ with no communication because it only depends on variables s_i and s_j and broadcast its description to all of the parties using an additional $\mathcal{O}(n + \log M)$ bits. ◀

For $k = 2$ this upper bound coincides with the lower bound given by Corollary 19 up to a logarithmic factor. For the larger value of k the upper bound and the lower bound are polynomially related. This upper bound shows that the dependence on k in the lower bound is not an artifact of the proof, but a genuine phenomenon.

6.3 Short $\mathrm{Th}(\log n)$ proof of BPHP^{m}_{n}

In this section we give a short tree-like $\mathrm{Th}(\log n)$ refutation of the bit pigeonhole principle BPHP^{m}_{n}. This observation is similar to the one of [5] that converts a resolution proof of the unary encoding of the pigeonhole principle PHP^{m}_{n} to a proof of BPHP^{m}_{n} in $\mathrm{Res}(\log n)$.

Namely we prove the following:

▶ **Proposition 43.** *If there exists a tree-like* $\mathrm{Th}(1)$*-refutation of* $\mathrm{PHP}^{m}_{2^\ell}$ *of size S. Then there exists a tree-like* $\mathrm{Th}(\ell)$*-refutation of* $\mathrm{BPHP}^{m}_{2^\ell}$ *of size $\mathcal{O}(S)$.*

Proof. Let $p_{i,j}$ for $i \in [m]$ and $j \in [2^\ell]$ be a variable of $\mathrm{PHP}^{m}_{2^\ell}$ indicating that the ith pigeon flies to the jth hole. Let $s_{i,k}$ for $i \in [m]$, $k \in [\ell]$ be a variable of $\mathrm{BPHP}^{m}_{2^\ell}$ indicating the ℓth bit of the ith string s_i.

Let $Q_j(x_1, x_2, \dots, x_\ell)$ for $j \in [2^k]$ be a multilinear polynomial over reals such that for all $a_1, a_2, \dots, a_\ell \in \{0,1\}^\ell$, $Q_j(a_1, a_2, \dots, a_\ell) = 1$ if $(a_1, a_2, \dots, a_\ell) = \mathtt{bin}_\ell(j-1)$ and $Q_j(a_1, a_2, \dots, a_\ell) = 0$ otherwise. We ma define Q_j as follows $Q_j(x_1, \dots, x_\ell) = \prod_{k=1}^{\ell}(1 - x_k + \alpha_\ell)$ for $i \in [m]$, $j \in [2^k]$, where $\alpha = \mathtt{bin}_\ell(j-1)$. By the construction $\deg(Q_j) = \ell$.

Let $P_{i,j} = Q_j(s_{i,1}, s_{i,2}, \dots, s_{i,\ell})$.

Consider a tree-like $\mathrm{Th}(1)$-refutation of $\mathrm{PHP}^{m}_{2^\ell}$ of size S: $f_1 \geq 0, f_2 \geq 0, \dots, f_S \geq 0$, where f_i are linear real polynomials over variables $p_{i,j}$ and $f_S \geq 0$ is unsatisfiable on Boolean cube. For each of the inequalities on the following conditions hold:

(a) $f_i \geq 0$ is semantically implied by $f_j \geq 0$ and $f_k \geq 0$ *on the Boolean cube* for $j, k < i$.

(b) f_i is a linear representation of an axiom of $\mathrm{PHP}^{m}_{2^\ell}$;

Let F_i be a polynomial obtained of substitution $p_{j,k} := P_{j,k}$ to f_i for all $j \in [m]$; $k \in [2^\ell]$. Consider a sequence of inequalities $F_1 \geq 0, \dots, F_S \geq 0$. Observe that $F_S \geq 0$ is unsatisfiable on the Boolean cube since $P_{i,j} \in \{0,1\}$ on the Boolean cube. Let us verify that the sequence $F_1 \geq 0, \dots, F_S \geq 0$ may be extended to a correct tree-like $\mathrm{Th}(\ell)$ refutation of $\mathrm{BPHP}^{m}_{2^\ell}$:

(a) If $f_i \geq 0$ is semantically implied by $f_j \geq 0$ and $f_k \geq 0$, then $F_i \geq 0$ is also implied by $F_j \geq 0$ and $F_k \geq 0$, since $P_{i,j}$ is Boolean on the Boolean cube.

(b) If f_i is a linear representation of a

hole axiom then $f_i \geq 0$ is equivalent to the function $(1 - p_{a,b}) + (1 - p_{c,b}) \geq 1$ on $\{0,1\}^{\mathrm{Vars}(\mathrm{PHP}^m_{2^\ell})}$ for $a, c \in [m]$, $b \in [2^\ell]$. Thus $F_i \geq 0$ is also equivalent to $(1 - P_{a,b}) + (1 - P_{c,b}) \geq 1$ on the Boolean cube. Observe that the restriction of $(1 - P_{a,b}) + (1 - P_{c,b}) \geq 1$ to the Boolean cube coincides with the predicate $s_a \neq \mathtt{bin}_\ell(b) \vee s_c \neq \mathtt{bin}_\ell(b)$ which is an axiom of $\mathrm{BPHP}^m_{2^\ell}$.

pigeon axiom then $f_i \geq 0$ is equivalent to $\sum_{j=1}^{2^\ell} p_{a,j} \geq 1$ on the Boolean cube for some $a \in [m]$. Thus $F_i \geq 0$ is equivalent to $\sum_{j=1}^{2^\ell} P_{a,j} \geq 1$ on $\{0,1\}^{\mathrm{Vars}(\mathrm{BPHP}^m_{2^\ell})}$. Observe that the latter inequality is identically true, since $P_{a,j}$ is equivalent to $s_a = \mathtt{bin}_\ell(j-1)$, so for exactly one value of $j \in [2^\ell]$, $P_{a,j} = 1$. Since $F_i \geq 0$ is identically true it can be semantically derived from two arbitrary axioms of $\mathrm{BPHP}^m_{2^\ell}$.

It is easy to see that the size of the resulting refutation is at most $3S$. ◀

▶ **Proposition 44** ([4]). *For $m > n$ there exists a tree-like Cutting Planes (which is a subsystem of* $\mathrm{Th}(1)$*) refutation of* PHP^m_n *of size* $\mathcal{O}(m^2 n)$ *.*

▶ **Proposition 6.** *For $m > 2^\ell$ there exists a tree-like* $\mathrm{Th}(\ell)$ *refutation of* $\mathrm{BPHP}^m_{2^\ell}$ *of size* $\mathcal{O}(m^2 \cdot 2^\ell)$.

Proof. Follows from Propositions 43 and 44. ◀

References

1 Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity. *SIAM J. Comput.*, 37(3):845–869, 2007.

2 Paul Beame and Søren Riis. More on the relative strength of counting principles. In Paul Beam and Samuel R. Buss, editors, *Proof Complexity and Feasible Arithmetics, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, April 21-24, 1996*, volume 39 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 13–35. DIMACS/AMS, 1996. doi:10.1090/dimacs/039/02.

3 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

4 William Cook, Collette R. Coullard, and Gy. Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987.

5 Stefan S. Dantchev, Nicola Galesi, and Barnaby Martin. Resolution and the binary encoding of combinatorial principles. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 6:1–6:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.6.

6 Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. Random Θ(log n)-CNFs are hard for cutting planes. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 109–120. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.19.

7 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 902–911. ACM, 2018. doi:10.1145/3188745.3188838.

8 Michal Garlík and Leszek Aleksander Kolodziejczyk. Some subsystems of constant-depth frege with parity. *ACM Trans. Comput. Log.*, 19(4):29:1–29:34, 2018. doi:10.1145/3243126.

9 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 847–856, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2591796.2591838.

10 Pavel Hrubes and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 121–131. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.20.

11 Pavel Hrubes and Pavel Pudlák. A note on monotone real circuits. *Inf. Process. Lett.*, 131:15–19, 2018. doi:10.1016/j.ipl.2017.11.002.

12 Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 233–248. ACM, 2012. doi:10.1145/2213977.2214000.

13 Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*, pages 220–228. IEEE Computer Society, 1994. doi:10.1109/LICS.1994.316069.

14 Dmitry Itsykson and Dmitry Sokolov. Lower bounds for splittings by linear combinations. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 2014. doi:10.1007/978-3-662-44465-8_32.

15 Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Annals of Pure and Applied Logic*, 171(1), January 2020. doi:10.1016/j.apal.2019.102722.

16 Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discret. Math.*, 5(4):545–557, 1992. doi:10.1137/0405044.

17 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Discret. Math.*, 3(2):255–265, 1990. doi:10.1137/0403021.

18 H. Kesten. An introduction to probability theory and its applications, volume i, (william feller). *SIAM Review*, 11(1):96–96, 1969. doi:10.1137/1011021.

19 Erfan Khaniki. On proof complexity of resolution over polynomial calculus. *Electronic Colloquium on Computational Complexity (ECCC)*, 27:34, 2020. URL: https://eccc.weizmann.ac.il/report/2020/034.

20 Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symb. Log.*, 62(2):457–486, 1997. doi:10.2307/2275541.

21 Jan Krajíček. An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams. *J. Symb. Log.*, 73(1):227–237, 2008. doi:10.2178/jsl/1208358751.

22 Jan Krajíček. Randomized feasible interpolation and monotone circuits with a local oracle. *J. Mathematical Logic*, 18(2):1850012:1–1850012:27, 2018. doi:10.1142/S0219061318500125.

23 Jan Krajíček. *Proof complexity*, volume 170. Cambridge University Press, 2019.

24 Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

25 Edward I Nechiporuk. A boolean function. *Engl. transl. in Sov. Phys. Dokl.*, 10:591–593, 1966.

26 Vsevolod Oparin. Tight upper bound on splitting by linear combinations for pigeonhole principle. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 77–84. Springer, 2016. doi:10.1007/978-3-319-40970-2_6.

27 Fedor Part and Iddo Tzameret. Resolution with counting: Dag-like lower bounds and different moduli. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 19:1–19:37. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.19.

28 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

29 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. doi:10.1007/s004930050062.

30 Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992. doi:10.1145/146637.146684.

31 Alexander A. Sherstov. The multiparty communication complexity of set disjointness. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 525–548, 2012.

32 Alexander A. Sherstov. Communication lower bounds using directional derivatives. *J. ACM*, 61(6), December 2014. doi:10.1145/2629334.

33 Dmitry Sokolov. Dag-like communication and its applications. In *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, pages 294–307, 2017. doi:10.1007/978-3-319-58747-9_26.

A Proof of Lemma 12

▶ **Lemma 12.** *Let T be a binary tree with m vertices such that the ith vertex is labeled with $a_i \in \{0,1\}$ with the* hereditary property*: for each inner vertex i with direct descendants c_1 and c_2, if $a_i = 1$, then $a_{c_1} = 1$ or $a_{c_2} = 1$. We also assume that if r is the root of T, then $a_r = 1$. Assume that we have a one-sided bounded error oracle access to a_i i.e. if we request a value of a_i and $a_i = 0$ we get 1 with probability at most $\frac{1}{2}$ and 0 with probability at least $\frac{1}{2}$; if $a_i = 1$ we get 1 with probability 1. Then there exists an algorithm $\mathcal{A}$ that with probability at least $\frac{2}{3}$ returns a leaf ℓ of T with $a_\ell = 1$ and makes $\mathcal{O}(\log m)$ oracle queries to $a_1, \ldots, a_m$.*

Proof of Lemma 12. For a tree F we denote by $|F|$ the number of nodes in F and for a node v of F we denote by $\textsc{Subtree}(F, v)$ the subtree of F with root v. Let $\texttt{Oracle}(i)$ be the oracle function returning the correct value of a_i with probability at least $\frac{9}{10}$. We can implement such a function using the majority vote of a constant number of initial oracle queries. Let C be a constant; an appropriate value of C we choose later. Consider Algorithm 2 on the following page.

We claim that at any iteration T_i has the hereditary property. This is the case in the beginning and if i decreases at some iteration, then the next T_i was considered at an earlier iteration. Otherwise, the next T_i is either a subtree of the current T_i (in that case the hereditary property is clearly maintained), or is obtained by removal a subtree with 0-labeled root (here we use that the oracle has a one-sided error) from the previous T_i (the hereditary property is also maintained in that case).

We first consider a variant of the algorithm that works infinitely long (i.e., $C = +\infty$) and compute the expected number of the first iteration such that T_i consists of a single 1-labeled leaf of T. Notice that after the first such iteration the value of T_i stays the same for all further iterations. We show that that the expected value is at most $C \log m$ for some constant C. Then by running the algorithm for $3C\lceil\log m\rceil$ iterations we obtain the required error probability by Markov's inequality.

Algorithm 2 Search for 1-leaf.

```
$T_0 := T$                                      ▷ Initialize the tree
$i := 0$
for $j := 1$ to $3C\lceil \log_{3/2} m \rceil$ do
    $r :=$ root of $T_i$
    if $\texttt{Oracle}(r) = 0$ then
        $i := \max\{0, i-1\}$                   ▷ Backtrack since the current tree may not contain a 1-leaf
    else if $|T_i| \neq 1$ then
        $v :=$ a centroid node of $T_i$          ▷ i.e. such that $|\textsc{Subtree}(T_i, v)| \in [\frac{1}{3}|T_i|, \frac{2}{3}|T_i|]$
        if $\texttt{Oracle}(v) = 1$ then
            $T_{i+1} := \textsc{Subtree}(T_i, v)$
        else
            $T_{i+1} := T_i - \textsc{Subtree}(T_i, v)$   ▷ $T_{i+1}$ is obtained from $T_i$ by the deletion of $\textsc{Subtree}(T_i, v)$
        $i := i + 1$
return the only node of $T_i$, if $|T_i| = 1$
```

Let $\texttt{T}(j)$ denote the value of T_i before the start of jth iteration, $i(j)$ denote i at the start of jth iteration and $r(j)$ denote the root of $\texttt{T}(j)$. Notice that if $a_{r(j)} = 1$, then for every $j' > j$, $\texttt{T}(j')$ is a subtree of $\texttt{T}(j)$, since the algorithm never backtracks if the true value of the roots label is 1. Hence, if $a_{r(j)} = a_{r(j')} = 1$ for some $j < j'$, then $i(j) \le i(j')$.

Let us consider a sequence $j_1, j_2, j_3 \ldots$, where $j_1 = 0$, $j_s = \min\{j \mid a_{r(j)} = 1 \wedge j > j_{s-1} \wedge i(j) > i(j_{s-1})\}$, if such minimum exists.

Let us consider the iterations from j_s till $j_{s+1} - 1$. We consider the random variables $Y_{j_s}, Y_{j_s+1}, \ldots Y_{j_{s+1}-1}$ corresponding to these iterations with the following properties:

- If $\texttt{T}(j)$ coincides with $\texttt{T}(j_s)$, then its root is labeled with 1. Then $Y_j = -1$ if the second oracle query returns the correct answer and $Y_j = 1$ if the answer it incorrect. Notice that $\Pr[Y_j = -1] \ge \frac{9}{10}$.
- If the root of $\texttt{T}(j)$ is labeled with zero, then $Y_j = -1$, if the first oracle query returns the correct answer (i.e. the algorithm backtracks). Otherwise, if $\texttt{T}(j)$ consists of a single node $Y_j = 0$. Otherwise, if the root of $\texttt{T}(j+1)$ is labeled with 0, then $Y_j = 1$. If it is labeled with 1, then $Y_j = -\infty$. Notice that $\Pr[Y_j \le -1] \ge \frac{9}{10}$.

Notice that, $j_{s+1} = j_s + \min\{k \mid \sum_{j=j_s}^{j_s+k-1} Y_j \le -1\}$. In order to estimate the expected value of $j_{s+1} - j_s$ we consider an auxiliary random variables $X_{j_s}, X_{j_s+1}, \ldots, X_{j_{s+1}-1}$, defined as $X_j = \begin{cases} 1, & \text{if } Y_j \ge 0 \\ -1, & \text{if } Y_j < 0 \end{cases}$. Notice then $\sum_{j=j_s}^{j_s+k-1} Y_j \le \sum_{j=j_s}^{j_s+k-1} X_j$. We can apply the following fact about random walks in a straight line to the random variables X_j:

▶ **Theorem 45** (Section XII.2 of [18]). *Let $X_1, X_2, \ldots$ be a sequence of independent random variables that take value in $\{-1, 1\}$. Assume that for all i, $\Pr[X_i = 1] \le \frac{1}{10}$ and $\Pr[X_i = -1] \ge \frac{9}{10}$. Let M be a random variable that equals the minimal natural number k such that $\sum_{i=1}^{k} X_i = -1$. Then the expected value of M is at most C, where $C \in \mathbb{R}$ is an absolute constant.*

Fact 45 implies that $\mathbb{E}[j_{s+1} - j_s] \le C$. Then $\mathbb{E}[j_s] = \mathbb{E}[j_s - j_{s-1} + (j_{s-1} - j_{s-2}) + \cdots + (j_2 - j_1) + (j_1 - j_0)] \le sC$. Thus, by Markov's inequality $\Pr[j_s \le 3sC] \ge \frac{2}{3}$. Since $|T_{j_s}| \le \left(\frac{2}{3}\right)^s |T_{j_0}|$, the algorithm that runs for $3C\lceil \log_{3/2} m \rceil$ iterations terminates in a 1-labeled leaf with probability at least $\frac{2}{3}$. ◀

A Lower Bound on Determinantal Complexity

Mrinal Kumar

Department of Computer Science and Engineering, IIT Bombay, India

Ben Lee Volk[1]

Department of Computer Science, University of Texas at Austin, TX, USA

Abstract

The determinantal complexity of a polynomial $P \in \mathbb{F}[x_1, \ldots, x_n]$ over a field $\mathbb{F}$ is the dimension of the smallest matrix M whose entries are affine functions in $\mathbb{F}[x_1, \ldots, x_n]$ such that $P = \mathsf{Det}(M)$. We prove that the determinantal complexity of the polynomial $\sum_{i=1}^{n} x_i^n$ is at least $1.5n - 3$.

For every n-variate polynomial of degree d, the determinantal complexity is trivially at least d, and it is a long standing open problem to prove a lower bound which is super linear in $\max\{n, d\}$. Our result is the first lower bound for any explicit polynomial which is bigger by a constant factor than $\max\{n, d\}$, and improves upon the prior best bound of $n + 1$, proved by Alper, Bogart and Velasco [2] for the same polynomial.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Circuit complexity

Keywords and phrases Determinantal Complexity, Algebraic Circuits, Lower Bounds, Singular Variety

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.4

Related Version *Full Version*: `https://arxiv.org/abs/2009.02452`

Acknowledgements Mrinal thanks Ramprasad Saptharishi for various discussions on determinantal complexity over the years, and in particular for explaining the proof of the result of Mignon and Ressayre to him.

1 Introduction

1.1 Computing with Determinants

The *determinantal complexity* of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, denoted $\mathsf{dc}(f)$, is the minimal integer m such that there exists an affine map $L : \mathbb{F}^n \to \mathbb{F}^{m \times m}$ such that $f(\mathbf{x}) = \mathsf{Det}(L(\mathbf{x}))$, where for every square matrix M, $\mathsf{Det}(M)$ denotes the determinant of M.

This notion was first implicitly defined by Valiant [24], and it is tightly related to the VP vs. VNP problem, the algebraic analog of the P vs. NP problem. The essence of the VP vs. VNP problem is showing that some explicit polynomials are hard to compute. By defining natural notions of reductions and completeness, Valiant showed that this problem is in fact equivalent to showing that, for fields of characteristic different than two, the determinantal complexity of the permanent polynomial,

$$\mathsf{Perm}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)},$$

doesn't grow like a polynomial function in n.[2]

[1] A part of this work was done while at the Center for the Mathematics of Information, California Institute of Technology, USA.

[2] Strictly speaking, the VP vs. VNP question is equivalent to showing that the determinantal complexity of the Perm_n is at least $n^{\omega(\log n)}$, but we skip over this fine grained detail for now.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 4; pp. 4:1–4:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This fact is a consequence of the *completeness* property of the determinant: Valiant showed that if f has an algebraic formula of size s, then the determinantal complexity of f is at most s. This remains true even if f has an *algebraic branching program* (ABP) of size s: ABPs are a natural and more powerful model of computation than formulas. We refer to [22] and [20] for more background on algebraic complexity theory and for proofs of these statements.

Thus, Valiant also established en passant the non-obvious fact that the determinantal complexity of every polynomial is finite, and it's at most roughly $\binom{n+d}{n}$ for every n-variate polynomial of degree d. Standard counting arguments also show that this estimate is close to being tight for almost every such polynomial.

The benefit of this reformulation of the VP vs. VNP problem is that it appears to strip away altogether the notion of "computation": indeed, this problem can be stated without even defining a computational model in any standard sense of the word, and thus it can potentially be proved without having to argue about the topology or structure of every possible arithmetic computation.

In practice, however, proving lower bounds on determinantal complexity is (unsurprisingly) difficult. Currently, for n-variate polynomials, there are no known lower bounds which are super-linear in n (see Subsection 1.2 for more details on previous work). Due to the completeness property mentioned above, a lower bound of s on the determinantal complexity of f will imply the same lower bound for algebraic formulas and even algebraic branching programs. However, super-linear lower bounds for formulas are well-known for decades [12], and super-linear lower bounds for ABPs were recently established in [7], so there doesn't seem to be any major complexity-theoretic barrier for proving such lower bounds for determinantal complexity: the main obstacle is seemingly lack of techniques for reasoning about computations using determinants, and hence it is important to study this model and to develop techniques to understand it and to prove lower bounds, for the permanent as well as for other explicit polynomials.

Even for the purpose of separating VP and VNP, one need not necessarily prove a lower bound on the determinantal complexity of the permanent; the same conclusion will hold if the lower bound on determinantal complexity is shown for any "explicit" polynomial (formally, in the class VNP, which we don't define here) in lieu of the permanent.

Before we describe the previous work concerning determinantal complexity, we provide a brief remark about the notion of a "trivial" lower bound in this context which is worth remembering when evaluating the previous results (and our result). Unlike most standard computational models, observe that for an n-variate polynomial of degree d, even a lower bound of n is non-trivial for determinantal complexity. This is because every coordinate of the affine map L can depend on all n variables. Nevertheless, since the determinant of an $m \times m$ matrix is a degree m polynomial, and thus $\mathsf{Det}(L(\mathbf{x}))$ is a polynomial of degree at most m for every affine map L, the degree d *is* a trivial lower bound on the determinantal complexity of f. Therefore, it is natural to consider polynomial families in which $d \leq n$ or alternatively to hope to prove lower bounds stronger than $\max\{n, d\}$.

1.2 Previous work

The early work on determinantal complexity mostly focused on proving lower bounds for the permanent. Recall that the $n \times n$ permanent, Perm_n, is a degree n polynomial, so the trivial lower bound is $\mathsf{dc}(\mathsf{Perm}_n) \geq n$. Since over characteristic 2 the permanent and determinant coincide, the results described here hold for characteristic not equal to 2.

Already in 1913, Szegő [23], answering a question of Pólya [19], showed that there's no way to generalize the 2 × 2 identity

$$\mathsf{Perm}\begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix} = \mathsf{Det}\begin{pmatrix} x_{1,1} & x_{1,2} \\ -x_{2,1} & x_{2,2} \end{pmatrix}$$

by affixing ± signs to an $n \times n$ matrix of variables for $n \geq 3$.

Marcus and Minc [16] strengthened this result by showing that for every n, $\mathsf{dc}(\mathsf{Perm}_n) > n$. Subsequent work by von zur Gathen [25], Babai and Seress (see [25]), Cai [5] and Meshulam [17] obtained the slightly stronger lower bound $\mathsf{dc}(\mathsf{Perm}_n) \geq \sqrt{2}n$.

Mignon and Ressayre [18] greatly improved the lower bound by proving $\mathsf{dc}(\mathsf{Perm}_n) \geq n^2/2$, over the complex numbers. Cai, Chen and Li [6] extended this lower bound to fields of positive characteristic different than two, and Landsberg, Manivel and Ressayre [15] extended this result to the *border* version of determinantal complexity, that is, they showed that the permanent is not even in the closure of polynomials with determinantal complexity less than $n^2/2$. Finally, Yabe [26] obtained an improved lower bound of $(n-1)^2+1$ over the real numbers.

However, while these lower bounds are quadratic in the degree, Perm_n is a polynomial with n^2 many variables, and notably none of these lower bounds is larger than the number of variables. In particular, these results don't even recover a weak form of the n^3 formula lower bound of Kalorkoti for Perm_n [12].

Landsberg and Ressayre [14] considered determinantal representations that respect certain symmetries (which they called *equivariant determinantal complexity* and denoted edc), and proved that $\mathsf{edc}(\mathsf{Perm}_n)$ is exponential in n. It's unclear how stringent the symmetry requirement is; Ladnsberg and Ressayre put forward the ambitious conjecture that edc and dc are polynomially related, which, if true, would imply VP ≠ VNP. To the best of our knowledge, this conjecture remains open, but it's worth mentioning that in the context of *regular determinantal complexity*, another notion defined and studied by [14], it can be shown unconditionally that requiring symmetry may result in a super-polynomial blow-up [11].

The question of lower bounds for other explicit polynomial was also considered: Mignon and Ressayre [18] proved that the determinantal complexity of quadratic polynomials of rank r is *exactly* $\lceil (r+1)/2 \rceil$ (this, of course, cannot give a lower bound beyond $\lceil (n+1)/2 \rceil$). Chen, Kayal and Wigderson [8] observed that the technique of Mignon and Ressayre implies an $n/2$ lower bound on the determinantal complexity of the elementary symmetric polynomial of degree 2, $\sum_{1 \leq i < j \leq n} x_i x_j$. Kumar [13] used a different technique to prove a similar lower bound for the power symmetric polynomials $\sum_i x_i^d$ for $d \geq 2$ over $\mathbb{C}$.

The last lower bound was improved in a recent work of Alper, Bogart and Velasco [2]: an immediate corollary of their main theorem is that $\mathsf{dc}\left(\sum_i x_i^d\right) \geq n+1$, for every $d \geq 2$. Note that this lower bound is (only slightly) larger than the number of variables n, which is the first lower bound we are aware of with this feature. The results of Alper et al. are more general, and are stated as a function of the co-dimension of the singular locus of the polynomial, a notion we use as well (see Section 3). In particular they are able to prove that $\mathsf{dc}(\mathsf{Perm}_3) = 7$, but their main statement can't imply any lower bound stronger than $n+1$ for an n-variate polynomial.

1.3 Our result

Our main result is the following theorem.

▶ **Theorem 1.** *For every natural number $n \geq 6$, the determinantal complexity of the polynomial $\sum_{i=1}^n x_i^n$ over the field of complex numbers is at least $1.5n - 3$.*

Although for simplicity we state our results for the complex numbers, all the results in this paper also hold for algebraically closed fields of positive characteristic p, as long as p doesn't divide n. This assumption is not only an artifact of the proof. For example, when $n = p^k$, and over characteristic p,

$$\sum_{i=1}^{p^k} x_i^{p^k} = \left(\sum_{i=1}^{p^k} x_i\right)^{p^k}$$

has determinantal complexity at most $n = p^k$; it is also a polynomial of degree n, so its determinantal complexity is at least, and hence equals, n.

As discussed in Subsection 1.2, this is the first non-trivial[3] lower bound of the form $(1+\epsilon)n$, for any $\epsilon > 0$ for any explicit n variate polynomial family, and improves the previous best bound of $n+1$ by Alper, Bogart and Velasco [2] by a constant factor.

This result, of course, is not fully satisfactory. The best upper bound we're aware of for $\mathsf{dc}(\sum_{i=1}^{n} x_i^n)$ is $O(n^2)$, which follows from converting the natural algebraic formula or ABP computing this polynomial to a determinantal expression. We suspect that the true complexity might be $\Omega(n^2)$ or at the very least $\omega(n)$.

Quantitatively, the situation here is somewhat similar to the case of lower bounds on the rank of 3-dimensional tensors, where the best lower bounds are only a constant factor away from the trivial lower bound, and proving super-linear lower bounds remains a challenging open problem (cf. [1, 4, 3, 21], among others).

We now give an outline of the main ideas in our proof.

1.4 Overview of the proof

Let $M \in \mathbb{F}[x_1, x_2, \ldots, x_n]^{m \times m}$ matrix of affine functions such that $\sum_{i=1}^{n} x_i^n = \mathsf{Det}(M(\mathbf{x}))$. Theorem 1 shows a lower bound of $1.5n-3$ on m. There are essentially three main ingredients to the proof of Theorem 1, and we now discuss them in some more detail.

Converting the matrix M into a normal form

Let $M_0 \in \mathbb{F}^{m \times m}$ be the *constant part* of the matrix M, i.e. $M_0 = M(\mathbf{0})$. As a first step of our proof, we show (in Lemma 5) that without loss of generality, M_0 can be assumed to be a diagonal matrix of rank equal to $m-1$. We a say that a matrix M is in *normal form* if it has this additional structure.

It is quite easy to observe that the rank of M_0 is at most $m-1$. However, for technical reasons, we actually need the lower bound on the rank as well, and this fact is a consequence of comparing the dimensions (as algebraic varieties) of the singular locus (which is just the the set of zeroes of a polynomial of multiplicity at least two) of the determinant and that of the polynomial $\sum_{i=1}^{n} x_i^n$. Observations of this nature have been used in the context of determinantal complexity lower bounds before, and indeed, we crucially rely on a well known lemma of von zur Gathen (see Lemma 7) for the proof. The details can be found in Subsection 3.1.

[3] This means that the degree of the polynomials is at most the number of variables.

Determinantal complexity of higher degree polynomial maps

As the key ingredient of our proof, we show that for any matrix $M(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{m\times m}$ where the entries of M are polynomials of degree at most $n-1$ and M is in normal form, if $\mathsf{Det}(M(\mathbf{x})) = \sum_{i=1}^n x_i^n$, then $m \geq n/2$. Moreover, roughly the same lower bound continues to hold as long as $\det(M) = \left(\sum_{i=1}^n x_i^n\right)(1+Q)$ for any polynomial Q, with $Q(\mathbf{0}) = 0$.

Thus, this is a significant generalization of the $n/2$ lower bound on the standard notion determinantal complexity (where the entries of M are affine functions) of $\sum_{i=1}^n x_i^n$ as shown in [13]: this shows that roughly the same lower bound continues to hold even when the entries of the matrix are arbitrary polynomials of degree as high as $n-1$ and the determinant of the matrix equals an arbitrary multiple of $\sum_{i=1}^n x_i^n$ with a non-zero constant term.

The proof of the lemma relies on the observation that the polynomial $\sum_{i=1}^n x_i^n$ does not vanish with multiplicity at least two very often. This seemingly simple observation has been previously used in the context of lower bounds on algebraic branching programs computing this polynomial [13, 7] in a crucial way. See Subsection 3.2 for further details.

Trading dimension of the matrix for degree

As the final ingredient of our proof, we use a well known property of determinants (Lemma 10) to show that if there is an $m \times m$ matrix M whose entries are affine functions and $\mathsf{Det}(M) = \sum_{i=1}^n x_i^n$, then there is an $(m-n+2)\times(m-n+2)$ matrix N whose entries are polynomials of degree at most $n-1$ and $\mathsf{Det}(N) = (\sum_{i=1}^n x_i^n)(1+Q)$ for a polynomial Q which vanishes at zero. Moreover, if the matrix M is in normal form, then the matrix N continues to be in normal form.

Thus, we are in a setup where we can invoke the lower bound in Lemma 11 discussed earlier and we get that the dimension of N which equals $m-n+2$ must be at least $n/2-1$, thereby implying that m is at least $1.5n-3$. The details of this step can be found in Subsection 3.3.

2 Preliminaries

In this paper $\mathbb{F}$ always denotes an algebraically closed field. We use $\mathbf{x}$ to denote a tuple of n variables $x_1, \ldots, x_n$, where n is understood from the context (or is otherwise explicitly mentioned).

We consider polynomial maps $M : \mathbb{F}^n \to \mathbb{F}^{m\times m}$ given by m^2 polynomials $(M_{i,j})_{i,j\in[m]}$. The same object can be thought of as a matrix of polynomials $M(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{m\times m}$ and we use both points of view interchangeably. The degree of M is the maximum degree of its coordinates, i.e., $\deg M = \max_{i,j} \deg M_{i,j}$.

Each $M(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{m\times m}$ can be uniquely written as $M(\mathbf{x}) = M'(\mathbf{x}) + M_0$, where $M_0 \in \mathbb{F}^{m\times m}$ and in all m^2 coordinates of M', the constant term is zero. We then call M_0 the *constant part* of the map. A polynomial in which the constant term is zero is called *constant free*, and a polynomial map is called constant free if all of its coordinates are constant free, i.e., in the above decomposition, $M_0 = 0$.

We denote the determinant polynomial by Det. In cases where it is important to emphasize the dimension of the matrices in question we write it in the subscript, so for example the $m \times m$ determinant polynomial is denoted by Det_m.

We assume knowledge of basic concepts in algebraic geometry such as affine varieties $V \subseteq \mathbb{C}^n$ and their dimension, which we denote $\dim(V)$. We encourage readers unfamiliar with those terms to consult the excellent textbook [9].

Determinantal Complexity

We now formally define the notion of determinantal complexity, which is the focus of this paper.

▶ **Definition 2** (Determinantal Complexity). *The determinantal complexity of a polynomial $P \in \mathbb{F}[\mathbf{x}]$ is defined as the minimum $m \in \mathbb{N}$ such that there is a $m \times m$ matrix $M \in \mathbb{F}[\mathbf{x}]$ whose entries are polynomials of degree at most one such that*

$$P = \mathsf{Det}(M)\,.$$

▶ Remark 3. The above definition naturally generalizes to a family of polynomials in the following sense. A family $\{P_n\}_{n\in\mathbb{N}}$ of polynomials is said to have determinantal complexity at most $f(n) : \mathbb{N} \to \mathbb{N}$ if there exists an $n_0 \in \mathbb{N}$, such that for every $n \geq n_0$, the determinantal complexity of P_n is at most $f(n)$.

3 A lower bound on determinantal complexity

This section will be devoted for a proof of Theorem 1. We begin with the following lemma, which was instrumental in the recent proofs of lower bounds for algebraic formulas and algebraic branching programs.

▶ **Lemma 4** ([7, 13]). *Let $d \geq 2$ be a natural number. Let $P_1, P_2, \ldots, P_t, Q_1, \ldots, Q_t, L \in \mathbb{C}[\mathbf{x}]$ be polynomials such that $\deg(P') < d$, $P_1, \ldots, P_t, Q_1, \ldots, Q_t$ have a common zero and*

$$\sum_{i=1}^{n} x_i^d = \sum_{j=1}^{t} P_j(\mathbf{x})Q_j(\mathbf{x}) + P'\,.$$

Then, $t \geq n/2$.

We now show that without loss of generality, the constant part of every polynomial map M such that $\sum_{i=1}^{n} x_i^d = \mathsf{Det}_m(M(\mathbf{x}))$ has a very special form: is it an $m \times m$ diagonal matrix with 0 in the $(1, 1)$ coordinate and 1 in all diagonal entries.

3.1 Reducing the matrix M to a normal form

This claim is not entirely new and very similar statements were proved, for example, in [18, 2]. For completeness, and since the exact statement we need is slightly more general, we provide a proof.

▶ **Lemma 5.** *Let $d \geq 2$ be a natural number and let $M(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{m\times m}$ be a polynomial map such that*

$$\mathsf{Det}_m(M(\mathbf{x})) = \sum_{i=1}^{n} x_i^d\,.$$

Then, there exists a matrix $\tilde{M}(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{m\times m}$ with $\deg(\tilde{M}) \leq \deg(M)$,

$$\mathsf{Det}_m(\tilde{M}(\mathbf{x})) = \sum_{i=1}^{n} x_i^d\,,$$

and the constant part of $\tilde{M}$ is a diagonal $m \times m$ matrix $\tilde{M}_0$ such that $(\tilde{M}_0)_{1,1} = 0$ and $(\tilde{M}_0)_{i,i} = 1$, for $2 \leq i \leq m$.

To prove Lemma 5 we require a few preliminaries. We begin with the definition of a singular locus of a polynomial (or a hypersurface).

▶ **Definition 6.** *Let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial. The* singular locus *of f, denoted* $\operatorname{Sing}(f)$, *is the variety defined by*

$$\operatorname{Sing}(f) = \left\{ \mathbf{a} : \frac{\partial f}{\partial x_i}(\mathbf{a}) = 0, 1 \leq i \leq n \right\}.$$

The singular locus of the determinant was studied by von zur Gathen, who proved the following lemma.

▶ **Lemma 7** ([25]). *Let $\mathbb{F}$ be an algebraically closed field and let Det_m denote the $m \times m$ determinant polynomial. Then $\operatorname{Sing}(\mathsf{Det}_m) \subseteq \mathbb{F}^{m \times m}$ is precisely the set of matrices of rank at most $m-2$, and $\dim \operatorname{Sing}(\mathsf{Det}_m) = m^2 - 4$.*

The following is a slight generalization of a lemma of von zur Gathen (cf. also [2]).

▶ **Lemma 8.** *Let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial, and let $M : \mathbb{F}^n \to \mathbb{F}^{m \times m}$ be a polynomial map such that $f(\mathbf{x}) = \mathsf{Det}_m(M(\mathbf{x}))$. Suppose further that $\dim(\operatorname{Sing}(f)) < n - 4$. Then $\operatorname{Im}(M) \cap \operatorname{Sing}(\mathsf{Det}_m) = \emptyset$. Furthermore, all matrices in $\operatorname{Im}(M)$ have rank at least $m - 1$.*

Proof. Let $y_{i,j}$ denote the coordinates of $\mathbb{F}^{m \times m}$ and write $M = (M_{i,j})_{i,j \in [m]}$. Using the chain rule, we compute

$$\frac{\partial f}{\partial x_k} = \sum_{i,j \in [m]} \frac{\partial \mathsf{Det}_m}{\partial y_{i,j}}(M(\mathbf{x})) \cdot \frac{\partial M_{i,j}}{\partial x_k}(\mathbf{x}), \quad k \in [n]. \tag{1}$$

Suppose $A \in \operatorname{Im}(M) \cap \operatorname{Sing}(\mathsf{Det}_m)$, and let B be such that $A = M(B)$. By definition of $\operatorname{Sing}(\mathsf{Det}_m)$, $\frac{\partial \mathsf{Det}_m}{\partial y_{i,j}}(M(B)) = 0$ for all $i, j \in [m]$, and by (1) we get that $B \in \operatorname{Sing}(f)$. Thus $M^{-1}(\operatorname{Sing}(\mathsf{Det}_m)) \subseteq \operatorname{Sing}(f)$, and $\dim(M^{-1}(\operatorname{Sing}(\mathsf{Det}_m))) \leq \dim \operatorname{Sing}(f) < n - 4$. On the other hand, using a standard lower bound on the dimension of pre-images of polynomial maps (see Theorem 17.24 of [10]), if $\operatorname{Im}(M)$ and $\operatorname{Sing}(\mathsf{Det}_m)$ aren't disjoint,

$$\dim(M^{-1}(\operatorname{Sing}(\mathsf{Det}_m))) \geq n + (m^2 - 4) - m^2 = n - 4.$$

This contradiction implies that $\operatorname{Im}(M) \cap \operatorname{Sing}(\mathsf{Det}_m) = \emptyset$. The "furthermore" part of the theorem follows from Lemma 7. ◀

We will also need the following easy fact which shows that $\sum_{i=1}^{n} x_i^d$ satisfies that assumption of Lemma 8.

▶ **Lemma 9** ([13, 7]). *For every $d \geq 2$, $\dim(\operatorname{Sing}(\sum_{i=1}^{n} x_i^d)) = 0$.*

We are now ready to prove Lemma 5.

Proof of Lemma 5. Let $f = \sum_{i=1}^{n} x_i^d$ and let $M : \mathbb{F}^n \to \mathbb{F}^{m \times m}$ be a polynomial map such that $f(\mathbf{x}) = \mathsf{Det}_m(M(\mathbf{x}))$, and write $M = M' + M_0$ where M_0 is the constant part of M.

First, observe that

$$0 = f(\mathbf{0}) = \mathsf{Det}_m(M(\mathbf{0})) = \mathsf{Det}_m(M_0),$$

which implies that $\mathsf{rank}(M_0) < m$. By Lemma 8 and Lemma 9, we also know that $\mathsf{rank}(M_0) = \mathsf{rank}(M(\mathbf{0})) \geq m - 1$, so $\mathsf{rank}(M_0) = m - 1$.

By performing Gaussian elimination on the rows and on the columns, we can find two $m \times m$ matrices G_1, G_2 such that $\det(G_i) = \pm 1$ for $i = 1, 2$ and $N_0 := G_1 M_0 G_2$ is a diagonal matrix such that $(N_0)_{1,1} = 0$ and $(N_0)_{i,i} \neq 0$ for $2 \leq i \leq m$.

Now define a diagonal $m \times m$ matrix Δ such that $\Delta_{i,i} = 1/(N_0)_{i,i}$ for $2 \leq i \leq m$, and

$$\Delta_{1,1} = \mathsf{Det}(G_1) \cdot \mathsf{Det}(G_2) \cdot \prod_{i=2}^{m} (N_0)_{i,i}.$$

It readily follows that $\mathsf{Det}(\Delta) = \mathsf{Det}(G_1) \cdot \mathsf{Det}(G_2)$, and that $\tilde{M_0} := (G_1 M_0 G_2)\Delta$ is a diagonal matrix such that $(\tilde{M_0})_{1,1} = 0$ and $(\tilde{M_0})_{i,i} = 1$ for all $2 \leq i \leq m$.

Finally, define $\tilde{M} = G_1 M G_2 \Delta$. We verify that indeed

$$\begin{aligned}\mathsf{Det}(\tilde{M}(\mathbf{x})) &= \mathsf{Det}(G_1) \cdot \mathsf{Det}(M(\mathbf{x})) \cdot \mathsf{Det}(G_2) \cdot \mathsf{Det}(\Delta) \\ &= \mathsf{Det}(M(\mathbf{x})) \cdot (\mathsf{Det}(G_1) \cdot \mathsf{Det}(G_2))^2 = \mathsf{Det}(M(\mathbf{x})) = f(\mathbf{x}).\end{aligned}$$

We also have that

$$\tilde{M} = G_1(M' + M_0)G_2\Delta = G_1 M' G_2 \Delta + G_1 M_0 G_2 \Delta = G_1 M' G_2 \Delta + \tilde{M_0}.$$

Since $G_1, G_2, \Delta \in \mathbb{F}^{m \times m}$, it also holds that $\tilde{M'} := G_1 M' G_2 \Delta$ is a matrix of constant-free polynomials, and that $\deg \tilde{M} \leq \deg M$. ◀

We will also use the following simple and well known property of the determinant of a block matrix.

▶ **Lemma 10.** *Let $M \in \mathbb{F}^{m \times m}$ be a matrix, and let $A \in \mathbb{F}^{t \times t}, B \in \mathbb{F}^{t \times m-t}, C \in \mathbb{F}^{m-t \times t}, D \in \mathbb{F}^{m-t \times m-t}$ be its submatrices as follows:*

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

If D is invertible, then

$$\mathsf{Det}(M) = \mathsf{Det}(A - BD^{-1}C) \cdot \mathsf{Det}(D)\,.$$

Proof. Follows directly from the decomposition

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A - BD^{-1}C & BD^{-1} \\ 0 & I_{m-t} \end{pmatrix} \cdot \begin{pmatrix} I_t & 0 \\ C & D \end{pmatrix}$$

and the multiplicativity of the determinant. ◀

3.2 Determinantal complexity of higher degree polynomial maps

In the following lemma we prove a lower bound of $n/2$ on the determinantal complexity in a more general model than the standard model. This is a generalization with respect to two properties. First, the entries of the matrix are no longer constrained to be polynomials of degree at most 1, and can have degree as high as $d - 1$, while computing the degree d polynomial $\left(\sum_{i=1}^{n} x_i^d\right)$. Moreover, the determinant of the matrix M does not even have to compute the candidate hard polynomial $\left(\sum_{i=1}^{n} x_i^d\right)$ exactly. It suffices if the determinant is equal to a polynomial of the form $\left(\sum_{i=1}^{n} x_i^d\right) \cdot (\beta + Q)$ where β is a non-zero field constant and Q is an arbitrary polynomial (of potentially very high degree!) which is constant free, i.e. $Q(\mathbf{0}) = 0$.

▶ **Lemma 11.** *Let $d \geq 2$ be a natural number and let $M(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{m \times m}$ such that $\deg(M) \leq d-1$, and the constant part of M is a diagonal matrix M_0 such that $(M_0)_{1,1} = 0$ and $(M_0)_{i,i} = 1$ for $2 \leq i \leq m$. Suppose that*

$$\mathsf{Det}(M) = \left(\sum_{i=1}^{n} x_i^d\right) \cdot (\beta + Q)\,,$$

where $\beta \in \mathbb{F}$ is non-zero and Q is a constant free polynomial. Then $m \geq n/2 - 1$.

Proof. Using the Laplace expansion of $\mathsf{Det}(M)$ along the first row, we get

$$\mathsf{Det}(M) = \sum_{j=1}^{m} (-1)^{(j+1)} M_{1,j} \cdot \mathsf{Det}(N_{1,j})\,,$$

where $N_{i,j}$ is the submatrix of M obtained by deleting the i-th row and the j-th column. For every $j \in [m]$, $j > 1$, we claim that $\mathsf{Det}(N_{1,j})$ is a constant free polynomial, i.e.

$$\mathsf{Det}(N_{1,j})(\mathbf{0}) = \mathsf{Det}\left(N_{1,j}(\mathbf{0})\right) = 0\,.$$

To see this, we observe that for every $j \in [m] \setminus \{1\}$, $N_{1,j}(\mathbf{0})$ is a $(m-1) \times (m-1)$ matrix, which has at most $m-2$ non-zero entries. This follows since M_0 has at most $m-1$ non-zero entries and in obtaining $N_{1,j}$ from M, we drop the entry $M_{j,j}$, which is one of the $(m-1)$ entries of M with a non-zero constant term, and hence one of the $(m-1)$ non-zero entries of M_0. However, we note that $N_{1,1}(\mathbf{0})$ is the $(m-1) \times (m-1)$ identity matrix, so the constant term of $\mathsf{Det}(N_{1,1})$ is 1, and we write $\mathsf{Det}(N_{1,1}) = 1 + P(\mathbf{x})$ where P is constant free. Therefore, we have

$$\left(\sum_{i=1}^{n} x_i^d\right) \cdot (\beta + Q) = \mathsf{Det}(M) = M_{1,1}(1 + P) + \sum_{j=2}^{m} (-1)^{(j+1)} M_{1,j} \cdot \mathsf{Det}(N_{1,j})$$

In other words,

$$\left(\sum_{i=1}^{n} x_i^d\right) \cdot (\beta + Q) = \mathsf{Det}(M) = M_{1,1} + M_{1,1} \cdot P + \sum_{j=2}^{m} (-1)^{(j+1)} M_{1,j} \cdot \mathsf{Det}(N_{1,j})$$

Slightly rearranging (and using $\beta \neq 0$), we get

$$\sum_{i=1}^{n} x_i^d = \frac{1}{\beta}\left(-\left(\sum_{i=1}^{n} x_i^d\right) \cdot Q + M_{1,1} + M_{1,1} \cdot P + \sum_{j=2}^{m} (-1)^{(j+1)} M_{1,j} \cdot \mathsf{Det}(N_{1,j})\right)$$

Since, $\deg(M_{1,1}) < d$ and $M_{1,1}, P, M_{1,2}, \mathsf{Det}(N_{1,2}), \ldots, M_{1,k}, \mathsf{Det}(N_{1,k}), Q$ are all constant free (and hence share a common zero, namely $\mathbf{0}$), we have from Lemma 4 that $m \geq n/2 - 1$. ◀

3.3 Completing the proof of Theorem 1

We are now ready to complete the proof of Theorem 1.

Proof of Theorem 1. Let M be an $m \times m$ matrix with $\deg(M) \leq 1$ such that

$$\sum_{i=1}^{n} x_i^n = \mathsf{Det}(M)\,.$$

From Lemma 5, we can assume without loss of generality that the constant part M_0 of M is a diagonal matrix such that $(M_0)_{1,1} = 0$ and $(M_0)_{i,i} = 1$ for $2 \leq i \leq m$. In particular, all the off diagonal entries of M and $M_{1,1}$ are homogeneous linear forms or zero, and $M_{j,j} \neq 0$ for $j > 1$.

Observe that for every $t \leq m-1$, the principal minor D_t of M which is obtained by deleting the first $m-t$ rows and columns of M is invertible over the field of rational functions $\mathbb{F}(\mathbf{x})$. To see this, observe that the matrix $D_t(\mathbf{0})$ is the identity matrix, which implies that $\mathsf{Det}(D_t)$ is a non-zero polynomial. Moreover, since every entry of M has degree at most 1, and $\mathsf{Det}(M)$ has degree n, we know that $m \geq n$. So, we conclude that the principal minor $D := D_{(n-2)}$ of M is invertible over $\mathbb{F}(\mathbf{x})$. Thus, if B and C are respectively the submatrices of M defined as

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

then by Lemma 10 we have

$$\mathsf{Det}(M) = \mathsf{Det}(A - BD^{-1}C) \cdot \mathsf{Det}(D)\,. \tag{2}$$

Since $D^{-1} = \mathrm{adj}(D)/\det(D)$, where $\mathrm{adj}(D)$ is the adjugate matrix of D, the entries of D^{-1} can be written as as a ratio of two polynomials, where the numerator has degree at most $n-3$ and the denominator, which is equal to $\mathsf{Det}(D)$, has degree at most $n-2$. Moreover, as discussed earlier in the proof, the constant part of D is the identity matrix, so there is a constant free polynomial $R \in \mathbb{F}[\mathbf{x}]$ such that

$$\mathsf{Det}(D) = 1 + R\,.$$

Thus, every entry of the $(m-n+2) \times (m-n+2)$ matrix $A - BD^{-1}C$ can be written as a ratio of two polynomials with the numerator being a polynomial of degree at most $n-1$ and the denominator being equal to $\mathsf{Det}(D) = 1 + R$. Therefore, by clearing the denominators and using (2), we get that

$$\mathsf{Det}(M) \cdot (1+R)^{m-n+2} = \mathsf{Det}(N) \cdot (1+R)\,,$$

where N is the matrix with polynomial entries of degree at most $n-1$ obtained by multiplying every entry of $A - BD^{-1}C$ by $1+R$. Simplifying further, we get

$$\left(\sum_{i=1}^{n} x_i^n\right) \cdot (1+R)^{m-n+1} = \mathsf{Det}(M) \cdot (1+R)^{m-n+1} = \mathsf{Det}(N)\,.$$

We are almost ready to invoke Lemma 11 to obtain a lower bound on the size of N (and hence M), but to do that we need to ensure that the constant part of N, N_0, is a diagonal matrix with $(N_0)_{1,1} = 0$ and $(N_0)_{i,i} = 1$ for $2 \leq i \leq m-n+2$. We now verify that this is indeed the case.

Recall that by the structure of the constant part M_0 of M, all the entries of B and C and the $(1,1)$ entry of A are constant free, and the constant term of $A_{i,i}$ is 1 for $2 \leq i \leq m-n+2$. Thus, every entry of the matrix $BD^{-1}C$ is a rational function with a constant free numerator, and hence all the off-diagonal entries in $A - BD^{-1}C$ as well as its $(1,1)$ entry are rational functions with a constant free numerator. Moreover, the denominator of all the entries $\left(A - BD^{-1}C\right)$ equals $\mathsf{Det}(D) = 1 + R$, for a constant free polynomial R. So, expressing each entry of $A - BD^{-1}C$ as a quotient of polynomials, the constant term of each numerator on the diagonal is 1 except for the $(1,1)$ entry, which has a constant free numerator. Finally, observe that eliminating the denominator of the entries of $\left(A - BD^{-1}C\right)$ by multiplying every entry by $(1+R)$ gives us the matrix N.

Thus the matrix N satisfies the hypothesis of Lemma 11, and hence $(m-n+2) \geq n/2-1$. This gives us $m \geq 1.5n-3$ and completes the proof of Theorem 1. ◀

References

1 Boris Alexeev, Michael A. Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity (CCC 2011)*, pages 283–291. IEEE Computer Society, 2011. doi:10.1109/CCC.2011.28.

2 Jarod Alper, Tristram Bogart, and Mauricio Velasco. A lower bound for the determinantal complexity of a hypersurface. *Found. Comput. Math.*, 17(3):829–836, 2017. doi:10.1007/s10208-015-9300-x.

3 Markus Bläser. A $\frac{5}{2}n^2$-lower bound for the rank of $n \times n$ matrix multiplication over arbitrary fields. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1999)*, pages 45–50, 1999. doi:10.1109/SFFCS.1999.814576.

4 Mark R. Brown and David P. Dobkin. An improved lower bound on polynomial multiplication. *IEEE Trans. Computers*, 29(5):337–340, 1980. doi:10.1109/TC.1980.1675583.

5 Jin-Yi Cai. A note on the determinant and permanent problem. *Inf. Comput.*, 84(1):119–127, 1990. doi:10.1016/0890-5401(90)90036-H.

6 Jin-Yi Cai, Xi Chen, and Dong Li. Quadratic lower bound for permanent vs. determinant in any characteristic. *Comput. Complex.*, 19(1):37–56, 2010. doi:10.1007/s00037-009-0284-2.

7 Prerona Chatterjee, Mrinal Kumar, Adrian She, and Ben Lee Volk. A quadratic lower bound for algebraic branching programs. *CoRR*, abs/1911.11793, 2019. arXiv:1911.11793.

8 Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity. *Foundations and Trends in Theoretical Computer Science*, 2011. doi:10.1561/0400000043.

9 David A. Cox, John B. Little, and Donal O'Shea. *Ideals, Varieties and Algorithms.* Undergraduate texts in mathematics. Springer, 2007. doi:10.1007/978-0-387-35651-8.

10 Joe Harris. *Algebraic geometry*, volume 133 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1995. A first course, Corrected reprint of the 1992 original. doi:10.1007/978-1-4757-2189-8.

11 Christian Ikenmeyer and J.M. Landsberg. On the complexity of the permanent in various computational models. *Journal of Pure and Applied Algebra*, 221(12):2911–2927, 2017. doi:10.1016/j.jpaa.2017.02.008.

12 Kyriakos Kalorkoti. A Lower Bound for the Formula Size of Rational Functions. *SICOMP*, 14(3):678–687, 1985. doi:10.1137/0214050.

13 Mrinal Kumar. A quadratic lower bound for homogeneous algebraic branching programs. *Computational Complexity*, 28(3):409–435, 2019. doi:10.1007/s00037-019-00186-3.

14 J.M. Landsberg and Nicolas Ressayre. Permanent v. determinant: An exponential lower bound assuming symmetry and a potential path towards valiant's conjecture. *Differential Geometry and its Applications*, 55:146–166, 2017. doi:10.1016/j.difgeo.2017.03.017.

15 Joseph M. Landsberg, Laurent Manivel, and Nicolas Ressayre. Hypersurfaces with degenerate duals and the geometric complexity theory program. *Comment. Math. Helv.*, 88(2):469–484, 2013. doi:10.4171/CMH/292.

16 Marvin Marcus and Henryk Minc. On the relation between the determinant and the permanent. *Illinois J. Math.*, 5(3):376–381, September 1961. doi:10.1215/ijm/1255630882.

17 Roy Meshulam. On two extremal matrix problems. *Linear Algebra and its Applications*, 114–115:261–271, 1989. URL: https://www.sciencedirect.com/science/article/pii/0024379589904655.

18 Thierry Mignon and Nicolas Ressayre. A quadratic bound for the determinant and permanent problem. *International Mathematics Research Notes*, 2004(79):4241–4253, 2004. Available on `citeseer:10.1.1.106.4910`. `doi:10.1155/S1073792804142566`.

19 George Pólya. Aufgabe 424. *Archiv der Mathematik und Physik*, 20:271, 1913. URL: `http://babel.hathitrust.org/cgi/pt?id=mdp.39015085215716;seq=399`.

20 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015. URL: `https://github.com/dasarpmar/lowerbounds-survey/releases/`.

21 Amir Shpilka. Lower bounds for matrix product. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2001)*, pages 358–367, 2001. `doi:10.1109/SFCS.2001.959910`.

22 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010. `doi:10.1561/0400000039`.

23 Gábor Szegő. Lösung zu aufgabe 424. *Archiv der Mathematik und Physik*, 21:291–292, 1913. URL: `http://hdl.handle.net/2027/uc1.b2958231`.

24 Leslie G. Valiant. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 1979)*, pages 249–261, 1979. `doi:10.1145/800135.804419`.

25 Joachim von zur Gathen. Permanent and determinant. *Linear Algebra and its Applications*, 96:87–100, 1987. URL: `https://core.ac.uk/download/pdf/82095887.pdf`.

26 Akihiro Yabe. Bi-polynomial rank and determinantal complexity. *CoRR*, abs/1504.00151, 2015. `arXiv:1504.00151`.

Optimal Tiling of the Euclidean Space Using Permutation-Symmetric Bodies

Mark Braverman
Department of Computer Science, Princeton University, NJ, USA

Dor Minzer
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

What is the least surface area of a permutation-symmetric body B whose $\mathbb{Z}^n$ translations tile $\mathbb{R}^n$? Since any such body must have volume 1, the isoperimetric inequality implies that its surface area must be at least $\Omega(\sqrt{n})$. Remarkably, Kindler et al. showed that for general bodies B this is tight, i.e. that there is a tiling body of $\mathbb{R}^n$ whose surface area is $O(\sqrt{n})$.

In theoretical computer science, the tiling problem is intimately related to the study of parallel repetition theorems (which are an important component in PCPs), and more specifically in the question of whether a "strong version" of the parallel repetition theorem holds. Raz showed, using the odd cycle game, that strong parallel repetition fails in general, and subsequently these ideas were used in order to construct non-trivial tilings of $\mathbb{R}^n$.

In this paper, motivated by the study of a symmetric parallel repetition, we consider the permutation-symmetric variant of the tiling problem in $\mathbb{R}^n$. We show that any permutation-symmetric body that tiles $\mathbb{R}^n$ must have surface area at least $\Omega(n/\sqrt{\log n})$, and that this bound is tight, i.e. that there is a permutation-symmetric tiling body of $\mathbb{R}^n$ with surface area $O(n/\sqrt{\log n})$. We also give matching bounds for the value of the symmetric parallel repetition of Raz's odd cycle game.

Our result suggests that while strong parallel repetition fails in general, there may be important special cases where it still applies.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases PCP, Parallel Repetition, Tilings

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.5

Related Version *Full Version*: `https://arxiv.org/abs/2011.04071`

Funding *Mark Braverman*: Research supported in part by the NSF Alan T. Waterman Award, Grant No. 1933331, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.

1 Introduction

A body $D \subseteq \mathbb{R}^n$ is said to be tiling the Euclidean space $\mathbb{R}^n$, if its translations by $\mathbb{Z}^n$ cover the entire space and have disjoint interiors. The foam problem asks for the least surface area a tiling body D can have. The problem had been considered by mathematicians already in the 19th century [33], and it also appears in chemistry, physics and engineering [30]. More recently, the problem had received significant attention in the theoretical computer science community due to its strong relation with the parallel repetition problem [15, 24, 2].

The simplest example for a body that tiles the Euclidean space is the solid cube, $D = [0,1]^n$, which has surface area $2n$. At first glance, one may expect the solid cube to be the best example there is, or more modestly that any tiling body would need to have surface area $\Omega(n)$. The main results of [24, 2] show that this initial intuition is completely false, and that there are far more efficient tiling bodies whose surface area is $O(\sqrt{n})$. This is surprising, since spheres – which are the minimizers of surface area among all bodies with a given, fixed volume (in this case volume 1), have $\Theta(\sqrt{n})$ surface area and seem to be very far from forming a tiling of $\mathbb{R}^n$. As we will shortly discuss, the existence of such surprising tiling body is intimately related to the existence of another surprising object – namely non-trivial strategies for 2-prover-1-round games, repeated in parallel. The main goal of this paper is to understand the permutation-symmetric variant of the foam problem, which is closely related to the symmetric variant of parallel repetition.

1.1 2-Prover-1-Round Games and Parallel Repetition

▶ **Definition 1.** *A 2-Prover-1-Round Game $G = (L \cup R, E, \Phi, \Sigma_L, \Sigma_R)$ consists of a bipartite graph $(L \cup R, E)$, alphabets Σ_L, Σ_R, and a constraint $\Phi(u, v)$ for every edge $(u, v) \in E$. The goal is to find assignments $A_L : L \to \Sigma_L$, $A_R : R \to \Sigma_R$ that satify the maximum fraction of the constraints. A constraint $\Phi(u, v)$ is satisfied if $(A_L(u), A_R(v)) \in \Phi(u, v)$, where by abuse of notation, $\Phi(u, v) \subseteq \Sigma_L \times \Sigma_R$ denotes the subset of label pairs that are deemed satisfactory.*

The value of a game, denoted by $\mathsf{val}(G)$, is the maximum fraction of constraints that can be satisfied in G by any pair of assignments A_L, A_R.

Equivalently, a 2-Prover-1-Round Game can be viewed as a “game” between two provers and a verifier. The verifier picks a constraint (u, v) at random, asks the “question” u to the left prover, the “question” v to the right prover, receives “answers” $A_L(u), A_R(v)$ respectively from the provers; the verifier accepts if and only if $(A_L(u), A_R(v)) \in \Phi(u, v)$. It is easy to see that in this language, $\mathsf{val}(G)$ represents the maximum probability a verifier will accept, where the maximum is taken over all of the strategies of the provers.

2-Prover-1-Round games play an important role in the study of PCPs and Hardness of approximation, and in fact an equivalent statement of the seminal PCP Theorem [14, 5, 4] can be stated in that language. It will be convenient for us to use the notation of gap problems: for $0 < s < c \leqslant 1$, denote by $\mathsf{Gap2Prover1Round}(c, s)$ the promise problem in which the input is a 2-Prover-1-Round game G promised to either satisfy $\mathsf{val}(G) \geqslant c$ or $\mathsf{val}(G) \leqslant s$, and the goal is to distinguish between these two cases. The parameters c and s are referred to as the completeness and soundness parameters of the problem, respectively.

▶ **Theorem 2** (PCP Theorem, [14, 5, 4])**.** *There are $k \in \mathbb{N}$, $s < 1$ for which the problem $\mathsf{Gap2Prover1Round}(1, s)$ is NP-hard on instances with alphabet size at most k.*

The PCP Theorem, as stated above, can be used to establish some hardness of approximation results. However it turns out that to get strong hardness results, one must prove a variant of the theorem with small soundness, i.e. with s close to 0. One way to do that is by amplifying hardness using *parallel repetition*.

The t-fold repetition of a game G, denoted by $G^{\otimes t}$, is the game in which the verifier picks t independently chosen challenges, $(u_1, v_1), \dots, (u_t, v_t)$ and sends them to the provers in a single bunch, i.e. $\vec{u} = (u_1, \dots, u_t)$ to one prover and $\vec{v} = (v_1, \dots, v_t)$ to the second one. The provers are supposed to give an answer to each one of their questions, say $A_L(\vec{u}) = (a_1, \dots, a_t)$ and $A_R(\vec{v}) = (b_1, \dots, b_t)$, and the verifier accepts with only if $(a_i, b_i) \in \Phi(u_i, v_i)$ for all $i = 1, \dots, t$. What is the value of the t-fold repeated game, as a function of $\mathsf{val}(G)$ and t?

The idea of parallel repetition was first introduced in [16], wherein it was originally suggested that $\mathsf{val}(G^{\otimes t}) \approx \mathsf{val}(G)^t$. Alas, in a later version of that paper it was shown to be false, leaving the question wide open. Raz [27] was the first to prove that the value of the repeated game decreases exponentially with t, and with many subsequent works improving the result [18, 26, 13, 10]. The most relevant version for our purposes is the result of Rao [26], which makes the following statement. First, we say a game G is a projection game, if all of the constraints $\Phi(u, v)$ can be described by a projection map, i.e. there is a mapping $\pi_{u,v} \colon \Sigma_L \to \Sigma_R$ such that $\Phi(u, v) = \{\, (a, b) \mid b = \phi_{u,v}(a) \}$.

▶ **Theorem 3.** *If G is a projection game, and* $\mathsf{val}(G) = 1 - \varepsilon$, *then* $\mathsf{val}(G^{\oplus t}) \leqslant (1 - \varepsilon^2)^{\Omega(t)}$.

Rao's result seems nearly optimal, in the sense that a-priori, the best bound one can hope for is that $\mathsf{val}(G^{\oplus t}) \leqslant (1 - \varepsilon)^{\Omega(t)}$. Quantitatively speaking, one may think that for all intents and purposes, Rao's bound is just as good as the best one can hope for. However, as it turns out, there is at least one prominent problem where this quadratic gap is what makes the difference, which we describe next.

The Unique Games Conjecture and the Max-Cut Conjecture

The Unique Games problem is a specific type of projection 2-Prover-1-Round Game, in which the projection maps $\phi_{u,v}$ are also bijections. The Unique Games Conjecture of Khot [19] (abbreviated UGC henceforth) asserts that a strong PCP theorem holds for Unique-Games, and more specifically that for any $\varepsilon, \delta > 0$, the problem $\mathsf{GapUG}(1-\varepsilon, \delta)$ is NP-hard, when the alphabet sizes depend only on ε, δ. This conjecture is now of central importance in complexity theory, and it is known to imply many, often tight inapproximability results (see [20, 34] for more details). A prominent example is the result of [21], stating that assuming UGC, the Goemans-Williamson algorithm [17] for Max-Cut is optimal. In particular, for small enough $\varepsilon > 0$, if UGC is true, then $\mathsf{GapMaxCut}(1 - \varepsilon, 1 - \frac{2}{\pi}\sqrt{\varepsilon} + O(\varepsilon^{1.5}))$ is NP-hard. Does the converse hold? I.e., does the assumption that $\mathsf{GapMaxCut}(1 - \varepsilon, 1 - \frac{2}{\pi}\sqrt{\varepsilon} + O(\varepsilon^{1.5}))$ is NP-hard imply UGC? If so, that would be a promising avenue of attack on the Unique-Games Conjecture.

Noting that Max-Cut is a Unique-Game and that Parallel repetition preserves uniqueness, one may hope a reduction from $\mathsf{GapMaxCut}(1-\varepsilon, 1-\frac{2}{\pi}\sqrt{\varepsilon}+O(\varepsilon^{1.5}))$ to $\mathsf{GapUniqueGames}(1-\varepsilon', \delta)$ would simply follow by appealing to a parallel repetition theorem, such as Rao's result [26]. Alas, the quadratic loss there exactly matches the quadratic gap we have in Max-Cut, thereby nullifying it completely. This possibility was discussed in [31], who among other things proposed that perhaps a stronger version of Theorem 3 should hold for Unique-Games, in which the ε^2 is replaced with ε. This conjecture was referred to as the Strong Parallel Repetition Conjecture, and unfortunately it turns out to be false.

A Strong parallel repetition theorem?

The problem of understanding parallel repetition over a very simple game, called the odd cycle game and denoted below by C_n, was shown to be closely related to the foam problem [15]. In this game, we have a graph G which is an odd cycle of length n, and the provers try to convince the verifier that G is a bipartite graph (while it is clearly not). To test the provers, the verifier picks a vertex u from the cycle uniformly at random, and then picks v as $v = u$ with probability $1/2$, and otherwise v is one of the neighbours of u with equal probability. The verifier sends u as a question to one prover, and v as a question to the other prover, and expects to receive a bit from each one b_1, b_2. The verifier checks that $b_1 = b_2$ in case $u = v$, or that $b_1 \neq b_2$ in case $u \neq v$.

Note that clearly, $\mathsf{val}(C_n) = 1 - \Theta(1/n)$, and so the Strong Parallel Repetition Conjecture would predict that the value of the t-fold repeated game is $1 - \Theta(t/n)$ so long as $t \leqslant n$. Alas, this turns out to be false. First, in [15], it was shown that non-trivial solutions to the foam problem imply non-trivial strategies for the t-fold repeated game, and in particular the existence of a tiling body with surface area $o(n)$ would refute the Strong Parallel Repetition Conjecture. Subsequently, Raz [28] showed that the value of the t-fold repeated odd-cycle game is in fact at least $1 - O(\sqrt{t}/n)$ so long as $t \leqslant n^2$, and that Theorem 3 is optimal (i.e., the quadratic gap is necessary, even for Unique-Games, and more specifically for Max-Cut).

Subsequent works were able to use these insights to solve the foam problem for the integer lattice [24, 2] and lead to better understanding of parallel repetition and its variants [6, 8]. From the point of view of UGC, these results were very discouraging since they eliminate one of the main available venues (perhaps the main one) for the proof of UGC.

Partly due to this issue, the best partial results towards UGC had to take an entirely different approach [22, 12, 11, 23, 7], and currently can only prove that $\mathsf{GapUG}(1/2, \delta)$ is NP-hard for every $\delta > 0$.

1.2 A symmetric variant of Parallel Repetition

One may try to revive the plan for showing the equivalence of UGC and the hardness of Max-Cut by considering variants of parallel repetition. Ideally, for that approach to work, one should come up with a variant of parallel repetition, in which (a) the value decreases exponentially with the number of repetitions, and (b) the operation preserves uniqueness. One operation that had been considered in the literature, for example, is called fortification [25, 9]. Using this operation, the value of the game indeed decreases exponentially, however this operation does not preserve uniqueness and therefore is not useful for showing the equivalence of UGC and the Max-Cut Conjecture.

More relevant to us is the symmetric variant of parallel repetition that had been previously suggested as a replacement for parallel repetition. In this variant, given a basic game G, the verifier chooses the challenges $(u_1, v_1), \ldots, (u_t, v_t)$, and sends the questions to the provers as unordered tuples, i.e. $U = \{u_1, \ldots, u_t\}$ and $V = \{v_1, \ldots, v_t\}$. The verifier expects to receive a label for each element in U and each element in V, and checks that they satisfy each one of the constraints (u_i, v_i). We denote this game by $G^{\otimes_{\mathsf{sym}} t}$, and note that it clearly preserves uniqueness; also, we note that the arguments used to refute the strong Parallel Repetition Conjecture do not immediately apply to it. While a naive application of this variant can still be shown to fail in general,[1] there is still a hope that it can be used in a more clever way and establish the equivalence of UGC and the Max-Cut Conjecture. Our work is partly motivated by seeking such possibilities.

We are thus led to investigate the effect on symmetric repetition on the odd cycle game, and more specifically the symmetric variant of the foam problem which again is very much related.

1.3 Our results

In this paper, our main object of study mainly are tilings of $\mathbb{R}^n$ using a *permutation-symmetric body.*

▶ **Definition 4.** *A set $D \subseteq \mathbb{R}^n$ is called permutation-symmetric if for any $\pi \in S_n$ and $x \in \mathbb{R}^n$, it holds that $x \in D$ if and only if $\pi(x) \in D$.*

The main question we consider, is what is the least surface area a permutation-symmetric tiling body can have. Again, one has the trivial example of the solid cube $D = [0, 1]^n$, but inspired by the non-permutation-symmetric variant of the problem, one may expect there to be better examples. We first show that while this is possible, the savings are much milder, and can be at most a multiplicative factor of $\sqrt{\log n}$.

[1] This can be seen by considering a graph which is the disjoint union of many odd cycles (instead of a single odd cycle), say M, so that one would get a canonical ordering on most subsets of t vertices from this graph, so long as $t = o(\sqrt{M})$.

▶ **Theorem 5.** *Any permutation-symmetric tiling body D of volume 1 with piecewise smooth surface has surface area at least* $\Omega\left(\frac{n}{\sqrt{\log n}}\right)$.

Besides the quantitative result itself, we believe the argument used in the proof of Theorem 5 carries with it a lot of intuition regarding the additional challenge that the permutation-symmetric variant of the foam problem and the parallel repetition posses, and we hope that this intuition will help us to develop better understanding of symmetric parallel repetition in general. We remark that our proof actually shows a lower bound on the "noise sensitivity" parameter of the body, which is known to be smaller than the surface area of the body.

We complement Theorem 5 with a randomized construction showing that $O(\sqrt{\log n})$ savings are indeed possible.

▶ **Theorem 6.** *There exists a permutation-symmetric tiling body D of volume 1 with piecewise smooth surface that has surface area* $O\left(\frac{n}{\sqrt{\log n}}\right)$.

Our results also imply tight bounds for the value of the t-fold symmetric repetition of the odd cycle game, which we discuss next.

1.4 Significance of our results for symmetric parallel repetition

Using our techniques, one may give sharp estimates to the value of the t-fold symmetric repetition of the odd cycle game, as follows.

▶ **Theorem 7.** *There is $c > 0$, such that for an odd n, if $t \leqslant cn\sqrt{\log n}$ then* $\mathsf{val}(C_n^{\otimes_{\mathsf{sym}} t}) \leqslant 1 - c\frac{t}{n\sqrt{\log t}}$.

▶ **Theorem 8.** *For all $n, t \in \mathbb{N}$ it holds that* $\mathsf{val}(C_n^{\otimes_{\mathsf{sym}} t}) \geqslant 1 - O\left(\frac{t}{n\sqrt{\log t}}\right)$.

We remark that a similar connection between the standard foam problem and the value of the t-fold repeated game is well known. More precisely, in [15] the authors show that (1) tilings of the Euclidean space with small surface area can be used to derive good strategies for $C_n^{\otimes t}$, and (2) the Euclidean isoperimetric inequality (which gives a lower bound of $\Theta(\sqrt{n})$ on the surface area of a tiling body) can be used to prove upper bounds on the value of $C_n^{\otimes t}$. We remark that while (1) above is derived in a black-box way, the converse direction, i.e. (2), is done in a white-box way. That is, the authors in [15] do not actually use the Euclidean isoperimetric inequality, but rather convert one of its proofs into an upper bound of the value of the t-fold repeated odd cycle game.

In contrast to [15], our proof of Theorems 7, 8 follow more direct adaptations of the proofs of Theorems 5, 6. This is partly because our arguments work from scratch and are therefore more flexible. We outline these adaptations in Section 5.

We believe that Theorem 7 gives some new life to the possible equivalence between the Max-Cut Conjecture and UGC. For example, this would follow if such rate of amplification would hold for all graphs if we allow for a "mild" preprocessing phase first (i.e., preprocessing that doesn't change the value of the instance by much). For this reason, we believe it would be interesting to investigate other graph topologies on which symmetric parallel repetition performs well, and hope that the techniques developed herein will be useful.

On the flip side, Theorem 8 asserts that even symmetric parallel repetition on the odd cycle game admits non-trivial strategies. Thus, we cannot hope to use it in order to establish the equivalence of weaker forms of the Max-Cut Conjecture and UGC. Here, by weaker forms of the Max-Cut Conjecture, we mean the conjecture that $\mathsf{GapMaxCut}[1-\varepsilon, 1-\delta(\varepsilon)]$ is NP-hard for small enough ε, and $\delta(\varepsilon)$ is a nearly linear function of ε, e.g. $\delta(\varepsilon) = 100\varepsilon$ or $\delta(\varepsilon) = \varepsilon\sqrt{\log(1/\varepsilon)}$. Given that the best known NP-hardness results for Max-Cut in this regime are only known for $\delta = (1+\Omega(1))\varepsilon$, this means that there is still a significant road ahead to establish even the weakest version of the Max-Cut Conjecture that may be useful for UGC.

1.5 Techniques

In this section, we explain some of the intuition and idea that go into the proof of Theorems 5 and 6, focusing mostly on the former.

Let D be a permutation-symmetric tiling body. To prove that the surface area of D is at least A, it is enough to prove that D is sensitive to noise rate $1/A$. I.e., that if we take a point x from D uniformly at random, and walk along a random Gaussian direction u of expected length $1/A$, then with constant probability we escape D at some point on the line $\ell_{x,u}(t) = x + tu$.

We begin by describing an argument showing a worse bound than the one proved in Theorem 5, which is nevertheless helpful in conveying some of the intuition. To prove that a random line $\ell_{x,u}(t)$ escapes D with noticeable probability, we argue that for a Gaussian vector u of appropriate expected length, with constant probability the line $\ell_{x,u}$ will contain a point in which there are two coordinates differing by a non-zero integer, say y with the coordinates being i, j. Note that this is enough, since then if we assumed that $y \in D$, then the point y' in which the value of coordinates i, j is switched also lies in D (by symmetry), and then the difference of y and y' is a non-zero lattice vector, so they must be in different cells of the tiling. Therefore we conclude that $y \notin D$.

With this plan in mind, let $x = (x_1, \ldots, x_n)$ be uniformly sampled from D, and consider the coordinates of x modulo 1, i.e. $B = \{x_1 \pmod 1, \ldots, x_n \pmod 1\}$, as points in the one-dimensional torus $\mathbb{T}$. First, it can be shown without much difficulty that they are jointly distributed as uniform random points on $\mathbb{T}$, hence standard probabilistic tools tell us that any interval of length $100\log n/n$ on the torus contains at least two points from B. Now, regardless of how the body D looks, there would be two coordinates, say i and j, that almost differ by a non-zero integer, yet appear very close when projected on the torus, i.e. in distance at most $100\log n/n$. In this case, with constant probability the coordinates i, j get even closer along a random line $\ell_{x,u}(t) = x + tu$, and provided the length of u is long enough to cover the distance between x_i, x_j on the torus (i.e. each coordinate of magnitude $\Theta(\log n/n)$), the line $\ell_{x,u}(t)$ would contain a point as desired.

The above argument can indeed be formalized to yield a lower bound of $\Omega\left(\frac{n}{\log n}\right)$ on the surface area of D, but it carries more intuition than just the bound itself. In a sense, this argument says that if we project x onto the torus, we should be wary of coordinates whose projections are too close, and make sure that it would only occur if the coordinates themselves are close (as opposed to almost differing by a non-zero integer). Analyzing the event that two coordinates meet on the circle while being different is easily seen however to not yield a better bound than $\Omega(n/\log n)$, hence to prove Theorem 5 we must look at a different event. That being said, the argument does tell us that we should look at pairwise distances between coordinates of x when projected on the circle, and in particular on pairs that are "relatively close" and the way they move along a line in a random direction.

It turns out that it is enough to come up with some parameter that behaves differently on the endpoints of the line, assuming the line does not escape D. This is because that if the escape probability from D is small, then the distributions of x and $x + u$ are close in statistical distance, and in particular any parameter should behave roughly the same on x and on $x+u$. Indeed, our proof utilizes an energy function (inspired by the previous argument) that considers the pairwise distances between coordinates of x; the contribution from a pair of coordinates that are in distance d in the circle is proportional to $e^{-Z\cdot d}$, where $Z \sim \frac{n}{\sqrt{\log n}}$. We show that with high probability, the energy increases along a random line $\ell_{x,u}(t)$ provided it does not escape D, while on the other hand, if the escape probability is small, then x and $x+u$ are close in statistical distance and hence $\Pr_{x,u}\left[\mathsf{Energy}(x+u) > \mathsf{Energy}(x)\right] \approx \frac{1}{2}$. This implies that the escape probability must be constant.

We remark that the above high-level intuition also plays a role in the proof of Theorem 5. I.e., when constructing a permutation-symmetric tiling body D, all we really need to care about are the pairwise distances between coordinates, and that we must make sure that somewhat far coordinates will project to far points on the torus. Indeed, given a point $x \in \mathbb{R}^n$, in order to decide which integer lattice point $y \in \mathbb{Z}^n$ we round x to, we only look at this pairwise distances of x on the torus. We try to find a point z on the torus that is far from all the coordinates of x, and do the rounding according to it. One naive attempt would be to take z that is furthest from all coordinates of x, however this point turns out to be very noise sensitive and therefore yield a body with large surface area. Instead, we consider a probability distribution that only puts significant weight on z's that are somewhat far from all x_i's, yet is not too concentrated around the maximizers. Coming up and analyzing a construction along these lines turns out to require considerable technical effort, and we defer a more elaborate discussion to Section 4

Organization of the paper

In Section 2, we set up basic notations and preliminaries. Section 3 is devoted to the proof of Theorem 3, and Section 4 is devoted for the proof of Theorem 4. In Section 5 we prove Theorems 7, 8, and in Section 6 we state some open problems.

2 Preliminaries

Notations

We write $X \lesssim Y$ or $X = O(Y)$ to say that there exists an absolute constant $C > 0$ such that $X \leqslant CY$, and similarly write $X \gtrsim Y$ or $X = \Omega(Y)$ to say that there exists an absolute constant $c > 0$ such that $X \geqslant cY$. We write $X \asymp Y$ or $X = \Theta(Y)$ to say that $Y \lesssim X \lesssim Y$.

We denote random variables by boldface letters such as $\mathbf{x}$ and $\mathbf{\Delta}$. We denote by $\mathcal{N}(\mu, \sigma^2)$ the distribution of a standard Gaussian random variable with mean μ and variance σ^2, and by $\mathcal{N}(\vec{\mu}, \Sigma)$ the distribution of a multi-dimensional Gaussian random variable with means $\vec{\mu}$ and covariance matrix Σ.

For a measurable set D of finite measure, we denote by $\mathbf{a} \in D$ or $\mathbf{a} \in_R D$ a uniform sample from D.

2.1 Needles

▶ **Definition 9.** *Let $\delta > 0$, and let $a \in \mathbb{R}^n$. A random δ-needle is a line defined as $\ell_{a,\mathbf{u}} = \{a + t\mathbf{u} \mid t \in [0,1]\}$ where the direction vector $\mathbf{u}$ is a chosen as a standard Gaussian $\mathcal{N}(0, \delta I_n)$.*

Given a tiling body D, a random δ-needle from D is a random δ-needle $\ell_{\mathbf{a},\mathbf{u}}$ where $\mathbf{a} \in D$ is chosen uniformly. Random needles are a useful tool to measure the surface area of a D, as shown in the following two lemmas. First, given a tiling body D and a needle $\ell_{a,u}$, we may think of the needle as “wrapping around” around D, i.e. its points are taken modulo D. We denote this “wrapped around” line by $\tilde{\ell}_{a,u}$. We will use the following formula from [32]; the case $n = 2$ is formula (8.10) therein, and the extension to general n is discussed in page 274.

▶ **Lemma 10.** *There is a constant $C_n = \Theta(1)$, such that the following holds. Let S be a piecewise smooth surface in a tiling body D of volume 1, and let $\delta > 0$. Then*

$$\mathop{\mathbb{E}}_{\mathbf{a}\in D,\mathbf{u}\sim\mathcal{N}(0,\delta I_n)}\left[\left|\tilde{\ell}_{\mathbf{a},\mathbf{u}} \cap S\right|\right] = C_n \cdot \sqrt{\delta} \cdot \mathsf{area}(S).$$

▶ **Lemma 11.** *Let D be a tiling body of volume 1, and let $\delta > 0$. Then*

$$\Pr_{\mathbf{a}\in D,\mathbf{u}\sim\mathcal{N}(0,\delta I_n)}[\ell_{\mathbf{a},\mathbf{u}} \cap \partial D \neq \emptyset] \leqslant \Theta(\sqrt{\delta})\mathsf{area}(\partial D).$$

Proof. Set $S = \partial D$, and note that whenever $\ell_{a,\delta u} \cap \partial D \neq \emptyset$, we have that $\left|\tilde{\ell}_{a,\delta u} \cap S\right| \geqslant 1$. Hence by the previous lemma we get that

$$\Pr_{\mathbf{a}\in D,\mathbf{u}\sim\mathcal{N}(0,\delta I_n)}[\ell_{\mathbf{a},\mathbf{u}} \cap \partial D \neq \emptyset] \leqslant \mathop{\mathbb{E}}_{\mathbf{a}\in D,\mathbf{u}\sim\mathcal{N}(0,\delta I_n)}\left[\left|\tilde{\ell}_{\mathbf{a},\mathbf{u}} \cap \partial D\right|\right] \leqslant \Theta(\sqrt{\delta}) \cdot \mathsf{area}(\partial D). \quad ◀$$

We will use the above lemma to prove lower bounds on the surface area of a tiling body, by finding δ such that the probability on the left hand side of Lemma 11 is at least $\Omega(1)$; this would imply that $\mathsf{area}(\partial D) \geqslant \Omega(1/\sqrt{\delta})$.

2.2 Basic useful properties of tiling bodies

▶ **Lemma 12.** *Let $D \subseteq \mathbb{R}^n$ be a permutation-symmetric body, such that for all $z \in \mathbb{Z}^n \setminus \{0\}$ we have $D \cap (D + z) = \emptyset$, and let $x \in D$. Then for every $1 \leqslant i, j \leqslant n$, if $x_i - x_j \in \mathbb{Z}$, then $x_i = x_j$.*

Proof. Assume towards contradiction $x_i - x_j$ is a non-zero integer k, and let $S_{i,j} \in S_n$ be the permutation that maps i to j, j to i and has any $r \neq i, j$ as a fixed point. Since D is permutation-symmetric, we have that $S_{i,j}(x) \in D$. Also, we have

$$x - S_{i,j}(x) = (x_i - x_j)(e_i - e_j) = k(e_i - e_j),$$

where e_i is the ith element in the standard basis. In other words, we get that $x = S_{i,j}(x) + z$ for non-zero $z \in \mathbb{Z}^n$, and therefore $x \in D + z$. This contradict the fact that D and $D + z$ are disjoint. ◀

▶ **Lemma 13.** *Let D be a volume 1 tiling body, and choose $a = (a_1, \dots, a_n) \in D$ uniformly at random. Then the random variable $(a_1 (\text{mod } 1), \dots, a_n (\text{mod } 1))$ is uniform over $[0, 1)^n$.*

Proof. Sample $\mathbf{x} \in [0, 1)^n$, and take $\mathbf{a} = \mathbf{x} \pmod{D}$. Note that the distribution of $\mathbf{a}$ is uniform over D. Indeed, for that we note that the map $x \to x \pmod{D}$ is bijection from $[0, 1)^n$ to D: otherwise, there were $x \neq x'$ in $[0, 1)^n$ that are equal mod D, and therefore differ by non-zero lattice point (which is clearly impossible). Now as the distribution of $\mathbf{a}$ (mod 1) is just $\mathbf{x}$, the claim follows. ◀

3 The lower bound: proof of Theorem 5

In this section, we prove the lower bound on the surface area of a permutation-symmetric tiling body D. Throughout, we will have two parameters: σ, which is magnitude of each coordinates in the needle we consider (which will be of order $\frac{\sqrt{\log n}}{n}$), and an auxiliary parameter Z (which will be of order $\frac{n}{\log n}$). Let D be a permutation-symmetric tiling body containing 0. We denote by $\mathbf{a}$ a random point in D, and by $\mathbf{u}$ a Gaussian vector $\mathcal{N}(0,\sigma^2 I_n)$. We will prove that $\Pr_{\mathbf{a},\mathbf{u}}[\ell_{\mathbf{a},\mathbf{u}} \not\subseteq D] = \Omega(1)$, which by Lemma 11 implies that $\mathsf{area}(\partial D) \geqslant \Omega(1/\sigma)$. As $\sigma = \Theta(\sqrt{\log n}/n)$, this would establish Theorem 5.

Notations

For $x, y \in \mathbb{R}$, define

$$d(x,y) := \min_{z\in\mathbb{Z}, z\neq 0} |(x+z)-y| \in [0,1].$$

To gain some intuition for the definition of $d(x,y)$, suppose x and y are two entries of a point $a \in D$. Clearly, if $d(x,y)$ is small, then x, y nearly differ by an integer $z \neq 0$, and this says that the point a is somewhat close to the boundary of D (in the sense that Lemma 12 could kick in if we move along a direction that decreases this distance).

Our argument will indeed inspect $d(a_i, a_j)$ for all distinct $i, j \in [n]$ and the way they change along a random direction. A key measure that we will keep track of is the energy of a point $a \in D$, defined by

$$\Psi(a) := \sum_{i<j} e^{-Z\cdot d(a_i,a_j)}.$$

We show that for $\mathbf{a} \in_R D$ and $\mathbf{u} \sim \mathcal{N}(0,\sigma^2 I_n)$, if $\ell_{\mathbf{a},\mathbf{u}} \subseteq D$ with probability close to 1, then the energy of $\mathbf{a}$ increases along the line $\ell_{\mathbf{a},\mathbf{u}}$ with high probability, and in particular that $\Psi(\mathbf{a}+\mathbf{u}) > \Psi(\mathbf{a})$. We then argue that with high probability, this should be the case for the point $\mathbf{a}$ as well as for $\mathbf{a}-\mathbf{u}$, hence $\Psi(\mathbf{a}+\mathbf{u}) > \Psi(\mathbf{a}-\mathbf{u})$ with high probability. This event however can happen with probability at most 0.5 by symmetry, hence completing the proof.

3.1 Analyzing the energy along a random line

By definition of $d(x,y)$, we either have $d(x,y) = (x+z-y)$ or $d(x,y) = -(x+z-y)$ for some $z \in \mathbb{Z}\setminus\{0\}$, and this sign determines whether x, y need to move in different directions or the same direction for $d(x,y)$ to get smaller. To capture this, we denote

$$\gamma(x,y) := \begin{cases} +1 & \text{if } d(x,y) = x+z-y \text{ for some } z\in\mathbb{Z},\ z\neq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Next, we discuss the *energy* of a configuration, which is the key concept used in the proof. Let Z be a parameter to be chosen later (of the order $n/\log n$). As stated earlier, our goal is to analyze the behaviour of $\Psi(a)$ along a random σ^2-needle from a in direction u. Towards this end, note that we expect (at least if u_i, u_j are small) that $d(a_i+u_i, a_j+u_j) = d(a_i,a_j) + \gamma(a_i,a_j)(u_j-u_j)$, hence expect $\Psi(a+u)$ to be close to

$$\Psi(a,u) := \sum_{i<j} e^{-Z\cdot(d(a_i,a_j)+\gamma(a_i,a_j)\cdot(u_i-u_j))}.$$

Indeed, this is the content of the following claim.

▷ Claim 14. Suppose $|u_i| \leqslant 1/20$ for all i, and $a + [0,1] \cdot u \subset D$, then

$$|\Psi(a+u) - \Psi(a,u)| \leqslant n^2 \cdot e^{-Z/4}.$$

Proof. We consider the contribution of each pair (i,j) to $\Psi(a+u)$ and $\Psi(a,u)$ separately. Without loss of generality we may only consider pairs i,j that $\gamma(a_i,a_j) = 1$, and thus $d(a_i,a_j) = a_i - a_j + z$ for some $z \in \mathbb{Z}$, $z \neq 0$. Let

$$d = a_i - a_j + z + (u_i - u_j) = (a_i + u_i) - (a_j + u_j) + z.$$

First, we argue that $d \geqslant 0$. Otherwise, since $a_i - a_j + z \geqslant 0$ it follows by continuity that there is $\lambda \in [0,1]$ such that $a_i - a_j + z + \lambda(u_i - u_j) = 0$, and hence the point $a + \lambda u$ has entries that differ by an integer $z \neq 0$, and this contradicts Lemma 12 (as $a + \lambda u \in D$). We now consider two cases:

- **Case 1**: $d \in [0, 0.5]$. In this case, we have $d(a_i+u_i, a_j+u_j) = d$, and thus the contribution of the pair (i,j) to both sums is the same ($e^{-Z \cdot d}$).
- **Case 2**: $d > 0.5$. Since $|u_i - u_j| \leqslant 0.1$, it follows that $d(a_i,a_j) = d - (u_i - u_j) > 0.4$, which implies $d(a_i + u_i, a_j + u_j) > 0.3$. Therefore, the contribution to $\Psi(a,u)$ from i,j is at most $e^{-0.4 \cdot Z}$ and to $\Psi(a+u)$ is at most $e^{-0.3 \cdot Z}$, and in particular (i,j) contributes (in absolute value) at most $e^{-Z/4}$ to the difference between the sums.

Taking a sum over all pairs (i,j) concludes the proof. ◁

3.2 Analyzing the expectation and variance of $\Psi(a, \mathrm{u})$

Next, we consider $\Psi(a,\mathbf{u})$ as a random variable over the choice of $\mathbf{u}$ and compute its expectation and variance. In both computations we will use the well-known fact that $\mathbb{E}[e^{-Z \cdot N(0,c^2)}] = e^{Z^2c^2/2}$ for all $c > 0$.

▷ Claim 15. For every $a \in \mathbb{R}^n$ we have $\mathbb{E}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2 I_n)}[\Psi(a,\mathbf{u})] = \Psi(a) \cdot e^{(Z \cdot \sigma)^2}$.

Proof. By linearity of expectation we have that

$$\mathop{\mathbb{E}}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2 I_n)}[\Psi(a,\mathbf{u})] = \sum_{i<j} e^{-Z \cdot d(a_i,a_j)} \cdot \mathop{\mathbb{E}}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2 I_n)}\left[e^{-Z \cdot \gamma(a_i,a_j) \cdot (\mathbf{u}_i - \mathbf{u}_j)}\right].$$

Note that the above expectation does not depend on i,j: for every i,j the distribution of $\mathbf{u}_i - \mathbf{u}_j$ is $N(0,\sigma^2) - N(0,\sigma^2) \sim N(0, 2\sigma^2)$, so it is symmetric around 0 and thus the sign $\gamma(a_i,a_j)$ does not affect the expectation. Hence we have

$$\mathop{\mathbb{E}}_{\mathbf{u} \sim \mathcal{N}(0,\sigma^2 I_n)}[\Psi(a,\mathbf{u})] = \Psi(a) \cdot \mathbb{E}[e^{Z \cdot N(0,2\sigma^2)}] = \Psi(a) \cdot e^{Z^2\sigma^2}. \qquad ◁$$

Next, we turn our attention into upper bounding the variance of $\Psi(a,\mathbf{u})$, and for that we first define the notion of *good* points $a \in D$ and prove two preliminary claims. We say a point a is *good* if any interval of length $(10 \log n)/n$ on the torus contains at least $\log n$ and at most $100 \log n$ coordinates from $a (\text{mod } 1)$. Note by Lemma 13, if $\mathbf{a}$ is chosen randomly from D then $\mathbf{a} \pmod 1$ is uniform over $[0,1)^n$ and by Chernoff bound is easily shown to be good with probability > 0.999.

We first show that good points have high energy.

▷ Claim 16. There exists $c_2 > 0$, such that for $Z = 0.1\frac{\log n}{n}$, if a is good then $\Psi(a) > c_2 \log^2 n$.

Proof. Partition the torus $[0,1)$ into $m = n/(10 \log n)$ disjoint intervals of length $1/m = (10 \log n)/n$ each. We say that I_i is *unanimous*, if there is $b_i \in \mathbb{R}$ (called anchor) such that (1) $b_i (\text{mod } 1)$ is the middle of I_i, and (2) for the majority of points $a_j \in I_i$, $|a_j - b_i| < 1/m$.

We consider two cases:

Case 1: There is an interval I_i that is not unanimous. Note that there are at least $\log n$ coordinates j of a such that $a_j \in I_i$. Let $j^\star$ be such coordinate, and write $a_{j^\star} = z_{j^\star} + \{a_{j^\star}\}$ where $z_{j^\star} \in \mathbb{Z}$ and $\{a_{j^\star}\}$ is the fractional part of $a_{j^\star}$. Consider $b = z_{j^\star} + m_i$ where m_i is the middle of I_i. Then since I_i is not unanimous, b is not an anchor of it and so there are at least $\frac{1}{2}\log n$ coordinates of a, say $(a_k)_{k\in K_{i,j^\star}}$ that mod 1 are in I_i, and $|a_k - b| \geqslant 1/m$. Writing $a_k = z_k + \{a_k\}$, we observe that $z_k \neq z_{j^\star}$, since otherwise $|a_k - b| = |\{a_k\} - m_i| \leqslant 1/(2m)$. Hence the difference $a_k - a_{j^\star}$ is $10\log n/n$ close to an integer $z_k - z_{j^\star} \neq 0$, and so $d(a_k, a_{j^\star}) \leqslant 10\log n/n$, and the contribution of $\Psi(a)$ is at least e^{-1}. Summing we get

$$\Psi(a) \geqslant \frac{1}{2}\sum_{j^\star : a_{j^\star}\in I_i}\ \sum_{k\in K_{i,j^\star}} e^{-Zd(a_k,a_{j^\star})} \geqslant \frac{1}{2}\sum_{j:a_j\in I_i} e^{-1}\,|K_{i,j^\star}| \geqslant \frac{1}{4e}\log^2 n.$$

Case 2: All intervals are unanimous. Let b_i be an anchor of I_i. Note that since the fractional part of two adjacent anchors, i.e. of b_i, b_{i+1}, are $1/m$ apart, we have that either $|b_i - b_{i+1}| \leqslant 1/m$ or $|b_i - b_{i+1}| \geqslant 1 - 1/m$. We claim there exists i for which the latter condition holds. To see this, assume that for all $i = 1,\dots,m-1$ we have that the first condition holds. Then we have $b_i = z + i\frac{10\log n}{n}$ for some $z\in\mathbb{Z}$ for all $i = 1,\dots,m$, and hence $|b_m - b_1| \geqslant 1 - 1/m$ (and the condition holds for $i = m$).

Thus, we fix i such that $|b_i - b_{i+1}| \geqslant 1 - 1/m$, and thus $b_i - b_{i+1} = z + \alpha$ for $z \neq 0$ and $|\alpha| \leqslant 1/m$. Let K_i be the coordinates j of a such that $|a_j - b_i| \leqslant 1/m$ for $j \in K_i$ and similarly define K_{i+1}. We have that $a_r - a_j = z + \alpha + (a_r - b_{i+1}) + (a_j - b_i)$, hence $a_r - a_j = z + \beta$ for $|\beta| \leqslant 3/m$ for all $r \in K_{i+1}$, $j \in K_i$. Thus $d(a_r, a_j) \leqslant 3/m$, and we get

$$\Psi(a) \geqslant |K_i|\,|K_{i+1}|\,e^{-Z\cdot 3/m} \geqslant \frac{1}{4}e^{-3}\log^2 n \qquad \triangleleft$$

Let $C_i = \sum_{j\neq i} e^{-Z\cdot d(a_i,a_j)}$ be the contribution of a_i to $\Psi(a)$. Note that $\Psi(a) = \frac{1}{2}\sum_i C_i$.

▷ Claim 17. There exists $c_3 > 0$, such that if a is good, then for all i we have $C_i < c_3\Psi(a)/\log n$.

Proof. Note that $d(a_i, a_j) \geqslant |\{a_i\} - \{a_j\}|$. Since any interval of length $10\log n/n$ on the torus contains at most $100\log n$ points of a, we have that the number of j's such that $|\{a_i\} - \{a_j\}|$ is between $10\log n/n\cdot k$ and $10\log n/n\cdot(k+1)$ is at most $200\log n$ (for all k). Therefore,

$$C_i < 200\log n \cdot \sum_{k=0}^{\infty} e^{-Z\cdot k\cdot(10\log n)/n} = 200\log n\cdot\sum_{k=0}^{\infty} e^{-k} \leqslant 400\log n.$$

Using Claim 16, we may bound $\log n \leqslant \frac{1}{c_2}\frac{\Psi(a)}{\log n}$, finishing the proof. $\triangleleft$

We are now ready to bound the variance of $\Psi(a, \mathbf{u})$.

▷ Claim 18. There exists $c_1 > 0$ such that the following holds. Let $Z = n/10\log n$, let $a \in \mathbb{R}^n$ be good and let $\mathbf{u} \sim \mathcal{N}(0, \sigma^2 I_n)$. Then

$$\mathsf{var}_{\mathbf{u}}[\Psi(a,\mathbf{u})] \leqslant \frac{c_1}{\log n}\cdot\left(e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2}\right)\cdot\Psi(a)^2.$$

Proof. Using Claim 15 to compute the expectation of $\Psi(a, \mathbf{u})$, we have by definition that

$$\begin{aligned}
&\mathsf{var}_{\mathbf{u}}(\Psi(a,\mathbf{u})) = \mathop{\mathbb{E}}_{\mathbf{u}}\left[\left(\sum_{i<j} e^{-Z\cdot d(a_i,a_j)}\cdot\left(e^{Z\cdot\gamma(a_i,a_j)\cdot(\mathbf{u}_i-\mathbf{u}_j)} - e^{(Z\cdot\sigma)^2}\right)\right)^2\right]\\
&= \sum_{i<j} e^{-2Z\cdot d(a_i,a_j)}\cdot\mathop{\mathbb{E}}_{\mathbf{u}}\left[\left(e^{Z\cdot\gamma(a_i,a_j)\cdot(\mathbf{u}_i-\mathbf{u}_j)} - e^{(Z\cdot\sigma)^2}\right)^2\right]+\\
&\sum_{\substack{(i,j,k)\\ \text{distinct}}} e^{-Z\cdot(d(a_i,a_j)+d(a_i,a_k))}\\
&\quad\cdot\mathop{\mathbb{E}}_{\mathbf{u}}\left[(e^{Z\cdot\gamma(a_i,a_j)\cdot(\mathbf{u}_i-\mathbf{u}_j)} - e^{(Z\cdot\sigma)^2})(e^{Z\cdot\gamma(a_i,a_k)\cdot(\mathbf{u}_i-\mathbf{u}_k)} - e^{(Z\cdot\sigma)^2})\right].
\end{aligned}$$

Here, we used that fact that if i, j, k, r are distinct then $e^{Z\cdot\gamma(a_i,a_j)\cdot(\mathbf{u}_i-\mathbf{u}_j)}$, $e^{Z\cdot\gamma(a_k,a_r)\cdot(\mathbf{u}_k-\mathbf{u}_r)}$ are independent with expectation $e^{(Z\cdot\sigma)^2}$, hence the contribution of these terms is 0. Computing, we see that

$$\mathop{\mathbb{E}}_{\mathbf{u}}\left[\left(e^{Z\cdot\gamma(a_i,a_j)\cdot(\mathbf{u}_i-\mathbf{u}_j)} - e^{(Z\cdot\sigma)^2}\right)^2\right] = \mathbb{E}\left[e^{Z\cdot N(0,8\sigma^2)}\right] - e^{2(Z\cdot\sigma)^2} = e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2},$$

and

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{u}}\left[(e^{Z\cdot\gamma(a_i,a_j)\cdot(\mathbf{u}_i-\mathbf{u}_j)} - e^{(Z\cdot\sigma)^2})(e^{Z\cdot\gamma(a_i,a_k)\cdot(\mathbf{u}_i-\mathbf{u}_k)} - e^{(Z\cdot\sigma)^2})\right]\\
&= \mathbb{E}\left[e^{(\gamma(a_i,a_j)+\gamma(a_i,a_k))Z\cdot N(0,\sigma^2)}\right]\mathbb{E}\left[e^{Z\cdot N(0,2\sigma^2)}\right] - e^{2(Z\cdot\sigma)^2}\\
&\leqslant \mathbb{E}\left[e^{2Z\cdot N(0,\sigma^2)}\right]\mathbb{E}\left[e^{Z\cdot N(0,2\sigma^2)}\right] - e^{2(Z\cdot\sigma)^2}\\
&= e^{3(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2}.
\end{aligned}$$

Thus, we get that

$$\begin{aligned}
&\mathsf{var}_{\mathbf{u}}(\Psi(a,\mathbf{u}))\\
&\leqslant \sum_{i<j} e^{-2Z\cdot d(a_i,a_j)}(e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2})\\
&+ \sum_{(i,j,k)\text{ distinct}} e^{-Z\cdot(d(a_i,a_j)+d(a_i,a_k))}(e^{3(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2})\\
&\leqslant (e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2})\sum_i\left(\sum_{j\neq i} e^{-2Z\cdot d(a_i,a_j)} + \sum_{j,k\neq i} e^{-Z\cdot(d(a_i,a_j)+d(a_i,a_k))}\right)\\
&= (e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2})\sum_i\left(\sum_{j\neq i} e^{-2Z\cdot d(a_i,a_j)}\right)^2\\
&= (e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2})\cdot\sum_i C_i^2.
\end{aligned}$$

Therefore, using Claim 17 we conclude that

$$\mathsf{var}_{\mathbf{u}}(\Psi(a,\mathbf{u})) \leqslant (e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2})\frac{c_3\Psi(a)}{\log n}\cdot\sum_i C_i = \frac{2c_3}{\log n}\cdot(e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2})\cdot\Psi(a)^2.$$

Setting $c_1 := 2c_3$ completes the proof. ◁

Putting the last two claims together, we have:

▷ Claim 19. Let $\sigma = 10^4\sqrt{c_1}\frac{\sqrt{\log n}}{n}$ and let $a \in \mathbb{R}^n$ be good. Then

$$\Pr_{\mathbf{u}}[\Psi(a,\mathbf{u}) > \Psi(a) + \frac{(Z\sigma)^4}{2}\Psi(a)] \geqslant 0.96.$$

Proof. We upper bound the probability of the complement event. Using Claim 15 (and $e^t \geqslant 1 + t + t^2/2$ for $t \geqslant 0$), we get

$$\mathop{\mathbb{E}}_{\mathbf{u}}[\Psi(a,\mathbf{u})] \geqslant \Psi(a) \cdot \left(1 + (Z\sigma)^2 + \frac{(Z\sigma)^4}{2}\right).$$

Hence

$$\Pr_{\mathbf{u}}\left[\Psi(a,\mathbf{u}) \leqslant \Psi(a) + \frac{(Z\sigma)^4}{2}\Psi(a)\right] \leqslant \Pr_{\mathbf{u}}\left[\left|\Psi(a,\mathbf{u}) - \mathop{\mathbb{E}}_{\mathbf{u}'}[\Psi(a,\mathbf{u}')]\right| \geqslant \Psi(a)\cdot(Z\sigma)^2\right].$$

We want to upper bound the probability of the last event using Chebyshev's inequality. Since a is good, the conclusion of Claim 18 holds. Since $Z\sigma = o(1)$, for large enough n we get

$$\mathsf{var}_{\mathbf{u}}[\Psi(a,\mathbf{u})] \leqslant \frac{c_1}{\log n}\cdot\left(e^{4(Z\cdot\sigma)^2} - e^{2(Z\cdot\sigma)^2}\right)\cdot\Psi(a)^2 \leqslant \frac{c_1}{\log n}\cdot\Psi(a)^2\cdot 8(Z\sigma)^2.$$

Therefore, applying Chebyshev's inequality we see the probability in question is at most

$$\frac{\mathsf{var}_{\mathbf{u}}[\Psi(a,\mathbf{u})]}{\Psi(a)^2\cdot(Z\sigma)^4} \leqslant \frac{c_1\cdot\Psi(a)^2\cdot 8(Z\sigma)^2}{(\log n)\cdot\Psi(a)^2\cdot(Z\sigma)^4} = \frac{8c_1}{(\log n)\cdot(Z\sigma)^2} = \frac{4c_1}{10^2 c_1} = 0.04. \qquad \triangleleft$$

3.3 Finishing the argument

For each u, denote $\varepsilon_u = \Pr_{\mathbf{a}\in D}[\ell_{\mathbf{a},u} \not\subseteq D]$, $\varepsilon = \mathbb{E}_{\mathbf{u}\sim\mathcal{N}(0,\sigma^2 I_n)}[\varepsilon_{\mathbf{u}}] = \Pr_{\mathbf{a},\mathbf{u}}[\ell_{\mathbf{a},\mathbf{u}} \not\subseteq D]$.

▷ Claim 20. For each u, $\mathcal{D}_{TV}[\mathbf{a};\mathbf{a}-u] \leqslant \varepsilon_u + \varepsilon_{-u}$.

Proof. Let K be a Borel set. Note that it is enough to show that (1) if $K \subseteq D$ then $0 \leqslant \Pr_{\mathbf{a}\in D}[\mathbf{a}\in K] - \Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K] \leqslant \varepsilon_u$, and (2) if $K \subseteq \bar{D}$, then $-\varepsilon_{-u} \leqslant \Pr_{\mathbf{a}\in D}[\mathbf{a}\in K] - \Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K] \leqslant 0$. Indeed, given both (1) and (2), the triangle inequality implies for any Borel set $K \subseteq \mathbb{R}^n$,

$$\begin{aligned}&\left|\Pr_{\mathbf{a}\in D}[\mathbf{a}\in K] - \Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K]\right|\\ &\leqslant \left|\Pr_{\mathbf{a}\in D}[\mathbf{a}\in K\cap D] - \Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K\cap D] + \Pr_{\mathbf{a}\in D}[\mathbf{a}\in K\setminus D] - \Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K\setminus D]\right|\\ &\leqslant \varepsilon_u + \varepsilon_{-u}.\end{aligned}$$

To prove (1), note that $\Pr_{\mathbf{a}\in D}[\mathbf{a}\in K] = \mu(K)$ and

$$\Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K] = \Pr_{\mathbf{a}\in D}[\mathbf{a}\in K+u] = \mu((K+u)\cap D).$$

This is at most $\mu(K+u) = \mu(K)$ (hence the expression in (1) is non-negative) and at least $\geqslant \mu(K+u) - \mu((K+u)\setminus D) = \mu(K) - \mu(K\setminus(D-u))$. Therefore

$$0 \leqslant \Pr_{\mathbf{a}\in D}[\mathbf{a}\in K] - \Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K] \leqslant \mu(K\setminus(D-u)) \leqslant \mu(D\setminus(D-u)) = \Pr_{\mathbf{a}\in D}[\mathbf{a}+u\notin D] \leqslant \varepsilon_u.$$

To prove (2), note that $\Pr_{\mathbf{a}\in D}[\mathbf{a}\in K] = 0$ (hence the expression in (2) is non-positive) and

$$\Pr_{\mathbf{a}\in D}[\mathbf{a}-u\in K] \leqslant \Pr_{\mathbf{a}\in D}[\mathbf{a}-u\notin D] \leqslant \varepsilon_{-u}. \qquad \triangleleft$$

▷ Claim 21. $\varepsilon \geqslant 0.1$.

Proof. Let E_1 be the event that $\mathbf{a}+\mathbf{u}[0,1] \subseteq D$, let E_2 be the event that $\Psi(\mathbf{a}) \leqslant 1$, let E_3 be the event that $|\mathbf{u}_i| > 1/20$ for some i and let E_4 be the event that $\Psi(\mathbf{a},\mathbf{u}) > \Psi(\mathbf{a})+\frac{(Z\sigma)^4}{2}\Psi(\mathbf{a})$. Finally, let E_5 be the event that $\Psi(\mathbf{a}+\mathbf{u}) > \Psi(\mathbf{a})$ and denote $E(\mathbf{a},\mathbf{u}) = E_1 \cap (\neg E_2) \cap (\neg E_3) \cap E_4$. Note that if the event E holds for a, u, then E_5 also holds, since by Claim 14:

$$\Psi(a+u) \geqslant \Psi(a,u) - n^2 \cdot e^{-Z/4} > \Psi(a) + \frac{(Z\sigma)^4}{2}\Psi(a) - n^2 \cdot e^{-Z/4} \geqslant \Psi(a).$$

By Claim 16 the probability of E_2 is at most the probability $\mathbf{a}$ is bad, hence it is at most 0.005. By definition, the probability of E_1 is $1-\varepsilon$. By the union bound and Chernoff inequality, the probability of E_3 is $o(1)$. Thus, by Claim 19 we have

$$\Pr_{\mathbf{a},\mathbf{u}}[E(\mathbf{a},\mathbf{u})] \geqslant 0.96 - \varepsilon - 0.005 - o(1) \geqslant 0.95 - \varepsilon. \tag{1}$$

Fix u. Using Claim 20 we get that

$$\Pr_{\mathbf{a}}\left[E(\mathbf{a}-u,u)\right] \geqslant \Pr_{\mathbf{a}}\left[E(\mathbf{a},u)\right] - \mathcal{D}_{TV}\left[\mathbf{a};\mathbf{a}-u\right] \geqslant \Pr_{\mathbf{a}}\left[E(\mathbf{a},u)\right] - \varepsilon_u - \varepsilon_{-u}.$$

By the union bound, we now conclude that

$$\Pr_{\mathbf{a}}\left[E(\mathbf{a}-u,u) \cap E(\mathbf{a},u)\right] \geqslant 1 - \Pr_{\mathbf{a}}\left[\overline{E(\mathbf{a}-u,u)}\right] - \Pr_{\mathbf{a}}\left[\overline{E(\mathbf{a},u)}\right] \geqslant 2\Pr_{\mathbf{a}}\left[E(\mathbf{a},u)\right] - 1 - \varepsilon_u - \varepsilon_{-u}.$$

Taking expectation over $\mathbf{u}$, we get that

$$\Pr_{\mathbf{a},\mathbf{u}}\left[E(\mathbf{a}-\mathbf{u},\mathbf{u}) \cap E(\mathbf{a},\mathbf{u})\right] \geqslant 2\Pr_{\mathbf{a},\mathbf{u}}\left[E(\mathbf{a},\mathbf{u})\right] - 1 - 2\mathop{\mathbb{E}}_{\mathbf{u}}\left[\varepsilon_{\mathbf{u}}\right] \geqslant 0.9 - 4\varepsilon.$$

Next, when both $E(a-u,u)$ and $E(a,u)$ hold, we have by the previous observation that E_5 holds for both pairs $(a-u,u)$ and (a,u), and so $\Psi(a+u) > \Psi(a) = \Psi((a-u)+u) > \Psi(a-u)$. Thus, we get that $\Pr_{\mathbf{a},\mathbf{u}}\left[\Psi(\mathbf{a}+\mathbf{u}) > \Psi(\mathbf{a}-\mathbf{u})\right] \geqslant 0.9 - 4\varepsilon$. On the other hand, the probability on the left hand side is at most 0.5; this follows as $\Pr_{\mathbf{a},\mathbf{u}}\left[\Psi(\mathbf{a}+\mathbf{u}) > \Psi(\mathbf{a}-\mathbf{u})\right] = \Pr_{\mathbf{a},\mathbf{u}}\left[\Psi(\mathbf{a}-\mathbf{u}) > \Psi(\mathbf{a}+\mathbf{u})\right]$ (since the distributions of $\mathbf{u}$ and $-\mathbf{u}$ are identical) and their sum is at most 1. Combining the two inequalities we get that $\varepsilon \geqslant 0.1$. ◁

4 The upper bound: proof of Theorem 6

In this section we prove a matching upper bound on the surface area of a permutation-symmetric foam by giving a (probabilistic) construction of a permutation-symmetric tiling body D of surface area $O(n/\sqrt{\log n})$. The main technical result proved in this section, Lemma 23, establishes a weaker statement, and in Section B we show how to deduce Theorem 6 from it.

4.1 Reduction to constructing a rounding scheme

Suppose S is function mapping (multi-)sets of n points from $\mathbb{R}/\mathbb{Z}$, to $\mathbb{R}/\mathbb{Z}$. We further assume that for all (multi-)sets A, it holds that $S(A) \notin \{0\} \cup A$.

Given such S, we may extend it to $\mathbb{R}^n$ by $S(x_1,\ldots,x_n) := S(\{\{x_1\},\ldots,\{x_n\}\})$, where $\{x_i\}$ is the fractional part of x. We can construct a rounding scheme $R\colon \mathbb{R}^n \to \mathbb{Z}^n$ using S as follows.

- On input $x = (x_1, \ldots, x_n)$, denote $z = S(x)$ and view z as a number in $[0, 1)$.
- For each $i \in [n]$:
  - if $\{x_i\} \in [0, z)$, set $R(x)_i = \lfloor x_i \rfloor$,
  - otherwise, $\{x_i\} \in (z, 1)$, and set $R(x)_i = \lceil x_i \rceil$.

First, R is well-defined since $z \notin \{0, \{x_1\}, \ldots, \{x_n\}\}$. Next, note that for any $t \in \mathbb{Z}^n$ it holds that $R(x + t) = R(x) + t$, thus R induces that the body $D = \{x \mid R(x) = 0\}$ is tiling with respect to the lattice $\mathbb{Z}^n$. Last, we note that since for any $\pi \in S_n$ we have that $S(\pi(x)) = S(x)$, we also have that $R(\pi(x)) = \pi(R(x))$, and hence D is permutation-symmetric.

In our proof we will define a distribution over mappings S, and we will want to study the noise sensitivity of the resulting body D using properties of the mappings S. The following claim gives useful conditions to study noise sensitivity in terms of mapping S.

▷ Claim 22. Let x and $x + \Delta$ two points in $\mathbb{R}^n$. Suppose that

1. $S(x) = S(x + \Delta) =: z$; and
2. for all i, $\{x_i + \lambda\Delta_i\} \neq z$, $\forall \lambda \in [0, 1]$.

Then the points $x, x + \Delta$ fall in the same cell in the tiling induced by D.

Proof. Suppose towards contradiction that the conclusion of the statement does not hold, i.e. x and $x + \Delta$ belong to different cells in the tiling induced by D. Thus, the rounding function R when applied on x and on $x + \Delta$ should produce different lattice points, so there is an i such that $R(x)_i \neq R(x + \Delta)_i$. We fix that i and assume without loss of generality that $\Delta_i \geqslant 0$ and that $x_i \in [0, 1)$. We now consider two cases, depending on the range x_i falls into:

1. If $x_i \in [0, z)$, then by definition of R we get that $R(x)_i = 0$, and $R(x + \Delta)_i = 0$ unless $x_i + \Delta_i > z$, which leads to a contradiction to the second condition (z is on the interval between x_i and $x_i + \Delta_i$).
2. If $x_i \in (z, 1)$, then $R(x)_i = 1$, and $R(x + \Delta)_i = 1$ unless $x_i + \Delta_i > 1 + z$, which again leads to a contradiction to the second condition ($1 + z$ is on the interval between x_i and $x_i + \Delta_i$). ◀

Our main technical statement is the following lemma.

▶ **Lemma 23.** *There exists a distribution over mappings $(S_{\vec{r}})_{\vec{r}}$ ($\vec{r}$ is a vector of randomness) such that for small enough $\varepsilon > 0$, setting $\sigma = \varepsilon\frac{\sqrt{\log n}}{n}$ we have*

$$\mathop{\mathbb{E}}_{\vec{r}}\left[\Pr_{\mathbf{x},\mathbf{\Delta}\sim\mathcal{N}(0,\sigma^2 I_n)}\left[\text{Conditions of Claim 22 hold for } \mathbf{x} \text{ and } \mathbf{x}+\mathbf{\Delta}\right]\right] \geqslant 1 - O(\varepsilon).$$

Deducing from Theorem 6 from Lemma 23 mostly involves measure-theoretic arguments, and we defer this deduction to Section B. We will actually need the following slightly more informative version of Lemma 23 above, using the reduction from mappings to tilings presented in the beginning of this section, and an inspection of the bodies $D_{\vec{r}}$ our proof gives.

▶ **Lemma 24.** *There exists a distribution over tiling bodies $(D_{\vec{r}})_{\vec{r}}$ such that*

1. *For small enough $\varepsilon > 0$, we have*

$$\mathop{\mathbb{E}}_{\vec{r}}\left[\Pr_{\mathbf{x},\mathbf{\Delta}\sim\mathcal{N}(0,\varepsilon^2 I_n)}\left[\text{At least one of the conditions of Claim 22 fail for } \mathbf{x} \text{ and } \mathbf{x}+\mathbf{\Delta}\right]\right] \lesssim \frac{n}{\sqrt{\log n}}\varepsilon.$$

2. *For each $\vec{r}$, $D_{\vec{r}}$ is a countable union of semi-algebraic sets (i.e., sets defined by finitely many polynomial inequalities).*

4.2 The construction of $S_{\vec{r}}$

4.2.1 Overview

Before jumping into the technical details, we start with some intuition. Recall that on input x (a set of n points from $\mathbb{R}/\mathbb{Z}$) we must output a number $z \in \mathbb{R}/\mathbb{Z}$, and our goal is to minimize the probability so that the conditions of Claim 22 fail on a short needle $\ell_{x,\Delta}$. Note that it would not be beneficial for us to choose z that are close to x_i. For example, if we chose z such that $|x_i - z_i| \leqslant \sigma$, then there is constant probability that the interval $\{x_i + \lambda\Delta_i\}_{\lambda\in[0,1]}$ would contain the point z, i.e. the second condition of Claim 22 would fail.

Thus, a natural candidate for the choice of z would be the maximizer of $\min_{i\in[n]} |x_i - z_i|$. It is not hard to see that this minimum is typically of the order $\log n/n$, so intuitively the second condition of Claim 22 should hold with probability $\geqslant 1-\varepsilon$. However, such choice for z would not be very stable: it is typically the case that there are numerous $z_1, \dots, z_r$ that nearly achieve this maximum, thus the maximizer among them could change when looking at $x + \Delta$ (i.e., this event would happen with probability significantly more than ε), leading to a failure of the first condition of Claim 22.

We must therefore assign each one of the near-maximizer $z_1, \dots, z_r$ some weight, so that the weight of each one of them does not significantly change when moving to $x + \Delta$. A general form of construction of this type is to design a scoring function $f\colon [0,\infty] \to [0,1]$, and given an input x to assign the weight $w(z) = \prod\limits_i f(|x_i - z|)$ to each z, and sample z with probability proportional to $w(z)$.

We remark that this general recipe essentially captures our (natural) attempts so far. On the one hand, we want f to penalize z if it is very close to x_i, hence we want $f(t)$ at least mildly increasing. On the other hand, if f is very sharply increasing (e.g exponential), then one runs into the same problems as we had when we thought of picking z that maximizes $\min_{i\in[n]} |x_i - z_i|$. We are thus led to consider "mildly increasing" scoring functions f, and polynomials turn out to be good choice. Indeed, our scoring function f will be "trivial" if $|x_i - z|$ is too small or too large (i.e. it'll be 0 if $|x_i - z| \leqslant \frac{\log n}{50n}$ and 1 if $|x_i - z| \geqslant \frac{\log n}{25n}$), and otherwise behaves cubically.

4.2.2 A basic scoring function

Our construction of $(S_{\vec{r}})_{\vec{r}}$ uses a non-negative scoring function f with the following properties.

▶ **Fact 25.** *There exists a function $f\colon [0,\infty) \to [0,1]$ that is twice differentiable with continuous second derivative with the following properties:*

1. $f(t) = 0$ *if* $t \leqslant 1$.
2. $f(t) = 1$ *if* $t \geqslant 2$.
3. $f(t) \asymp (t-1)^3$ *if* $1 \leqslant t \leqslant 2$.
4. $|f'(t)| \lesssim t^2$ *and* $|f''(t)| \lesssim t$ *for all* t.

Exhibiting function f as in Fact 25 is not hard, and we omit the proof. The function f defined by $f(t) = (t-1)^3$ if $1 \leqslant t \leqslant 2$ and $f(t) = 0$ for $t \leqslant 1$, $f(t) = 1$ for $t \geqslant 2$ is almost enough, except that it is not differentiable at $t = 1$. One can fix by convolving a smooth bump function with compact support.

Next, we wish to define the mapping $S_{\vec{r}}$. We view the input x as a multi-set, and the randomness vector $\vec{r}$ as an infinite sequence of (i, h) where i is a uniformly random element from $[m]$ and h is a uniform real-number from $[0,1]$.

Set $m = n^{1/3}$, partition the circle the circle $\mathbb{R}/\mathbb{Z}$ into m intervals of length $1/m$ each, $I_j := \left[\frac{j-1}{m}, \frac{j}{m}\right]$, and let $z_j = \frac{j-1/2}{m}$ be the middle of I_j. It will be convenient for us to define $g_j(t) = f(\frac{50n}{\log n}\left|t - z_j\right|)$, and subsequently $r_j(x) := \prod\limits_{y \in I_j \cap x} g_j(y)$. There two cases:

4.2.2.1 Case (A): $r_i(x) \neq 0$ for some $i \in [m]$

In this case, we define a probability distribution $p_i(x)$ over the i's proportionally to the $r_i(x)$'s, i.e. we define $p_i(x) := \frac{r_i(x)}{\sum_i r_i(x)}$. We now perform correlated sampling of $i \in [m]$ according to $p_i(x)$ using the randomness vector $\vec{r}$. More precisely, we go over the randomness vector $\vec{r} = (i_1, h_1), (i_2, h_2), \ldots$ and find the smallest j such that $h_j \leqslant p_{i_j}(x)$, in which case we choose $i = i_j$. We then define $S_{\vec{r}}(x) = z_{i_j}$.

4.2.2.2 Case (B): $r_i(x) = 0$ for all $i \in [m]$

If $1/2 \notin x$, we define $S_{\vec{r}}(x) = 1/2$. Otherwise, we define $S_r(x) = z$, where z is the first element from $\{\frac{1}{4n}, \frac{3}{4n}, \ldots, \frac{4n-1}{4n}\}$ that is at least $\frac{1}{4n}$-away from all the entries of x.

4.3 Estimating g_j on close points

▶ **Fact 26.** *Let $j \in [m]$ and $x_i \in [z_j - \frac{\log n}{25n} - \varepsilon^{0.95}, z_j + \frac{\log n}{25n} + \varepsilon^{0.95}] \setminus [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]$, $\Delta_i \in \mathbb{R}$, and denote $\alpha_i = \mathsf{dist}\left(x_i, [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]\right)$.*

1. *If $\alpha_i \geqslant 2\left|\Delta_i\right|$, then $\left|g_j(x_i + \Delta_i) - g_j(x_i)\right| \lesssim \frac{\left|\Delta_i\right|}{\alpha_i} g_j(x_i)$.*
2. *In general, $\left|g_j(x_i + \Delta_i) - g_j(x_i)\right| \lesssim n^3(\alpha_i^3 + \left|\Delta_i\right|^3)$.*

Proof. Using Taylor's approximation with remainder, there is $y_i \in [x_i, x_i + \Delta_i]$ such that $g_j(x_i + \Delta_i) = g_j(x_i) + g_j'(y_i)\Delta_i$, hence

$$\begin{aligned}\left|g_j(x_i + \Delta_i) - g_j(x_i)\right| \lesssim \left|\Delta_i\right| \left|g_j'(y_i)\right| &\lesssim \left|\Delta_i\right| \frac{50n}{\log n} f'\left(\frac{50n}{\log n}\left|y_i - z_j\right|\right) \\ &\lesssim \frac{n}{\log n}\left|\Delta_i\right| \left(\frac{50n}{\log n}\left|y_i - z_j\right| - 1\right)^2.\end{aligned}$$

For the second item, since $y_i \in [x_i, x_i + \Delta_i]$, we get that $\left|\frac{50n}{\log n}\left|y_i - z_j\right| - 1\right| \leqslant \frac{50n}{\log n}(\alpha_i + \left|\Delta_i\right|)$, and plugging that in yields

$$\left|g_j(x_i + \Delta_i) - g_j(x_i)\right| \lesssim n^3 \left|\Delta_i\right| (\alpha_i^2 + \Delta_i^2) \lesssim n^3(\alpha_i^3 + \left|\Delta_i\right|^3),$$

where the last inequality holds as $ab \lesssim a^3 + b^{3/2}$ for all $a, b > 0$ (Young's inequality). For the first item, note that since $y_i \in [x_i, x_i + \Delta_i]$ we get that $\left(\frac{50n}{\log n}\left|y_i - z_j\right| - 1\right) \geqslant \frac{50n}{\log n}(\alpha_i - \left|\Delta_i\right|)$, and by the lower bound on α_i this is $\geqslant \frac{25n}{\log n}\alpha_i$. Therefore we may continue as

$$\left|g_j(x_i + \Delta_i) - g_j(x_i)\right| \lesssim \frac{n}{\log n}\left|\Delta_i\right| \left(\frac{50n}{\log n}\left|y_i - z_j\right| - 1\right)^2 \lesssim \frac{\left|\Delta_i\right|}{\alpha_i}\left(\frac{50n}{\log n}\left|y_i - z_j\right| - 1\right)^3.$$

Also, we have that $\left(\frac{50n}{\log n}\left|y_i - z_j\right| - 1\right) \leqslant \frac{50n}{\log n}(\alpha_i + \left|\Delta_i\right|) \lesssim \frac{n}{\log n}\alpha_i$, so

$$\left|g_j(x_i + \Delta_i) - g_j(x_i)\right| \lesssim \frac{\left|\Delta_i\right|}{\alpha_i}\left(\frac{n}{\log n}\alpha_i\right)^3 \lesssim \frac{\left|\Delta_i\right|}{\alpha_i} g_j(x_i). \qquad ◀$$

4.4 Analysis of the construction

In this section we prove that Lemma 23 holds for the construction of $S_{\vec{r}}$ from the last section, and for that we show that for small enough ε, the expected probability of the complement event is $O(\varepsilon)$, i.e. that

$$\mathop{\mathbb{E}}_{\vec{r}}\left[\Pr_{\mathbf{x},\boldsymbol{\Delta}\sim\mathcal{N}(0,\sigma^2 I_n)}[\text{One of the conditions in Claim 22 fails for } \mathbf{x} \text{ and } \mathbf{x}+\boldsymbol{\Delta}]\right] \lesssim \varepsilon. \tag{2}$$

We will think of ε as very small (say $\varepsilon \leqslant 2^{-n^2}$), and analyze the contribution of x's from case (A) and case (B) separately. Case (A) is the main case that occurs often, and case (B) should be thought of rare.

4.4.1 Analysis of case (B)

First, we show that the probability $\mathbf{x}$ (or equivalently $\mathbf{x}+\boldsymbol{\Delta}$) falls into Case (B) is at most $n^{-\omega(1)}$. For this, it will be helpful for us to sample $\mathbf{x}$, a multi-set of n uniformly chosen numbers in $[0,1]$ in the following equivalent way:

- Sample $\mathbf{t}_1,\ldots,\mathbf{t}_m$ – where $\mathbf{t}_i$ is the number of i's such that $\mathbf{x}_i$'s that fall into interval I_i.
- Sample $\mathbf{t}_i$ points uniformly from I_i, for each $i=1,\ldots,m$.

Note that $\mathbb{E}[\mathbf{t}_i] = n/m$, hence by Chernoff bound $\Pr[\mathbf{t}_i \geqslant 2\cdot n/m] = e^{-\Omega(n/m)} = n^{-\omega(1)}$. Thus, by the union bound we have that

$$\Pr[\forall i\ \mathbf{t}_i \leqslant 2\cdot n/m] = 1 - n\cdot n^{-\omega(1)} = 1 - n^{-\omega(1)}.$$

Next, we condition on $\mathbf{t}_i = t_i$, and assume that indeed $t_i < 2\cdot n/m$ for all i. Let E_i be the event that $r_i(x)=0$. Note that conditioned on $\mathbf{t}_i = t_i$, the E_i's are independent and that

$$\begin{aligned}\Pr[\neg E_i|\ t_1,\ldots,t_m] = \Pr_{\mathbf{a}\in I_i}\left[\frac{50n}{\log n}|\mathbf{a}-z_i|\leqslant 1\right]^{t_i} &= \left(1-\frac{\log n/25n}{1/m}\right)^{t_i}\\ &\geqslant \left(1-\frac{m\log n}{25n}\right)^{2\cdot n/m}\\ &\geqslant e^{-4\log n/25} = n^{-4/25},\end{aligned} \tag{3}$$

where we used the fact that $e^{-\delta}\leqslant 1-\delta/2$ for small enough $\delta>0$. Therefore,

$$\Pr[E_1\wedge E_2\wedge\ldots\wedge E_m|\ t_1,\ldots,t_m] \leqslant (1-n^{-4/25})^m = (1-n^{-4/25})^{n^{1/3}} = n^{-\omega(1)},$$

as long as the t_i's satisfy the condition $t_i < 2\cdot n/m$. Therefore, the overall probability of case (B) is $n^{-\omega(1)}$.

Next, we analyze the probability that the conditions of Claim 22 fail given we are in case (B). Note that if the conditions of Claim 22 fail to hold, then either (I) exactly one of $\mathbf{x}$, $\mathbf{x}+\boldsymbol{\Delta}$ falls under Case (B), or (II) both $\mathbf{x}$ and $\mathbf{x}+\boldsymbol{\Delta}$ fall under Case (B), but $1/2\in\mathbf{x}+\lambda\boldsymbol{\Delta}$ for some $\lambda\in[0,1]$. We'll bound these cases separately.

4.4.1.1 Case (I)

Assuming $\mathbf{x}$ is under Case (B), we know that each of the m intervals of the form $J_i := [z_i - \frac{\log n}{50n}, z_i + \frac{\log n}{50n}]$ contains at least one point from $\mathbf{x}$. Let $\mathbf{x}_i$ be that point (if there are multiple, pick one at random). Then $\mathbf{x}_i$ is uniformly distributed in J_i. Therefore, the probability of $\mathbf{x}_i + \mathbf{\Delta}_i$ is outside J_i, where $\mathbf{\Delta}_i \sim N(0, \sigma^2)$ and $\sigma^2 \leqslant \varepsilon^2$, is $O(\varepsilon)$. Given case (B) occurs with probability $\leqslant n^{-\omega(1)}$, we conclude that its contribution to the conditions of Claim 22 failing is at most

$$O(m\varepsilon) \cdot n^{-\omega(1)} = O(\varepsilon).$$

4.4.1.2 Case (II)

Fix $\mathbf{\Delta} = \Delta$, and consider $\mathbf{x}_j$ conditioned on being in case (B). If $\mathbf{x}_j$ is in one of the intervals J_i, then its distribution is uniform over J_i, in which case we get that the probability $1/2$ falls inside the interval $[\mathbf{x}_j, \mathbf{x}_j + \Delta_j]$ is at most $m\,|\Delta_j|$. If $\mathbf{x}_j$ is not in one of the intervals J_i, then it is distributed uniformly on $[0,1] \setminus \cup_{i=1}^{m} J_i$, and the probability $1/2$ is in $[\mathbf{x}_j, \mathbf{x}_j + \Delta_j]$ is at most $2\,|\Delta_j| \leqslant m\,|\Delta_j|$.

Therefore by the union bound,

$$\Pr_{\mathbf{x}}\left[\exists j \in [n]\ 1/2 \in [\mathbf{x}_j, \mathbf{x}_j + \Delta_j] \mid \mathsf{case(B)}, \Delta\right] \leqslant m \sum_{j=1}^{n} |\Delta_i|.$$

Taking expectation over $\Delta \sim \mathcal{N}(0, \sigma^2 I_n)$ and using Cauchy-Schwarz we get that

$$\begin{aligned}\Pr_{\mathbf{x},\mathbf{\Delta}}\left[\exists j \in [n]\ \ 1/2 \in [\mathbf{x}_j, \mathbf{x}_j + \mathbf{\Delta}_j] \mid \mathsf{case(B)}\right] &\leqslant m \mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\sum_{j=1}^{n} |\mathbf{\Delta}_i|\right] \\ &\leqslant m\sqrt{n}\sqrt{\mathop{\mathbb{E}}_{\mathbf{\Delta} \sim \mathcal{N}(0,\sigma^2 I_n)}[\|\mathbf{\Delta}\|_2^2]} \\ &= mn\sigma.\end{aligned}$$

Therefore, the contribution of this case is upper bounded as

$$\Pr_{\mathbf{x},\mathbf{\Delta}}\left[\mathsf{case(B)} \wedge \exists j \in [n]\ 1/2 \in [\mathbf{x}_j, \mathbf{x}_j + \Delta_j]\right] \leqslant \Pr_{\mathbf{x},\mathbf{\Delta}}[\mathsf{case(B)}] mn\sigma = n^{-\omega(1)} \cdot \sigma = O(\varepsilon).$$

4.4.2 Analysis of case (A)

We now analyze the contribution of x's that fall into case (A) to the left hand side of (2).

4.4.2.1 Case (A), Condition 2

If $\mathbf{x}$ falls under Case (A), then the distance from all $\mathbf{x}_i$'s to $z = S(\mathbf{x})$ is at least $\frac{\log n}{100n}$. Therefore, Condition 2 holds as long as $|\mathbf{\Delta}_i| < \frac{\log n}{100n}$ for all i. Since for each i we have that

$$\Pr_{\mathbf{\Delta} \sim \mathcal{N}(0,\sigma^2 I_n)}\left[|\mathbf{\Delta}_i| \geqslant \frac{\log n}{100n}\right] = \Pr_{\mathbf{\Delta} \sim \mathcal{N}(0,\sigma^2 I_n)}\left[\mathbf{\Delta}_i^2 \geqslant \frac{\log^2 n}{100^2 n^2}\right] \lesssim \frac{\sigma^2}{\log^2 n/n^2} \lesssim \varepsilon^2/\log n,$$

we get by the union bound that

$$\Pr_{\mathbf{\Delta} \sim \mathcal{N}(0,\sigma^2 I_n)}\left[\exists i\ \ |\mathbf{\Delta}_i| \geqslant \frac{\log n}{100n}\right] \lesssim n\varepsilon^2/\log n \lesssim \varepsilon,$$

for a sufficiently small ε.

4.5 Case (A), Condition 1

This is the main part of the proof. We show that in case (A), the probability that $S_{\vec{r}}(\mathbf{x}) \neq S_{\vec{r}}(\mathbf{x}+\boldsymbol{\Delta})$ is at most $O(\varepsilon)$. Note that the procedure describing $S_{\vec{r}}$ in this case is the correlated sampling procedure of Holenstein [18], where $S_{\vec{r}}(x)$ samples i according to the distribution $p(x) = (p_1(x), \dots, p_m(x))$ and $S_{\vec{r}}(x+\Delta)$ samples i according to the distribution $p(x+\Delta)$. Therefore, the probability they sample different i's is at most the statistical distance between the distributions, $\|p(x) - p(x+\Delta)\|_1$. Therefore, we must show that

$$\mathbb{E}_{\mathbf{x},\mathbf{x}+\boldsymbol{\Delta}}\left[\|p(\mathbf{x}) - p(\mathbf{x}+\boldsymbol{\Delta})\|_1 \;\middle|\; \mathsf{case(A)}\right] = O(\varepsilon). \tag{4}$$

Before we turn to this task, we upper bound the contribution from several rare cases.

4.5.1 Contribution from some rare cases

First, we show that the case some $\boldsymbol{\Delta}_i$ is too large contributes at most $O(\varepsilon)$ to the LHS of (4).

▷ Claim 27. $\Pr_{\boldsymbol{\Delta}\sim\mathcal{N}(0,\sigma^2 I_n)}\left[|\boldsymbol{\Delta}_i| \geqslant \varepsilon^{0.95}/n \text{ for some } i\right] \leqslant \varepsilon$.

Proof. For each i, we have that

$$\Pr_{\boldsymbol{\Delta}\sim\mathcal{N}(0,\sigma^2 I_n)}\left[|\boldsymbol{\Delta}_i| \geqslant \varepsilon^{0.95}/n\right] \leqslant 2^{-\Omega((\varepsilon^{0.95}/n)^2/\sigma^2)} = 2^{-\Omega\left(\frac{1}{\varepsilon^{0.1}\log n}\right)} \leqslant \frac{\varepsilon}{n},$$

for small enough ε, and the claim follows from the union bound. ◁

From now on, we assume that the $\boldsymbol{\Delta}_i$'s are distributed from $N(0,\sigma^2)|_{|\boldsymbol{\Delta}_i|<\varepsilon^{0.95}/n}$. In particular, we can assume that if $\mathbf{t}_j$ is the number of $\boldsymbol{x}$'s that fall into interval I_j, these numbers stay the same under $\mathbf{x}+\boldsymbol{\Delta}$. [2] Next, we handle the case in which $p(\mathbf{x})$ is supported only on a single j. Note that in this case, if $p(\mathbf{x}+\boldsymbol{\Delta})$ is also only supported on this single j, then the contribution of these cases to the LHS of (4) is 0. We show that the contribution from the other case is $O(\varepsilon)$.

▷ Claim 28.

$$\Pr_{\mathbf{x},\boldsymbol{\Delta}}\left[\exists j^\star \text{ such that } p(\mathbf{x}) \text{ is only supported on } j^\star\text{, the support of } p(\mathbf{x}+\boldsymbol{\Delta}) \text{ is different}\right]$$
$$\lesssim \varepsilon.$$

Proof. In case (B), we have shown that the probability that $r_j(\mathbf{x}) = 0$ for all j is $n^{-\omega(1)}$, and the same argument shows that the probability $r_j(\mathbf{x}) = 0$ for all but a single $j^\star$ is still $n^{-\omega(1)}$. Denote this event by E.

Let us condition on the event E, on $j^\star$ and the number $\mathbf{t}_1, \dots, \mathbf{t}_m$ of $\mathbf{x}_i$'s that fall into $I_1, \dots, I_m$. Note that for each $j \neq j^\star$, since $r_j(\mathbf{x}) = 0$ there is i such that $\mathbf{x}_i \in J_j \overset{def}{=} [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]$, and we condition on that i_j for each j (if there is more than one, we choose one arbitrarily). Note that the distribution of $\mathbf{x}_{i_j}$ is thus uniform over J_j.

[2] Strictly speaking, $x_i + \Delta_i$ may be in a different interval than x_i, but in this case it doesn't affects the distribution $p(x)$. Indeed, suppose x_i is in I_j but $x_i + \Delta_i$ is in I_{j+1}. Then $|x_i + \Delta_i - z_{j+1}| \geqslant |z_{j+1} - j/m| - |\Delta_i| - |x_i - j/m| \geqslant 1/m - 2\varepsilon^{0.95}/n \geqslant 1/m - \varepsilon^{0.95}$. Therefore, $\frac{50n}{\log n}|x_i + \Delta_i - z_{j+1}| > 2$, and so $f(\frac{50n}{\log n}|x_i + \Delta_i - z_{j+1}|) = 1$.

Now note that if for each $j \neq j^\star$ it holds that $\mathbf{x}_{i_j} + \mathbf{\Delta}_{i_j} \in J_j$, then $r_j(\mathbf{x}+\mathbf{\Delta}) = 0$, so the only contribution to the probability of the event in question comes when $\mathbf{x}_{i_j} + \mathbf{\Delta}_{i_j} \notin J_j$ (or from case (B), which we have already accounted for earlier). Conditioned on $\mathbf{\Delta} = \Delta$, the probability for that is at most

$$\mathop{\mathbb{E}}_{(\mathbf{x}_{i_j})_{j\neq j^\star}}\left[\sum_{j\neq j^\star} 1_{\mathbf{x}_{i_j}+\Delta_{i_j}\notin J_j}\right] = \sum_{j\neq j^\star} \mathop{\mathbb{E}}_{\mathbf{x}_j}\left[1_{\mathbf{x}_{i_j}+\Delta_{i_j}\notin J_j}\right] \leqslant \sum_{j\neq j^\star} \frac{|\Delta_j|}{\log n/(50n)},$$

therefore taking expectation over Δ and using Cauchy-Schwarz we get that

$$\mathop{\mathbb{E}}_{\mathbf{\Delta},(\mathbf{x}_{i_j})_{j\neq j^\star}}\left[\sum_{j\neq j^\star} 1_{\mathbf{x}_{i_j}+\mathbf{\Delta}_{i_j}\notin J_j}\right] \lesssim \frac{n}{\log n}\sqrt{m}\sqrt{\sum_{j\neq j^\star}\mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\left|\mathbf{\Delta}_j\right|^2\right]} \leqslant \frac{n}{\log n}\sqrt{m}\sqrt{m\sigma^2} \leqslant n^2\sigma.$$

Therefore, we get that

$$\begin{aligned}
&\Pr_{\mathbf{x},\mathbf{\Delta}}\left[p(\mathbf{x}) \text{ is only supported on } j^\star\text{, but the support of } p(\mathbf{x}+\mathbf{\Delta}) \text{ is different}\right]\\
&\qquad\leqslant \Pr\left[E\right] n^2\sigma\\
&\qquad\leqslant n^{-\omega(1)} n^2\sigma\\
&\qquad\lesssim \varepsilon.
\end{aligned}$$

◁

Let E be the event that the support of $p(\mathbf{x})$ consists of at least two distinct j's. We condition on the event E in the subsequent argument. The following claim shows that conditioned on E, the sum of the $r_j(\mathbf{x})$'s is at least somewhat bounded away from 0. It will only come into play later in the proof.

▷ Claim 29. $\Pr_{\mathbf{x}}\left[\sum_j r_j(\mathbf{x}) \leqslant \varepsilon^{1.6} \,\middle|\, E\right] \lesssim \varepsilon.$

Proof. Since we conditioned on E, there are $j_1 \neq j_2$'s such that $r_{j_1}(\mathbf{x}), r_{j_2}(\mathbf{x}) > 0$. We condition on j_1 and j_2, and assume without loss of generality that $j_1 = 1, j_2 = 2$. We show that

$$\Pr_{\mathbf{x}}\left[r_1(\mathbf{x}) < \varepsilon^{1.6} \wedge r_2(\mathbf{x}) < \varepsilon^{1.6} \,\middle|\, r_1(\mathbf{x}), r_2(\mathbf{x}) > 0\right] \lesssim \varepsilon, \tag{5}$$

and thus the result would follow.

Let $\mathbf{t}_1$ be the number of i's such that $\mathbf{x}_i \in I_1$, and $\mathbf{t}_2$ be the number of i's such that $\mathbf{x}_i \in I_2$. Note that $\mathbf{t}_1, \mathbf{t}_2 \leqslant n$. In addition, conditioned on $\mathbf{t}_1 = t_1$ and $\mathbf{t}_2 = t_2$, the events $r_1(\mathbf{x}) < \varepsilon^{1.6}$ and $r_2(x) < \varepsilon^{1.6}$ become independent. Therefore, to prove (5), it suffices to show for all $t_1 \leqslant n, t_2 \leqslant n$,

$$\Pr_{\mathbf{x}}\left[r_1(\mathbf{x}) < \varepsilon^{1.6} \,\middle|\, r_1(\mathbf{x}) > 0,\ \mathbf{t}_1 = t_1,\ \mathbf{t}_2 = t_2\right] \lesssim \varepsilon^{0.5}. \tag{6}$$

Note that one way to sample $r_1(\mathbf{x}) | r_1(\mathbf{x}) > 0,\ \mathbf{t}_1 = t_1$ is as follows.

- Sample points $\mathbf{x}_1, \ldots, \mathbf{x}_{t_1}$ uniformly from I_1 conditioned on $|\mathbf{x}_i - z_1| > \frac{\log n}{50n}$;
- $r_1(\mathbf{x}) = \prod_{i=1}^{t_1} g_1(\mathbf{x}_i)$.

Let $\mathbf{Y}_i$ be the random variable $\mathbf{Y}_i := g_1(\mathbf{x}_i)^{-0.32}$, where $\mathbf{x}_i$ is sampled as above (we need $0.32 < 1/3$). Let E be the event that $|\mathbf{x}_i - z_1| \geqslant \frac{\log n}{25n}$. If E holds, then we get that $g_1(\mathbf{x}_i) = 1$, and otherwise $g_1(\mathbf{x}_i) \gtrsim \left|\frac{50n}{\log n}|\mathbf{x}_i - z_1| - 1\right|^3$, so

$$\mathbb{E}\left[\mathbf{Y}_i\right] \leqslant \Pr\left[E\right]\cdot 1 + \Pr\left[\bar{E}\right]\mathbb{E}\left[g_1(\mathbf{x}_i)^{-0.32} \,\middle|\, \bar{E}\right] \lesssim 1 + \mathbb{E}\left[\left|\frac{50n}{\log n}|\mathbf{x}_i - z_1| - 1\right|^{-0.96} \,\middle|\, \bar{E}\right].$$

We write the last expectation as an integral, noting that $|\mathbf{x}_i - z_1|$ is distributed uniformly on $\left[\frac{\log n}{50n}, \frac{\log n}{25n}\right]$, hence

$$\mathbb{E}\left[\left|\frac{50n}{\log n}|\mathbf{x}_i - z_1| - 1\right|^{-0.96}\,\middle|\,\bar{E}\right] \lesssim \frac{n}{\log n}\int_{\frac{\log n}{50n}}^{\frac{\log n}{25n}} \left|\frac{50n}{\log n}t - 1\right|^{-0.96} dt = \frac{1}{50}\int_0^1 y^{-0.96} dt \lesssim 1,$$

where we made the change of variables $y = \frac{50n}{\log n}t - 1$. Thus, $\mathbb{E}[\mathbf{Y}_i] \lesssim 1$, and so there is a constant B such that $\mathbb{E}[\mathbf{Y}_i] \leqslant B$. Therefore by independence $\mathbb{E}\left[\prod_{i=1}^{t_1} \mathbf{Y}_i\right] \leqslant B^{t_1} \leqslant B^n$, and so writing $r_1(\mathbf{x})$ in terms of the $\mathbf{Y}_i$'s and using Markov's inequality we get that

$$\begin{aligned}\Pr_{\mathbf{x}}\left[r_1(\mathbf{x}) < \varepsilon^{1.6} \,\middle|\, r_1(\mathbf{x}) > 0,\ \mathbf{t}_1 = t_1,\ \mathbf{t}_2 = t_2\right] &= \Pr\left[\prod_{i=1}^{t_1} \mathbf{Y}_i > \varepsilon^{-1.6\times 0.32}\right] \\ &\leqslant B^n \cdot \varepsilon^{0.512} \\ &\lesssim \varepsilon^{0.5}.\end{aligned}$$

◁

4.5.2 Analyzing the typical case

To expand out $\|p(\mathbf{x}) - p(\mathbf{x} + \mathbf{\Delta})\|_1$, we will be using the following claim. The set-up one should have in mind is that $r_j = r_j(x)$ and $d_j = r_j(x + \Delta)$ for some x and Δ that are typical enough.

▷ Claim 30. Let $r_j \geqslant 0$, d_j be real-numbers satisfying $|d_j| \leqslant r_j/2$ for all j. Denote $T = \sum r_j$, $T' = \sum(r_j + d_j)$, and let $p_j = r_j/T$ and $q_j = (r_i + d_i)/T'$ be two distributions. Then

$$\|p - q\|_1 \lesssim \sum_i \frac{|d_i|}{r_i} \cdot \frac{\min(r_i, T - r_i)}{T}. \tag{7}$$

We defer the proof of Claim 30 to Section A. Morally speaking, it says that

$$\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{\Delta}}\left[\|p(\mathbf{x}) - p(\mathbf{x} + \mathbf{\Delta})\|_1\right] \lesssim \sum_{j=1}^{m} \mathop{\mathbb{E}}_{\mathbf{x}}\left[\mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\frac{|r_j(\mathbf{x}) - r_j(\mathbf{x} + \mathbf{\Delta})|}{r_j(\mathbf{x})} \cdot \frac{\min(r_j(\mathbf{x}), T(\mathbf{x}) - r_j(\mathbf{x}))}{T(\mathbf{x})}\right]\right], \tag{8}$$

where $T(x) = \sum\limits_j r_j(x)$ (this is only morally because we are assuming that the supports of $p_j(x)$ and $p_j(x + \Delta)$ are the same, but formally speaking they may be different). In particular, to be able to handle with that we first must understand the expectation of $|r_j(x) - r_j(x + \mathbf{\Delta})|$ over $\mathbf{\Delta}$.

▷ Claim 31. Let $j \in [m]$, $x_1, \ldots, x_k \in [z_j - \frac{\log n}{25n} - \varepsilon^{0.95}, z_j + \frac{\log n}{25n} + \varepsilon^{0.95}] \setminus [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]$, and let $r(x) = \prod_{i=1}^{c} g_j(x_i)$. Denote $\alpha_i = \operatorname{dist}\left(x_i, [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]\right)$. and let $\mathbf{\Delta}_i \sim N(0, \sigma^2)|_{|\Delta_i| < \varepsilon^{0.95}}$. Then

$$\mathop{\mathbb{E}}_{\mathbf{\Delta}}[|r(x + \mathbf{\Delta}) - r(x)|] \lesssim \max\left(\varepsilon^{2.65}, r(x) \cdot \sigma \cdot \sqrt{\sum_{i=1}^{c} \frac{1}{\alpha_i^2}}\right). \tag{9}$$

Proof. We consider two cases.

4.5.2.1 Case 1: $\alpha_i \leqslant \varepsilon^{0.9}$ for some i

In this case, we have

$$g_j(x_i) \lesssim \left(\frac{50n}{\log n}\alpha_i\right)^3 \lesssim n^3\varepsilon^{3\cdot 0.9} \lesssim \varepsilon^{2.66}.$$

Similarly, we have $\mathsf{dist}(x_i+\mathbf{\Delta}_i, [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50}]) \leqslant \alpha_i + |\mathbf{\Delta}_i| \leqslant 2\alpha_i$, so $g_j(x_i+\mathbf{\Delta}_i) \lesssim \varepsilon^{2.66}$. We conclude that $r(x_i), r(x_i+\mathbf{\Delta}_i) \lesssim \varepsilon^{2.66}$, hence the contribution from these cases is at most $\varepsilon^{2.65}$.

4.5.2.2 Case 2: $\alpha_i > \varepsilon^{0.9}$ for all i

In this case, we get that $x_i + \mathbf{\Delta}_i$ is also not in the interval $[z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]$, hence $g_j(x_i + \mathbf{\Delta}_i) \neq 0$, so $r(x+\mathbf{\Delta}) > 0$. Since $r(x)$ are defined using products, it would be more convenient for us to analyze $\log(r(x+\mathbf{\Delta})/r(x))$ as opposed to $r(x+\mathbf{\Delta})/r(x) - 1$, and to justify we can do that we first argue that $r(x+\mathbf{\Delta})/r(x) = 1 + o(1)$.

To see that, note that as $|\mathbf{\Delta}_i| \leqslant \varepsilon^{0.95} \leqslant \alpha_i/2$, we may use Fact 26 to conclude that

$$|g(x_i+\mathbf{\Delta}_i) - g(x_i)| \lesssim \frac{|\mathbf{\Delta}_i|}{\alpha_i}|g(x_i)| \lesssim \varepsilon^{0.05}|g(x_i)|.$$

In particular, we get that $\frac{g_j(x_i+\mathbf{\Delta}_i)}{g_j(x_i)} = 1 \pm O(\varepsilon^{0.05})$, and hence $\frac{r(x+\mathbf{\Delta})}{r(x)} = 1 \pm O(k\varepsilon^{0.05})$. Writing $\frac{r(x+\mathbf{\Delta})}{r(x)} = 1 + \boldsymbol{\eta}$, we get $\boldsymbol{\eta}$ is small in absolute value, and hence $|\log(r(x+\mathbf{\Delta})/r(x))| \gtrsim |\boldsymbol{\eta}| \gtrsim \left|\frac{r(x+\mathbf{\Delta})}{r(x)} - 1\right| = \left|\frac{r(x+\mathbf{\Delta}) - r(x)}{r(x)}\right|$. I.e.,

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\frac{|r(x+\mathbf{\Delta}) - r(x)|}{r(x)}\right] \lesssim \mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\left|\log\left(\frac{r(x+\mathbf{\Delta})}{r(x)}\right)\right|\right] &= \mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\left|\sum_{i=1}^{k}\log\frac{g_j(x_i+\mathbf{\Delta}_i)}{g_j(x_i)}\right|\right] \\ &= \mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\left|\sum_{i=1}^{k}\mathbf{Y}_i\right|\right], \end{aligned} \tag{10}$$

where we define the random variables $\mathbf{Y}_i = \log\frac{g_j(x_i+\mathbf{\Delta}_i)}{g_j(x_i)}$.

Observe that $\mathbf{Y}_i$'s are mutually independent, since each $\mathbf{Y}_i$ only depends on the corresponding $\mathbf{\Delta}_i$. We wish to upper bound the average and variance of $\mathbf{Y}_i$, and to do that it would be more convenient to analyze $\mathbf{Z}_i = \frac{g_j(x_i+\mathbf{\Delta}_i) - g_j(x_i)}{g_j(x_i)}$ and then relate the two.

Using second order Taylor's approximation, we have that there is $\mathbf{y}_i \in [x_i, x_i + \mathbf{\Delta}_i]$ such that

$$g_j(x_i+\mathbf{\Delta}_i) = g_j(x_i) + g_j'(x_i)\mathbf{\Delta}_i + \frac{1}{2}g_j''(\mathbf{y}_i)\mathbf{\Delta}_i^2,$$

hence

$$\left|\mathop{\mathbb{E}}_{\mathbf{\Delta}}[\mathbf{Z}_i]\right| = \frac{1}{g_j(x_i)}\left|\mathop{\mathbb{E}}_{\Delta}\left[g_j'(x_i)\mathbf{\Delta}_i + \frac{1}{2}g_j''(\mathbf{y}_i)\mathbf{\Delta}_i^2\right]\right| = \frac{1}{2g_j(x_i)}\left|\mathop{\mathbb{E}}_{\Delta}\left[g_j''(\mathbf{y}_i)\mathbf{\Delta}_i^2\right]\right|. \tag{11}$$

Using properties of f, we have

$$|g_j''(\mathbf{y}_i)| = \left(\frac{50n}{\log n}\right)^2 f''\left(\frac{50n}{\log n}|\mathbf{y}_i - z_j|\right) \lesssim \left(\frac{50n}{\log n}\right)^2\left|\frac{50n}{\log n}|\mathbf{y}_i - z_j| - 1\right|.$$

Since $\mathbf{y}_i \in [x_i, x_i + \mathbf{\Delta}_i]$, we get that $\left(\frac{50n}{\log n}\left|\mathbf{y}_i - z_j\right| - 1\right) \geqslant \frac{50n}{\log n}\alpha_i - \varepsilon^{0.95} \geqslant \frac{25n}{\log n}\alpha_i$, and so we may continue the previous inequality as

$$|g''(\mathbf{y}_i)| \lesssim \left(\frac{50n}{\log n}\right)^2 \frac{\left|\frac{50n}{\log n}\left|\mathbf{y}_i - z_j\right| - 1\right|^3}{(\frac{25n}{\log n}\alpha_i)^2} \lesssim \frac{1}{\alpha_i^2}\left|g_j(\mathbf{y}_i)\right| \lesssim \frac{1}{\alpha_i^2}\left|g_j(x_i)\right|,$$

where the last inequality is by Fact 26. Plugging this into (11) we get that

$$\left|\mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\mathbf{Z}_i\right]\right| \lesssim \frac{1}{\alpha_i^2}\mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\mathbf{\Delta}_i^2\right] = \frac{1}{\alpha_i^2}\sigma^2.$$

In a similar fashion, we upper bound the second moment of $\mathbf{Z}_i$. Using Fact 26, we get that $|\mathbf{Z}_i| \leqslant \frac{\mathbf{\Delta}_i}{\alpha_i}$, and so $\mathbb{E}_{\mathbf{\Delta}}\left[\mathbf{Z}_i^2\right] \lesssim \frac{1}{\alpha_i^2}\mathbb{E}_{\mathbf{\Delta}}\left[\mathbf{\Delta}_i^2\right] = \frac{1}{\alpha_i^2}\sigma^2$.

We can now upper bound the average of $\mathbf{Y}_i$ as follows. Recall that, $|\mathbf{Z}_i| = o(1)$ so by Taylor's approximation $\mathbf{Y}_i = \log(1+\mathbf{Z}_i) = \mathbf{Z}_i - \frac{1}{2(1+\boldsymbol{\xi}_i)^2}\mathbf{Z}_i^2$ for some $\boldsymbol{\xi}_i \in [1, 1+\mathbf{Z}_i]$ and hence

$$\left|\mathbb{E}\left[\mathbf{Y}_i\right]\right| \lesssim |\mathbb{E}[\mathbf{Z}_i]| + \left|\mathbb{E}[\mathbf{Z}_i^2]\right| \lesssim \frac{1}{\alpha_i^2}\sigma^2. \tag{12}$$

This approximation (along with the fact that $|\mathbf{Z}_i| = o(1)$) also implies $|\mathbf{Y}_i| \lesssim |\mathbf{Z}_i|$, hence

$$\mathbb{E}[\mathbf{Y}_i^2] \lesssim \mathbb{E}[\mathbf{Z}_i^2] \lesssim \frac{1}{\alpha_i^2}\sigma^2. \tag{13}$$

We can now continue equation (10) to upper bound the LHS there. Denoting $\mu_i := \mathbb{E}[\mathbf{Y}_i]$, we have

$$\mathop{\mathbb{E}}_{\Delta}\left[\left|\sum_{i=1}^{k}\mathbf{Y}_i\right|\right] \leqslant \sum_{i=1}^{k}|\mu_i| + \mathop{\mathbb{E}}_{\Delta}\left[\left|\sum_{i=1}^{k}\mathbf{Y}_i - \mu_i\right|\right] \leqslant \sum_{i=1}^{k}|\mu_i| + \sqrt{\mathop{\mathbb{E}}_{\Delta}\left[\sum_{i=1}^{k}(\mathbf{Y}_i - \mu_i)^2\right]},$$

where in the last inequality we used Cauchy-Schwarz and the fact that Y_i's are independent. Using (12) we have that $\sum_{i=1}^{k}|\mu_i| \lesssim \sigma^2\sum_{i=1}^{k}\frac{1}{\alpha_i^2}$, and to upper bound the second term we use (13):

$$\mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[(\mathbf{Y}_i - \mu_i)^2\right] \leqslant \mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\mathbf{Y}_i^2\right] \lesssim \frac{1}{\alpha_i^2}\sigma^2.$$

Together, we get that

$$\mathop{\mathbb{E}}_{\mathbf{\Delta}}\left[\left|\sum_{i=1}^{k}\mathbf{Y}_i\right|\right] \lesssim \sigma^2\sum_{i=1}^{k}\frac{1}{\alpha_i^2} + \sqrt{\sigma^2\sum_{i=1}^{k}\frac{1}{\alpha_i^2}} \lesssim \sigma\sqrt{\sum_{i=1}^{k}\frac{1}{\alpha_i^2}},$$

where the last inequality holds since $\sigma^2\sum_{i=1}^{k}\frac{1}{\alpha_i^2} \lesssim 1$ (as $\sigma^2 \lesssim \varepsilon^2$ and $\alpha_i \geqslant \varepsilon^{0.9}$). ◁

Next, using the previous claim we upper bound the expectation of each summand on the RHS of (8). The following statement addresses a single term, and should be thought of as being applied after conditioning on x, Δ being not-too untypical, and focusing only on x_i's for which there is a chance that $g_j(x_i + \Delta_i) \neq g_j(x_i)$.

▷ Claim 32. Let $j \in [m]$, $k \leqslant n$, $S \geqslant 0$ and let $x_1, \ldots, x_k$ be chosen uniformly at random from $[z_j - \frac{\log n}{25n} - \varepsilon^{0.95}, z_j + \frac{\log n}{25n} + \varepsilon^{0.95}] \setminus [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]$. Let $\Delta_i \sim N(0, \sigma^2)|_{|\Delta_i| < \varepsilon^{0.95}}$. Then

$$\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{\Delta}}\left[\frac{|r_j(\mathbf{x}+\mathbf{\Delta}) - r_j(\mathbf{x})|}{r_j(\mathbf{x})} \cdot \frac{\min(r_j(\mathbf{x}), S)}{r_j(\mathbf{x}) + S + \varepsilon^{1.6}}\right] \lesssim \varepsilon^{1.05} + k\frac{\sigma n}{\log n} \cdot \Pr_{\mathbf{x}}[r_j(\mathbf{x}) \geqslant S]$$
$$+\sigma\frac{n}{\log n}\sqrt{k}\mathop{\mathbb{E}}_{\mathbf{x}}\left[\frac{r_j(\mathbf{x})}{r_j(\mathbf{x}) + S}\right].$$

Proof. Upper bounding $\max(a,b) \leqslant a + b$ for $a, b \geqslant 0$, by Claim 31, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{\Delta}}\left[\frac{|r_j(\mathbf{x}+\mathbf{\Delta}) - r_j(\mathbf{x})|}{r_j(\mathbf{x})} \cdot \frac{\min(r_j(\mathbf{x}), S)}{r_j(\mathbf{x}) + S + \varepsilon^{1.6}}\right] \\
&\quad\lesssim \mathop{\mathbb{E}}_{\mathbf{x}}\left[\frac{\varepsilon^{2.65} + r_j(\mathbf{x}) \cdot \sigma \cdot \sqrt{\sum_{i=1}^{k}\frac{1}{\boldsymbol{\alpha}_i^2}}}{r_j(\mathbf{x})} \cdot \frac{\min(r_j(\mathbf{x}), S)}{r_j(\mathbf{x}) + S + \varepsilon^{1.6}}\right] \\
&\quad\lesssim \varepsilon^{1.05} + \sigma\mathop{\mathbb{E}}_{\mathbf{x}}\left[\sqrt{\sum_{i=1}^{k}\frac{1}{\boldsymbol{\alpha}_i^2}} \cdot \frac{\min(r_j(\mathbf{x}), S)}{r_j(\mathbf{x}) + S + \varepsilon^{1.6}}\right],
\end{aligned}$$

and it is enough to bound the second term. Note that while we expect that each $\boldsymbol{\alpha}_i$ to be of the order $\log n/n$, convexity works against us and it could still be the case that $\sum\limits_{i=1}^{k}\frac{1}{\boldsymbol{\alpha}_i^2}$ could be large. The point is that in this case, some $\boldsymbol{\alpha}_i$ must be close to 0, in which case $g_j(\mathbf{x}_i)$ is very small – cubically with $\boldsymbol{\alpha}_i$ – thereby balancing the $1/\boldsymbol{\alpha}_i^2$ term. The following proposition formalizes this intuition, and the proof is deferred to Section A

▶ **Proposition 33.** *There is an absolute constant $A > 0$ such that for any $z > 0$ and $r \leqslant 1$ such that $r_j(x) = r \cdot g_j(x_i)$, it holds that*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\sqrt{z + \frac{1}{\boldsymbol{\alpha}_i^2}} \cdot \frac{\min(r \cdot g_j(\mathbf{x}_i), S)}{r \cdot g_j(\mathbf{x}_i) + S + \varepsilon^{1.6}}\right] \\
&\leqslant \mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\sqrt{z + A\frac{n^2}{\log^2 n}} \cdot \frac{\min(r \cdot g_j(\mathbf{x}_i), S)}{r \cdot g_j(\mathbf{x}_i) + S + \varepsilon^{1.6}} + A\frac{n}{\log n} \cdot \mathbb{1}_{r \cdot g_j(\mathbf{x}_i) \geqslant S}\right].
\end{aligned}$$

Applying Proposition 33 iteratively k times (once for each i, taking $r = \prod\limits_{i' \neq i} g_j(x_{i'})$ and the appropriate z), we get that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{x}}\left[\sqrt{\sum_{i=1}^{k}\frac{1}{\boldsymbol{\alpha}_i^2}} \cdot \frac{\min(r_j(\mathbf{x}), S)}{r_j(\mathbf{x}) + S + \varepsilon^{1.6}}\right] \\
&\leqslant \mathop{\mathbb{E}}_{\mathbf{x}}\left[\sqrt{k \cdot A\frac{n^2}{\log^2 n}} \cdot \frac{\min(r_j(\mathbf{x}), S)}{r_j(\mathbf{x}) + S + \varepsilon^{1.6}} + k \cdot A\frac{n}{\log n} \cdot \mathbb{1}_{r_j(\mathbf{x}) \geqslant S}\right].
\end{aligned}$$

The proof is concluded by noting that $\frac{\min(r_j(x), S)}{r_j(x) + S + \varepsilon^{1.6}} \leqslant \frac{r_j(x)}{r_j(x) + S}$. ◁

We are now ready to finish the proof of inequality (4).

Proof of inequality (4)

Let E be the event that: (1) the support of $p(\mathbf{x})$ has size at least 2, (2) $\sum_j r_j(\mathbf{x}) \geqslant \varepsilon^{1.6}$ and also for $\mathbf{x}+\boldsymbol{\Delta}$, and (3) $|\boldsymbol{\Delta}_i| \leqslant \varepsilon^{0.95}$ for all $i \in [n]$. As we argued in Claims 27, 28, 29 the contribution $(\mathbf{x},\boldsymbol{\Delta}) \notin E$ to the LHS of inequality (4) is $\lesssim \varepsilon$, hence it is enough to analyze the contribution of $(\mathbf{x},\boldsymbol{\Delta}) \in E$.

Denote $T(x) = \sum_{j\in[m]} r_j(x)$.

$$\mathop{\mathbb{E}}_{\mathbf{x},\boldsymbol{\Delta}}\left[\|p(\mathbf{x}) - p(\mathbf{x}+\boldsymbol{\Delta})\|_1 1_E\right] = \mathop{\mathbb{E}}_{\mathbf{x},\boldsymbol{\Delta}}\left[\sum_{j\in[m]}\left|\frac{r_j(\mathbf{x})}{T(\mathbf{x})} - \frac{r_j(\mathbf{x}+\boldsymbol{\Delta})}{T(\mathbf{x}+\boldsymbol{\Delta})}\right| 1_E\right]$$

$$= \underbrace{\mathop{\mathbb{E}}_{\mathbf{x},\boldsymbol{\Delta}}\left[\sum_{j\in[m]}\left|\frac{r_j(\mathbf{x})}{T(\mathbf{x})} - \frac{r_j(\mathbf{x}+\boldsymbol{\Delta})}{T(\mathbf{x}+\boldsymbol{\Delta})}\right| 1_E 1_{r_j(\mathbf{x})\leqslant\varepsilon^{2.7}}\right]}_{(I)}$$

$$+ \underbrace{\mathop{\mathbb{E}}_{\mathbf{x},\boldsymbol{\Delta}}\left[\sum_{j\in[m]}\left|\frac{r_j(\mathbf{x})}{T(\mathbf{x})} - \frac{r_j(\mathbf{x}+\boldsymbol{\Delta})}{T(\mathbf{x}+\boldsymbol{\Delta})}\right| 1_E 1_{r_j(\mathbf{x})>\varepsilon^{2.7}}\right]}_{(II)}.$$

First, we show that $(I) \lesssim \varepsilon$. As $T(\mathbf{x}) \geqslant \varepsilon^{1.6}$ (since E holds) and $r_j(\mathbf{x}) \leqslant \varepsilon^{2.7}$, we get that $r_j(\mathbf{x})/T(\mathbf{x}) \leqslant \varepsilon^{1.1}$, and next we argue that $r_j(\mathbf{x}+\boldsymbol{\Delta})/T(\mathbf{x}+\boldsymbol{\Delta}) \lesssim \varepsilon^{1.05}$. Fix j and suppose $\mathbf{x}_1,\dots,\mathbf{x}_{k_j}$ are the $\mathbf{x}_i$'s that fall inside I_j. The following easy fact will be helpful.

▶ **Fact 34.** *For all x,Δ we have $r_j(x+\Delta) = \sum_{S\subseteq[k_j]} \prod_{r\in S} g_j(x_r) \prod_{r\notin S} (g_j(x_r) - g_j(x_r+\Delta_r))$.*

Proof. Write $r_j(x+\Delta) = \prod_{r=1}^{k_j} g_j(x_r+\Delta_r) = \prod_{r=1}^{k_j} (g_j(x_r) + (g_j(x_r+\Delta_r) - g_j(x_r)))$ and expand out. ◀

Combining Fact 34 and Fact 26, we get that

$$\begin{aligned} r_j(x+\Delta) &\leqslant \sum_{S\subseteq[k_j]} \prod_{r\in S} g_j(x_r) \prod_{r\notin S} |g_j(x_r) - g_j(x_r+\Delta_r)| \\ &\leqslant \sum_{S\subseteq[k_j]} \prod_{r\in S} g_j(x_r) B^{|S|} n^{3|S|} \prod_{r\notin S} (\alpha_r^3 + |\Delta_r|^3) \\ &\leqslant \sum_{S\subseteq[k_j]} \prod_{r\in S} g_j(x_r) B^{|S|} n^{3|S|} \prod_{r\notin S} \alpha_r^3 \\ &\quad + 4^n B^{|n|} n^{3n} \max_r |\Delta_r|^3 . \end{aligned}$$

Consider the right hand side above. For the first term we use $\alpha_r^3 \lesssim g_j(x_r)$ to get it is at most

$$\sum_{S\subseteq[k_j]} B'^{|S|} n^{3|S|} r_j(x_r) \leqslant (B'')^n n^{3n} \varepsilon^{2.7} \leqslant \varepsilon^{2.65}/2.$$

For the second term we use $|\Delta_r| \leqslant \varepsilon^{0.95}$ to bound it by $\varepsilon^{2.65}/2$ as well. We thus get $r_j(x+\Delta) \leqslant \varepsilon^{2.65}$, and so $r_j(x+\Delta)/T(x+\Delta) \leqslant \varepsilon^{1.05}$. Combined, we get that

$$(I) \leqslant m(\varepsilon^{1.1} + \varepsilon^{1.05}) \lesssim \varepsilon.$$

Next, we handle (II). Denote $T'(x) = \sum_j r_j(x) 1_{r_j(x)\geqslant \varepsilon^{2.7}}$, and note that $T'(x) \geqslant T(x) - m\varepsilon^{2.7} \geqslant (1-m\varepsilon^{1.1})T(x)$ and similarly for $T'(x+\Delta)$. Thus, we may replace $T(x), T(x+\Delta)$ with $T'(x), T'(x+\Delta)$ and incur (by the triangle inequality) a loss of at most $m\varepsilon^{1.1} \lesssim \varepsilon$. Thus, we want to upper bound

$$\underbrace{\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{\Delta}}\left[\sum_{j\in[m]}\left|\frac{r_j(\mathbf{x})}{T'(\mathbf{x})} - \frac{r_j(\mathbf{x}+\mathbf{\Delta})}{T'(\mathbf{x}+\mathbf{\Delta})}\right| 1_E 1_{r_j(\mathbf{x})>\varepsilon^{2.7}}\right]}_{(III)} \lesssim \varepsilon.$$

We intend to apply Claim 30 with $r_j = r_j(x)$ and $d_j = r_j(x+\Delta) - r_j(x)$ for each x separately, but for that we first have to argue that $|d_j| \leqslant r_j/2$. For each $i \in [n]$ there is j such that $x_i \in I_j$, and we denote $\alpha_i = \text{dist}\left(x_i, [z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}]\right)$. Note that

$$\varepsilon^{2.7} \leqslant r_j(x) \leqslant g_j(x_i) \lesssim \left(\frac{n}{\log n}\alpha_i\right)^3,$$

hence $\alpha_i \gtrsim \frac{\log n}{n}\varepsilon^{0.9}$, and for small enough ε we get that $\alpha_i \geqslant \varepsilon^{0.91} \geqslant 2|\Delta_i|$. Therefore, Combining Fact 34 and Fact 26 we get

$$\begin{aligned}|d_j(x)| = |r_j(x) - r_j(x+\Delta)| &\leqslant \sum_{\substack{S\subseteq[k_j]\\ S\neq[k_j]}}\prod_{r\in S} g_j(x_r)\prod_{r\notin S}|g_j(x_r) - g_j(x_r+\Delta_r)| \\ &\leqslant \sum_{\substack{S\subseteq[k_j]\\ S\neq[k_j]}} B^{|S|} r_j(x)\prod_{r\notin S}\frac{|\Delta_i|}{\alpha_i}.\end{aligned}$$

Bounding $\frac{|\Delta_i|}{\alpha_i} \leqslant \varepsilon^{0.95}/\varepsilon^{0.91} = \varepsilon^{0.04}$ we get that

$$|r_j(x) - r_j(x+\Delta)| \leqslant r_j(x)\varepsilon^{0.04}\sum_{\substack{S\subseteq[k_j]\\ S\neq[k_j]}} B^{|S|} \leqslant B'^n\varepsilon^{0.04} r_j(x) \leqslant r_j(x)/2 = r_j/2.$$

Therefore, we may apply Claim 30 and get that

$$\begin{aligned}(III) &\lesssim \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{\Delta}}\left[\sum_{j=1}^{m}\frac{|r_j(\mathbf{x}) - r_j(\mathbf{x}+\mathbf{\Delta})|}{r_j(\mathbf{x})}\cdot\frac{\min(r_j(\mathbf{x}), T'(\mathbf{x}))}{T'(\mathbf{x})}1_E 1_{r_j(\mathbf{x})>\varepsilon^{2.7}}\right] \\ &\lesssim \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{\Delta}}\left[\sum_{j=1}^{m}\frac{|r_j(\mathbf{x}) - r_j(\mathbf{x}+\mathbf{\Delta})|}{r_j(\mathbf{x})}\cdot\frac{\min(r_j(\mathbf{x}), T'(\mathbf{x}))}{T'(\mathbf{x})+\varepsilon^{1.6}}1_E 1_{r_j(\mathbf{x})>\varepsilon^{2.7}}\right], \qquad (14)\end{aligned}$$

where the last inequality holds since $T'(\mathbf{x}) \gtrsim \varepsilon^{1.6}$. Next, we wish to discard $\mathbf{x}_i$ that are very far from their closest center z_j. For each j, note that $\left[z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}\right]$ is exactly the set of y's on which $g_j(y) = 0$, and let $R_j \subseteq I_j$ be $R_j = \left[z_j - \frac{\log n}{25n} - \varepsilon^{0.95}, z_j + \frac{\log n}{25n} + \varepsilon^{0.95}\right] \setminus \left[z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}\right]$. Note that for each $y \in I_j \setminus R_j$, we have that either $g_j(y) = 0$ if $y \in \left[z_j - \frac{\log n}{50n}, z_j + \frac{\log n}{50n}\right]$, and otherwise $g_j(y) = 1$. Furthermore, in the latter case we also have that $g_j(y + \mathbf{\Delta}_i) = 1$ since $|\mathbf{\Delta}_i| \leqslant \varepsilon^{0.95}$.

We sample $\mathbf{x}$ in the following way. First, sample $\mathbf{t}_1, \ldots, \mathbf{t}_m$ the number of $\mathbf{x}_i$'s in each interval $I_1, \ldots, I_m$, then for each j sample $\mathbf{k}_j$ to be the number of x_i's inside the interval I_j that fall inside R_j. Finally, for each $j \in [m]$ sample $\mathbf{k}_j$ points uniformly from R_j, $\mathbf{t}_j - \mathbf{k}_j$ uniformly from $I_j \setminus R_j$, and let $\mathbf{x}$ be the (multi-)set of all the sampled points. We condition on the $\mathbf{t}_j$'s and $\mathbf{k}_j$'s henceforth in (14). Furthermore, we condition on the identity of the i's for which $\mathbf{x}_i \in I_j$ for each j.

Since i's for which $\mathbf{x}_i \in I_j \in [z_j - \frac{\log n}{25n} - \varepsilon^{0.95}, z_j + \frac{\log n}{25n} + \varepsilon^{0.95}]$ do not affect both $r_j(\mathbf{x})$ and $r_j(\mathbf{x} + \mathbf{\Delta})$, we may ignore them and hence take expectation only over i's such $\mathbf{x}_i \in R_j$. Call these y's. Then from (14) we get

$$\begin{aligned}(III) &\lesssim \mathop{\mathbb{E}}_{\vec{\mathbf{t}},\vec{\mathbf{k}}}\left[\mathop{\mathbb{E}}_{\mathbf{y},\mathbf{\Delta}}\left[\sum_{j=1}^{m} \frac{|r_j(\mathbf{y}) - r_j(\mathbf{y}+\mathbf{\Delta})|}{r_j(\mathbf{y})} \cdot \frac{\min(r_j(\mathbf{y}), T'(\mathbf{y}))}{T'(\mathbf{y}) + \varepsilon^{1.6}}\right]\right] \\ &\leqslant \mathop{\mathbb{E}}_{\vec{\mathbf{t}},\vec{\mathbf{k}}}\left[\sum_{j=1}^{m} \mathop{\mathbb{E}}_{\mathbf{y},\mathbf{\Delta}}\left[\frac{|r_j(\mathbf{y}) - r_j(\mathbf{y}+\mathbf{\Delta})|}{r_j(\mathbf{y})} \cdot \frac{\min(r_j(\mathbf{y}), T'_{-j}(\mathbf{y}))}{r_j(\mathbf{y}) + T'_{-j}(\mathbf{y}) + \varepsilon^{1.6}}\right]\right],\end{aligned}$$

where $T'_{-j}(x) = \sum\limits_{j' \neq j} r_{j'}(x) 1_{r_{j'}(x) \geqslant \varepsilon^{2.7}}$. Note that conditioned on $\vec{\mathbf{t}} = \vec{t}, \vec{\mathbf{k}} = \vec{k}$, the values of $\mathbf{y}_i$'s such that $\mathbf{y}_i \in I_j$ are independent of $T'_{-j}(\mathbf{y})$, and they are distributed uniformly over R_j. Therefore, using Claim 32 we have

$$\begin{aligned}(III) &\lesssim \mathop{\mathbb{E}}_{\vec{\mathbf{t}},\vec{\mathbf{k}}}\left[\sum_{j=1}^{m} \varepsilon^{1.05} + \mathbf{k}_j \frac{\sigma n}{\log n} \cdot \Pr_{\mathbf{y}}\left[r_j(\mathbf{y}) \geqslant T'_{-j}(\mathbf{y}) \;\middle|\; \vec{\mathbf{t}}, \vec{\mathbf{k}}\right] + \mathop{\mathbb{E}}_{\mathbf{y}}\left[\sigma \frac{n}{\log n} \sqrt{\mathbf{k}_j} \frac{r_j(\mathbf{y})}{T'(\mathbf{y})}\right]\right]. \\ &\leqslant m\varepsilon^{1.05} + n^2 \sigma \sum_{j=1}^{m} \Pr_{y}\left[r_j(y) \geqslant T'_{-j}(y)\right] + \sigma \frac{n}{\log n} \mathop{\mathbb{E}}_{\vec{t},\vec{k}}\left[\sqrt{\max_j k_j}\right].\end{aligned}$$

Note that if $T'_{-j}(x) \leqslant r_j(x)$, then

$$T(x) \leqslant T'_{-j}(x) + r_j(x) + \sum_{j'} r_{j'}(x) 1_{r_{j'}(x) \leqslant \varepsilon^{2.7}} \leqslant 2r_j(x) + m \cdot \varepsilon^{2.7} \leqslant 3,$$

so we bound the sum on the right hand side by $m \Pr_x[T(x) \leqslant 3]$. For the expectation, we use Cauchy-Schwarz and overall we get

$$(III) \leqslant m\varepsilon^{1.05} + n^3 \sigma \Pr_{\mathbf{x}}[T(\mathbf{x}) \leqslant 3] + \sigma \frac{n}{\log n} \sqrt{\mathop{\mathbb{E}}_{\vec{\mathbf{t}},\vec{\mathbf{k}}}\left[\max_j \mathbf{k}_j\right]}.$$

The first term is clearly $\lesssim \varepsilon$. For the second term, we use Claim 35 below, that asserts that $\Pr_{\mathbf{x}}[T(\mathbf{x}) \leqslant 3] \leqslant n^{-\omega(1)}$, hence by the definition of σ the second term is also $\lesssim \varepsilon$. For the third term, note that each $\mathbf{k}_j$ is a sum of n independent Berounlli random variables with parameter $p \leqslant \log n/n$, therefore by Chernoff bound

$$\Pr[\mathbf{k}_j \geqslant 10 \log n] \leqslant e^{-\frac{1}{3} 9^2 \log n} \leqslant n^{-9}.$$

The union bound now implies that $\Pr[\max_j \mathbf{k}_j \geqslant 10 \log n] \leqslant n^{-8}$, and hence

$$\mathop{\mathbb{E}}_{\vec{\mathbf{t}},\vec{\mathbf{k}}}\left[\max_j \mathbf{k}_j\right] \leqslant n^{-8} \cdot n + 10 \log n \lesssim \log n.$$

Using the definition of σ, we get that the third term is also $\lesssim \varepsilon$. Combining all, we get that $(III) \lesssim \varepsilon$, and we are done.

▷ Claim 35.

$$\Pr_{\mathbf{x}}\left[\sum_j r_j(\mathbf{x}) \leqslant 3\right] < n^{-\omega(1)}. \tag{15}$$

Proof. The proof is very similar to the analysis of Case (B) above. In particular, similarly to inequality (3),

$$\Pr[r_j(\mathbf{x}) < 1] = \left(1 - \frac{2m\log n}{25n}\right)^{t_j} \geqslant \left(1 - \frac{2m\log n}{25n}\right)^{2\cdot n/m} > e^{-2\log n/25} = n^{-2/25},$$

as long as $t_j < 2 \cdot n/m$ (which is the case except with probability $n^{-\omega(1)}$. Since $m > n^{2/25} \cdot n^{\Omega(1)}$, the probability of not having at least three $r_j(x)$'s equal to 1 is $n^{-\omega(1)}$. ◁

5 The value of the t-fold symmetric odd cycle game

5.1 The upper bound: Theorem 7

Suppose that $n = 2m - 1$ and A is a strategy for $C_n^{\otimes_{\mathsf{sym}} t}$. We will view A as a symmetric function over ordered t tuples, i.e. as $A\colon C_n^t \to \{0,1\}^t$ satisfying $A(\pi(x)) = \pi(A(x))$ for all permutations π over $[t]$.

We identify $C_n = \left\{\frac{i}{n} \,\middle|\, i = 0, 1, \ldots, n-1\right\}$, consider the lattice $L = (C_n + \mathbb{Z})^t$ and define a rounding map $R\colon L \to \mathbb{Z}^t$ on it as follows. For $x \in C_n^t$, we define $R(x) = A(x) + nx \pmod 2$, and then we extend R to L by $R(x+z) = R(x) + z$ for $x \in C_n^t$ and $z \in \mathbb{Z}^t$.

Let $D = R^{-1}(0^t)$. The symmetry of A implies that D is permutation-symmetric, and we also note that D is a tiling of the lattice L.

▶ **Definition 36.** *A random ε-Bernoulli direction, denoted by $\mathbf{u} \sim \mathsf{B}(\varepsilon)$, is a random variable distributed on $\left\{\pm\frac{1}{n}, 0\right\}$, such that for each $i \in [t]$ independently,* $\Pr[\mathbf{u}_i = 0] = 1 - 2\varepsilon$ *and* $\Pr[\mathbf{u}_i = 1/n] = \Pr[\mathbf{u}_i = -1/n] = \varepsilon$.

We will mostly be concerned with $\varepsilon = 1/4$, in which case the distribution of $\mathbf{x}, \mathbf{x}+\mathbf{u} \pmod 1$ where $\mathbf{x} \in_R C_n^t$ and $\mathbf{u}$ is an independent $\frac{1}{4}$-Bernoulli step, is exactly the distribution of challenges to the players. Inspecting, we see that players succeed on these challenges if and only if $R(\mathbf{x}) = R(\mathbf{x}+\mathbf{u})$, as the following claim shows.

▷ Claim 37. Let $x \in C_n^t$ and $u \in \left\{\pm\frac{1}{n}, 0\right\}^t$. Then the players succeed on challenges $(x, x+u \pmod 1)$ if and only if $R(x) = R(x+u)$.

Proof. Note that x and $x+u$ are either in the same cell of D or in adjacent cells, so to prove the statement it is enough to show that the players succeed on the challenge if and only if $R(x) = R(x+u) \pmod 2$.

Write $x + u = d + z$ where $d \in C_n^t$ is $x + u \pmod 1$, and $z \in \mathbb{Z}^t$. Note that

$$R(x+u) = R(d) + z = A(d) + dn + z \pmod 2, \qquad R(x) = A(x) + nx \pmod 2$$

and subtracting the equations we get that

$$R(x+u) - R(x) = A(d) - A(x) + dn + z - nx \pmod 2.$$

Multiplying the equality $x + u = d + z$ by n and taking modulo 2 we get that $nu + nx = nd + nz = nd + z \pmod 2$ where the last transition used the fact that n is odd. Thus, $R(x+u) - R(x) = A(d) - A(x) + nu \pmod 2$. Note that the players succeed on the challenge if and only if $A(x) = A(d) + nu \pmod 2$, and plugging that in we get that they succeed if and only if $R(x+u) - R(x) = 0 \pmod 2$, as desired. ◁

Claim 37 implies that the failure probability of the players is

$$\Pr_{\mathbf{x}\in C_n^t, \mathbf{u}\sim \mathbf{B}(1/4)} [\mathbf{x}, \mathbf{x}+\mathbf{u} \text{ are in different cells of } D].$$

Setting $\mathbf{y} = \mathbf{x} \pmod{D}$, it is easily seen that the distribution of $\mathbf{y}$ is uniform over D, so the probability of the above event is equal to

$$\eta \overset{def}{=} \Pr_{\mathbf{y}\in D, \mathbf{u}\sim \mathsf{B}(1/4)} [\mathbf{y}+\mathbf{u} \notin D].$$

The rest of the proof is devoted to lower bounding η. Setting $k = M\frac{n\sqrt{\log t}}{t}$ for large constant M to be determined later, we show:

▶ **Lemma 38.** $\eta \geqslant \Omega(1/k)$.

Below, we will assume k is an integer, otherwise we may multiply it by a constant factor close to 1 and make it an integer. We then further assume k is prime, otherwise we may find a prime in $[k, 2k]$ and replace k by it. Define $\delta = \Pr_{\mathbf{x}\in D, \mathbf{u}\sim \mathsf{B}(1/4)} [\mathbf{x} + k\mathbf{u} \notin D]$ and observe the following easy relation between δ and η.

▷ Claim 39. $\delta \leqslant k\eta$.

Proof. By the union bound

$$\delta \leqslant \sum_{j=0}^{k-1} \Pr_{\mathbf{x}\in D, \mathbf{u}} [\mathbf{x}+j\mathbf{u} \in D, \mathbf{x}+(j+1)\mathbf{u} \notin D].$$

Note that for each j, the distribution of $y = \mathbf{x} + j\mathbf{u} \pmod{D}$ is uniform over D, the jth term in the above sum is at most $\Pr_{\mathbf{y}\in D, \mathbf{u}} [\mathbf{y} + \mathbf{u} \notin D] = \eta$. ◁

5.1.1 Disjoint Bernoulli steps

We will also consider the situation after making two Bernoulli steps whose support is disjoint, and for that we make the following definition.

▶ **Definition 40.** *The distribution of two disjoint ε-Bernoulli direction, denoted by $(\mathbf{u}^1, \mathbf{u}^2) \sim \mathsf{DB}(\varepsilon)$, is defined as follows. For each i independently, set each one of the following options with probability $\frac{\varepsilon}{2}$: $(\mathbf{u}^1_i, \mathbf{u}^2_i) = (1/n, 0)$, $(\mathbf{u}^1_i, \mathbf{u}^2_i) = (-1/n, 0)$, $(\mathbf{u}^1_i, \mathbf{u}^2_i) = (0, 1/n)$, $(\mathbf{u}^1_i, \mathbf{u}^2_i) = (0, -1/n)$; otherwise, set $(\mathbf{u}^1_i, \mathbf{u}^2_i) = (0, 0)$.*

We note that if $(\mathbf{u}^1, \mathbf{u}^2) \sim \mathsf{DB}(\varepsilon)$, then $\mathbf{u}^1 + \mathbf{u}^2$ is distributed as $\mathsf{B}(\varepsilon)$. Therefore:

▷ Claim 41. It holds that:

- $\Pr_{\mathbf{x}\in D, \mathbf{u}\sim \mathsf{B}(1/4)} [\mathbf{x} + k\mathbf{u} \notin D] \leqslant 2\delta$;
- $\Pr_{\mathbf{x}\in D, \mathbf{u}\sim \mathsf{B}(1/4)} [\mathbf{x} + \mathbf{u} \notin D] \leqslant 2\eta$.

Proof. We prove the first item, and the second item is proved analogously. To sample $\mathbf{u} \sim \mathsf{B}(1/4)$, we sample $(\mathbf{u}^1, \mathbf{u}^2) \sim \mathsf{DB}(1/4)$ and take $\mathbf{u} = \mathbf{u}^1 + \mathbf{u}^2$, so by the union bound the probability in the first item is at most

$$\Pr_{\mathbf{x}\in D, (\mathbf{u}^1, \mathbf{u}^2)\sim \mathsf{DB}(1/4)} \left[\mathbf{x} + k\mathbf{u}^1 \notin D\right] + \Pr_{\mathbf{x}\in D, (\mathbf{u}^1, \mathbf{u}^2)\sim \mathsf{DB}(1/4)} \left[\mathbf{x} + k\mathbf{u}^1 \in D, \mathbf{x} + k\mathbf{u}^1 + k\mathbf{u}^2 \notin D\right].$$

The first probability is δ, and we argue that the second probability is at most the first. Indeed, setting $\mathbf{y} = \mathbf{x} + k\mathbf{u}^1$, this probability is at most the probability that $\mathbf{y}, \mathbf{y} + k\mathbf{u}^2$ are in different cells of D. Note that this occurs if and only if $\mathbf{y} \pmod D$ and $\mathbf{y} \pmod D + k\mathbf{u}^2$ are in different cells of D; note also that for every fixing of $\mathbf{u}^1$, the distribution of $\mathbf{y} \pmod D$ is uniform over D. Thus

$$\Pr_{\mathbf{x}\in D,(\mathbf{u}^1,\mathbf{u}^2)\sim \mathsf{DB}(1/4)}\left[\mathbf{x}+k\mathbf{u}^1\in D, \mathbf{x}+k\mathbf{u}^1+k\mathbf{u}^2\notin D\right]$$
$$\leqslant \Pr_{\mathbf{y}\in D,(\mathbf{u}^1,\mathbf{u}^2)\sim \mathsf{DB}(1/4)}\left[\mathbf{y}+k\mathbf{u}^2\notin D\right]=\delta. \qquad ◀$$

▶ **Definition 42.** *Let $x \in D$ and u be a direction. We say (x,u) is decent if*

$$\Pr_{(\mathbf{u}^1,\mathbf{u}^2)\sim \mathsf{DB}(1/4)}\left[x+\mathbf{u}_1\notin D\vee x+\mathbf{u}_2\notin D\vee x+k\mathbf{u}_1\notin D\vee x+k\mathbf{u}_2\notin D \mid \mathbf{u}^1+\mathbf{u}^2=u\right]$$
$$<\frac{1}{32}.$$

▷ Claim 43. $\Pr_{\mathbf{x}\in_R D,\mathbf{u}\sim \mathsf{B}(1/4)}\left[(\mathbf{x},\mathbf{u}) \text{ is decent}\right]\geqslant 1-64(\eta+\delta)$

Proof. Denote

$$p(x,u)$$
$$=\Pr_{(\mathbf{u}^1,\mathbf{u}^2)\sim \mathsf{DB}(1/4)}\left[x+\mathbf{u}_1\notin D\vee x+\mathbf{u}_2\notin D\vee x+k\mathbf{u}_1\notin D\vee x+k\mathbf{u}_2\notin D \mid \mathbf{u}^1+\mathbf{u}^2=u\right].$$

Note that

$$\mathop{\mathbb{E}}_{\substack{\mathbf{x}\in_R D\\ \mathbf{u}\sim \mathsf{B}(1/4)}}\left[p(\mathbf{x},\mathbf{u})\right]=\Pr_{\substack{\mathbf{x}\in_R D\\ (\mathbf{u}^1,\mathbf{u}^2)\sim \mathsf{DB}(1/4)}}\left[\mathbf{x}+\mathbf{u}^1\notin D\vee \mathbf{x}+\mathbf{u}^2\notin D\vee \mathbf{x}+k\mathbf{u}^1\notin D\vee \mathbf{x}+k\mathbf{u}^1\notin D\right],$$

which is at most $2(\delta+\eta)$ by the union bound. Thus, by Markov's inequality

$$\Pr_{\mathbf{x}\in_R D,\mathbf{u}\sim \mathsf{B}(1/4)}\left[(\mathbf{x},\mathbf{u}) \text{ is not decent}\right]=\Pr_{\mathbf{x}\in_R D,\mathbf{u}\sim \mathsf{B}(1/4)}\left[p(\mathbf{x},\mathbf{u})\geqslant\frac{1}{32}\right]\leqslant 64(\delta+\eta). \qquad ◀$$

5.1.2 Analyzing the potential function

Our argument closely follows the argument in Section 3, and below we focus on the necessary adjustments. Set $Z=\frac{t}{10\log t}$. The definition of the potential function stays as is. We will have several constants floating around in the proof which are not important for the most part, however we make the distinction between the constants $c_1,\ldots,c_6$ that will be absolute (i.e. not depending on M), and the constants $t_0(M),t_1(M),t_2(M)$ that will depend on M.

The following is a variant of Claim 14, which is the main difference with the argument from Section 3.

▷ Claim 44. If $x, x+u, x-u, x+ku, x-ku\in D$ and both $(x,u),(x,-u)$ are decent, then

$$|\Psi(x+ku)-\Psi(x,ku)|\leqslant t^2\cdot e^{-Z/4}.$$

Proof. We consider the contribution of each pair (i,j) to $\Psi(x+ku)$ and $\Psi(x,ku)$ separately. Without loss of generality we may only consider pairs i,j that $\gamma(x_i,x_j)=1$, and thus $d(x_i,x_j)=x_i-x_j+z$ for some $z\in\mathbb{Z}$, $z\neq 0$. Let $d=x_i-x_j+z+k(u_i-u_j)$.

▶ **Proposition 45.** $d\geqslant 0$.

Proof. Assume otherwise. Since $x_i - x_j + z \geqslant 0$ it follows by continuity that there is $\lambda \in [0,1)$ such that $x_i - x_j + z + \lambda k(u_i - u_j) = 0$. Note that $u_i - u_j$ can either be $0, \pm\frac{1}{n}, \pm\frac{2}{n}$. If $u_i - u_j = 0$, we get that $x_i - x_j + z = 0$, and as $x \in D$ this contradicts Lemma 12. Otherwise, multiplying by n, we get that $\lambda kn(u_i - u_j)$ is an integer. Note that $kn(u_i - u_j)$ is either $\pm k$ or $\pm 2k$, and as k is prime we get that $\lambda = \frac{1}{2}$, $\lambda = \frac{1}{k}$ or $\lambda = \frac{1}{2k}$, and we analyze each case separately. If $\lambda = \frac{1}{k}$ then we get $x_i - x_j + u_i - u_j + z = 0$, so $x + u \in D$ has two coordinates differing by a non-zero integer, contradicting Lemma 12. We next consider the other two cases separately, and assume that $u_i - u_j > 0$ – otherwise we use $-u$ instead of u in the argument below.

If $\lambda = \frac{1}{2k}$, then necessarily $u_i - u_j = \frac{2}{n}$ and and we get that $x_i - x_j + z + \frac{1}{n} = 0$. Sample $(\mathbf{u}^1, \mathbf{u}^2) \sim \mathsf{DB}(1/4)$ conditioned on $\mathbf{u}^1 + \mathbf{u}^2 = u$. Note that the event that $\mathbf{u}^1_i = 1/n$ and $\mathbf{u}^1_j = 0$ occurs with probability $1/32$. Since (x, u) is decent, we get that $x + \mathbf{u}^1 \in D$ with probability strictly greater than $\frac{31}{32}$. Thus, the probability that $x + \mathbf{u}^1 \in D$ and $(\mathbf{u}^1_i, \mathbf{u}^1_j) = (1/n, 0)$ is positive, and in this case we get

$$(x + \mathbf{u}^1)_i - (x + \mathbf{u}^1)_j = x_i - x_j + \frac{1}{n} = -z \neq 0,$$

contradicting Lemma 12.

The case that $\lambda = \frac{1}{2}$ is similar. We must have that $u_i - u_j = \frac{2}{n}$, and thus we get $x_i - x_j + \frac{k}{n} + z = 0$. Sample $(\mathbf{u}^1, \mathbf{u}^2) \sim \mathsf{DB}(1/4)$ conditioned on $\mathbf{u}^1 + \mathbf{u}^2 = u$. Note that the event that $\mathbf{u}^1_i = 1/n$ and $\mathbf{u}^1_j = 0$, occurs with probability $1/32$. Since (x, u) is decent, we get that $x + k\mathbf{u}^1 \in D$ with probability strictly greater than $\frac{31}{32}$. Thus, the probability that $x + k\mathbf{u}^1 \in D$ and $(\mathbf{u}^1_i, \mathbf{u}^1_j) = (1/n, 0)$ is positive, and in this case we get

$$(x + k\mathbf{u}^1)_i - (x + k\mathbf{u}^1)_j = x_i - x_j + \frac{k}{n} = -z \neq 0,$$

contradicting Lemma 12. ◀

We therefore get that $d \geqslant 0$, and the rest of the proof is identical to the proof of Claim 14. ◁

▷ Claim 46. There is an absolute constants $c_1 > 0$ and $t_0(M) > 0$, such that if $t \geqslant t_0$ then for every $x \in D$

$$\Psi(x) \cdot e^{c_1 k^2 Z^2/n^2} \leqslant \mathop{\mathbb{E}}_{\mathbf{u} \sim \mathbf{B}(1/4)} [\Psi(x, k\mathbf{u})] \leqslant \Psi(x) \cdot e^{c_1^{-1} k^2 Z^2/n^2}.$$

Proof. By linearity of expectation we have

$$\mathop{\mathbb{E}}_{\mathbf{u} \sim \mathbf{B}(1/4)} [\Psi(x, k\mathbf{u})] = \sum_{i<j} e^{-Z \cdot d(x_i, x_j)} \cdot \mathop{\mathbb{E}}_{\mathbf{u} \sim \mathbf{B}(1/4)} \left[e^{-Z \cdot \gamma(x_i, x_j) \cdot k(\mathbf{u}_i - \mathbf{u}_j)} \right].$$

Note that the above expectation does not depend on i, j: for every i, j the distribution of $\mathbf{u}_i - \mathbf{u}_j$ is $\mathbf{w}$, where $\Pr[\mathbf{w} = 2/n] = \Pr[\mathbf{w} = -2/n] = \frac{1}{16}$, $\Pr[\mathbf{w} = 1/n] = \Pr[\mathbf{w} = -1/n] = \frac{1}{4}$, $\Pr[\mathbf{w} = 0] = \frac{3}{8}$. In particular, this distribution is symmetric around 0 and thus the sign $\gamma(x_i, x_j)$ does not affect the expectation. Hence we have

$$\mathop{\mathbb{E}}_{\mathbf{u}} [\Psi(x, \mathbf{u})] = \Psi(x) \cdot \mathop{\mathbb{E}}_{\mathbf{w}} [e^{kZ \cdot \mathbf{w}}] = \Psi(x) \cdot \mathop{\mathbb{E}}_{\mathbf{w}} \left[\frac{e^{kZ \cdot \mathbf{w}} + e^{-kZ \cdot \mathbf{w}}}{2} \right].$$

Note that $|kZ \cdot \mathbf{w}| \leqslant M \frac{n\sqrt{\log t}}{t} \frac{t}{10 \log t} \frac{1}{n} \leqslant 1$ for large enough t, so we have that

$$e^{c_1 (kZ \cdot \mathbf{w})^2} \leqslant \frac{e^{kZ \cdot \mathbf{w}} + e^{-kZ \cdot \mathbf{w}}}{2} \leqslant e^{c_1^{-1} (kZ \cdot \mathbf{w})^2}.$$

Finally, the expectation of $e^{c(kZ\cdot\mathbf{w})^2}$ is at least $e^{c'k^2Z^2/n^2}$ and at most $e^{c''k^2Z^2/n^2}$, and the claim follows. $\lhd$

The proofs of the following several claims are essentially identical to their analogs in Section 3, and are therefore omitted. We say a point x is *good* if any interval of length $\frac{10\log t}{t}$ on the circle contains at least $\log t$ and at most $100\log t$ coordinates from x (mod 1). By Chernoff bound, a random $x \in D$ is good with probability > 0.999 given t is large enough.

$\rhd$ Claim 47. There exists an absolute constant $c_2 > 0$, such that if x is good then $\Psi(x) > c_2 \log^2 t$.

Proof. The proof is identical to the proof of Claim 16. $\lhd$

$\rhd$ Claim 48. There exists an absolute constant $c_3 > 0$, such that if x is good, then for all i we have $C_i < c_3\frac{\Psi(x)}{\log t}$.

Proof. The proof is identical to the proof of Claim 17. $\lhd$

$\rhd$ Claim 49. There exists an absolute constant $c_5, c_6 > 0$ and $t_1(M) > 0$, such that if $t \geqslant t_1$ then for all good $x \in D$ we have

$$\mathsf{var}_{\mathbf{u}\sim\mathbf{B}(1/4)}[\Psi(x,\mathbf{u})] \leqslant \frac{c_5}{\log t}\cdot\left(e^{c_6^{-1}\frac{k^2Z^2}{n^2}} - e^{c_6\frac{k^2Z^2}{n^2}}\right)\cdot\Psi(x)^2.$$

Proof. The proof is a straightforward adaptation of the proof of Claim 18. $\lhd$

Consequently, we have to adjust Claim 19 as follows.

$\rhd$ Claim 50. There is an absolute constant $M > 0$ and $t_2 > 0$ such that if $k = M\frac{n\sqrt{\log t}}{t}$ and $t \geqslant t_1$, then for all good $x \in D$ we have

$$\Pr_{\mathbf{u}\sim\mathbf{B}(1/4)}\left[\Psi(x,\mathbf{u}) > \Psi(x) + \frac{c_1^2}{2}\frac{k^4Z^4}{n^4}\Psi(x)\right] \geqslant 0.99.$$

Proof. Let $c_1, \ldots, c_6$ be the constants from the previous claims, and choose $M = \sqrt{\frac{200c_5}{c_1^2c_6}}$. Then take $t_0(M)$, $t_1(M)$ from Claims 46 49 and choose $t_2(M) = \max(t_0(M), t_1(M))$. We upper bound the probability of the complement event. Using Claim 46 (and $e^t \geqslant 1+t+t^2/2$), we get

$$\mathop{\mathbb{E}}_{\mathbf{u}\sim\mathbf{B}(1/4)}[\Psi(x,\mathbf{u})] \geqslant \Psi(x)\cdot\left(1 + c_1\frac{k^2Z^2}{n^2} + \frac{c_1^2}{2}\frac{k^4Z^4}{n^4}\right).$$

Hence

$$\begin{aligned}&\Pr_{\mathbf{u}\sim\mathbf{B}(1/4)}\left[\Psi(x,\mathbf{u}) \leqslant \Psi(x) + \frac{c_1^2}{2}\frac{k^4Z^4}{n^4}\Psi(x)\right]\\ \leqslant\ &\Pr_{\mathbf{u}\sim\mathbf{B}(1/4)}\left[\left|\Psi(x,\mathbf{u}) - \mathop{\mathbb{E}}_{\mathbf{u}'\sim\mathbf{B}(1/4)}[\Psi(x,\mathbf{u}')]\right| \geqslant \Psi(x)c_1\frac{k^2Z^2}{n^2}\right].\end{aligned}$$

We want to upper bound the probability of the last event using Chebyshev's inequality. Since x is good, the conclusion of Claim 49 holds, and so

$$\mathsf{var}_{\mathbf{u}\sim\mathbf{B}(1/4)}[\Psi(x,\mathbf{u})] \leqslant \frac{c_5}{\log t}\left(e^{c_6^{-1}\frac{k^2Z^2}{n^2}} - e^{c_6\frac{k^2Z^2}{n^2}}\right)\cdot\Psi(x)^2 \leqslant \frac{c_5}{\log t}\cdot\frac{2c_6^{-1}k^2Z^2}{n^2}\cdot\Psi(x)^2,$$

for sufficiently large t. Therefore, applying Chebyshev's inequality we see the probability in question is at most

$$\frac{\mathsf{var}_{\mathbf{u}\sim\mathbf{B}(1/4)}[\Psi(x,\mathbf{u})]}{\Psi(x)^2\cdot c_1^2\frac{k^4Z^4}{n^4}} \leqslant \frac{\frac{c_5}{\log t}\cdot\frac{2c_6^{-1}k^2Z^2}{n^2}\cdot\Psi(x)^2}{\Psi(x)^2\cdot c_1^2\frac{k^4Z^4}{n^4}} = \frac{2c_5}{c_1^2c_6}\frac{n^2}{k^2Z^2\log t} = \frac{2c_5}{c_1^2c_6}\frac{1}{M^2} \leqslant 0.01. \blacktriangleleft$$

5.1.3 Finishing the argument

For each u, denote $\delta_u = \Pr_{\mathbf{x}\in D}\left[\mathbf{x}+ku \notin D\right]$, and note that $\delta = \mathbb{E}_{\mathbf{u}}\left[\delta_{\mathbf{u}}\right]$.

▷ Claim 51. For each u, $\mathcal{D}_{TV}[\mathbf{x};\mathbf{x}-ku] \leqslant \delta_u + \delta_{-u}$.

Proof. The proof is a direct conversion of the proof of Claim 20 to the discrete setting, replacing the notion of "Borel sets" with finite sets. ◁

We can now prove Lemma 38.

Proof of Lemma 38. Take M and t_2 from Claim 50. We may assume that $t \geqslant t_2$, otherwise the lemma just follows from the fact that $\eta \geqslant \Omega(1/n)$, which holds as the value of the t-fold symmetric repeated game is at most the value of the original game, which is $1-\Theta(1/n)$.

Take $\mathbf{x} \in_R D$, $\mathbf{u} \sim \mathsf{B}(1/4)$. Let E_1 be the event that $(\mathbf{x},\mathbf{u}),(\mathbf{x},-\mathbf{u})$ are decent, E_2 be the event that $\Psi(\mathbf{x}) \leqslant c_2 \log^2 t$, E_3 the event that $\mathbf{x}+k\mathbf{u}, \mathbf{x}-k\mathbf{u}, \mathbf{x}+\mathbf{u}, \mathbf{x}-\mathbf{u} \in D$, and let E_4 be the event that $\Psi(\mathbf{x},\mathbf{u}) \geqslant \Psi(\mathbf{x}) + \frac{c_1^2}{2}\frac{k^4 Z^4}{n^4}\Psi(\mathbf{x})$. Finally, let E_5 be the event that $\Psi(\mathbf{x}+\mathbf{u}) > \Psi(\mathbf{x})$ and denote $E(\mathbf{x},\mathbf{u}) = E_1 \cap \overline{E_2} \cap E_3 \cap E_4$. Note that if the event E holds for x,u, then E_5 also holds, since by Claim 44:

$$\Psi(x+u) \geqslant \Psi(x,u) - t^2 \cdot e^{-Z/4} \geqslant \Psi(x) + \frac{c_1^2}{2}\frac{k^4 Z^4}{n^4}\Psi(x) - t^2 \cdot e^{-Z/4} > \Psi(x).$$

In the last inequality, we used the fact that if E holds, then $\frac{c_1^2}{4}\frac{k^4 Z^4}{n^4}\Psi(x) \geqslant \Omega(1)$, and $t^2 \cdot e^{-Z/4} = n^2 e^{-t/40\log t} = o(1)$ for large enough t.

By Claim 43, $\Pr\left[E_1\right] \geqslant 1-128(\delta+\eta)$. By Claim 47 the probability of E_2 is at most the probability $\mathbf{x}$ is bad, hence it is at most 0.005, by Claim 41 $\Pr\left[E_3\right] \geqslant 1-4(\delta+\eta)$, and by Claim 50, $\Pr\left[E_4\right] \geqslant 0.99$. We thus get

$$\Pr_{\mathbf{x},\mathbf{u}}[E(\mathbf{x},\mathbf{u})] \geqslant 0.99 - 4(\delta+\eta) - 0.005 - 128(\delta+\eta) \geqslant 0.95 - 132(\delta+\eta). \tag{16}$$

Fix u. Using Claim 51 we get that

$$\Pr_{\mathbf{x}}\left[E(\mathbf{x}-u,u)\right] \geqslant \Pr_{\mathbf{x}}\left[E(\mathbf{x},u)\right] - \mathcal{D}_{TV}[\mathbf{x};\mathbf{x}-u] \geqslant \Pr_{\mathbf{x}}\left[E(\mathbf{x},u)\right] - \delta_u - \delta_{-u}.$$

By the union bound, we now conclude that

$$\Pr_{\mathbf{x}}\left[E(\mathbf{x}-u,u) \cap E(\mathbf{x},u)\right] \geqslant 1 - \Pr_{\mathbf{x}}\left[\overline{E(\mathbf{x}-u,u)}\right] - \Pr_{\mathbf{x}}\left[\overline{E(\mathbf{x},u)}\right] \geqslant 2\Pr_{\mathbf{x}}\left[E(\mathbf{x},u)\right] - 1 - \delta_u - \delta_{-u}.$$

Taking expectation over a random step $\mathbf{u}$, we get that

$$\Pr_{\mathbf{x},\mathbf{u}}\left[E(\mathbf{x}-\mathbf{u},\mathbf{u}) \cap E(\mathbf{x},\mathbf{u})\right] \geqslant 2\Pr_{\mathbf{x},\mathbf{u}}\left[E(\mathbf{x},\mathbf{u})\right] - 1 - 2\mathop{\mathbb{E}}_{\mathbf{u}}\left[\delta_{\mathbf{u}}\right] \geqslant 0.9 - 270(\delta+\eta),$$

where we used (16). Next, when both $E(x-u,u)$ and $E(x,u)$ hold, we have by the previous observation that E_5 holds for both pairs $(x-u,u)$ and (x,u), and so $\Psi(x+u) > \Psi(x) = \Psi((x-u)+u) > \Psi(x-u)$. Thus, we get that $\Pr_{\mathbf{x},\mathbf{u}}\left[\Psi(\mathbf{x}+\mathbf{u}) > \Psi(\mathbf{x}-\mathbf{u})\right] \geqslant 0.9 - 270(\delta+\eta)$. On the other hand, the probability on the left hand side is at most 0.5; this follows as $\Pr_{\mathbf{x},\mathbf{u}}\left[\Psi(\mathbf{x}+\mathbf{u}) > \Psi(\mathbf{x}-\mathbf{u})\right] = \Pr_{\mathbf{x},\mathbf{u}}\left[\Psi(\mathbf{x}-\mathbf{u}) > \Psi(\mathbf{x}+\mathbf{u})\right]$, and their sum is at most 1. Combining the two inequalities we get that $\eta+\delta \geqslant \Omega(1)$, which using Claim 39 implies that $\eta = \Omega(1/k)$ as desired. ◀

5.2 The lower bound: proof of Theorem 8

In this section we use the permutation-symmetric body constructed in Theorem 6 in order to prove Theorem 8.

5.2.1 Tools

We need the following isoperimetric inequality.

▶ **Fact 52.** *For all $\varepsilon > 0$ there is $\delta > 0$ such that the following holds. Let $A \subseteq [0,1]^n$ be a measurable set such that $\varepsilon \leqslant \mathsf{vol}(A) \leqslant 1-\varepsilon$. Then $\mathsf{area}(A \cap \mathsf{interior}([0,1]^n)) \geqslant \delta$.*

Proof. This is the combination of [29, Theorem 6, Theorem 7] as we explain below. Theorem 7 therein asserts that if $A \subseteq [0,1]^n$ has Lebesgue measure α and surface area S, then there is a measurable set in Gaussian space $B \subseteq \mathbb{R}^n$ with Gaussian measure α and (Gaussian) surface area at most S. Now [29, Theorem 7] asserts among sets with Gaussian measure α, the minimizers of surface area are halfspaces of the form $B_\beta = \{\, z \in \mathbb{R}^n \mid z_1 \leqslant \beta\}$ where β is chosen so that the Gaussian measure of B_β is α, so $S \geqslant \mathsf{surface} - \mathsf{area}(B_\beta)$, which is bounded away from 0 if α is bounded away from 0 and 1. ◀

Secondly, we need a slight strengthening of Theorem 6. Recall that in Sections 4 and B we have constructed a semi-algebraic, bounded tiling body $D \subseteq \mathbb{R}^t$ whose surface area is $A = O(t/\sqrt{\log t})$, and for small enough ε we have

$$\Pr_{\mathbf{x}\in D, \mathbf{\Delta}\sim N(0,\varepsilon^2 I_t)}[\mathbf{x}+\mathbf{\Delta} \notin D] \lesssim A\varepsilon.$$

We note that the argument in Section 4 holds in fact for more general class of $\mathbf{\Delta}$ (we only used the fact it is independent of $\mathbf{x}$, has mean 0 and is sub-Gaussian). Thus, we consider the distribution $\mathbf{\Delta}_\varepsilon \in \{0, \pm\varepsilon/n\}^t$ of Bernoulli steps, namely for each i independently choosing $(\mathbf{\Delta}_\varepsilon)_i$ as $\Pr[(\mathbf{\Delta}_\varepsilon)_i = 0] = \frac{1}{2}$, $\Pr\left[(\mathbf{\Delta}_\varepsilon)_i = -\frac{\varepsilon}{n}\right] = \frac{1}{4}$, $\Pr\left[(\mathbf{\Delta}_\varepsilon)_i = \frac{\varepsilon}{n}\right] = \frac{1}{4}$. Thus, running the argument therein we get:

▶ **Lemma 53.** *The distribution over tiling bodies $(D_{\vec{r}})_{\vec{r}}$ from Lemma 24 satisfies, for small enough $\varepsilon > 0$*

$$\mathop{\mathbb{E}}_{\vec{r}}\left[\Pr_{\mathbf{x},\mathbf{\Delta}_\varepsilon}[\textit{At least one of the conditions of Claim 22 fail for } \mathbf{x} \textit{ and } \mathbf{x}+\mathbf{\Delta}_\varepsilon]\right] \lesssim A\frac{\varepsilon}{n}.$$

Slightly adapting the argument from Section B, we may ensure that the chosen body D also has small noise sensitivity for Bernoulli random steps $\mathbf{\Delta}_\varepsilon$ for small enough ε,[3] but we will only need this to happen for a specific suitably chosen ε which can be ensured as follows. Take ε small enough for which Lemma 53 holds, and note that by Markov's inequality we get from Lemma 53 that

$$\Pr_{\vec{r}}\left[\Pr_{\mathbf{x},\mathbf{\Delta}_\varepsilon}[\text{At least one of the conditions of Claim 22 fail for } \mathbf{x} \text{ and } \mathbf{x}+\mathbf{\Delta}_\varepsilon] \geqslant C\cdot A\cdot\frac{\varepsilon}{n}\right]$$
$$\leqslant \frac{1}{4}$$

for an absolute constant C. Thus, from Claim 59 and the union bound we get that there is $\vec{r}^\star \in \cap_{k\geqslant k_0} G_k$ such that the above event holds, and the rest of the proof in Section B shows that $D = D_{\vec{r}^\star}$ has surface area $O(A)$. We summarize this discussion with the following lemma.

[3] The proof is essentially the same, adapting the definition of G_k therein to be
$$G_k = \left\{ \vec{r} \;\middle|\; \begin{array}{l} \Pr_{\substack{\mathbf{x}\in D_{\vec{r}} \\ \mathbf{\Delta}\sim N(0,4^{-k}\cdot I_n)}}[\mathbf{x},\mathbf{x}+\mathbf{\Delta} \text{ lie in different cells of } S_{\vec{r}}] \leqslant 4\cdot A2^{-k} \\ \Pr_{\substack{\mathbf{x}\in D_{\vec{r}} \\ \mathbf{\Delta}_{2^{-k}}}}[\mathbf{x},\mathbf{x}+\mathbf{\Delta}_{2^{-k}} \text{ lie in different cells of } S_{\vec{r}}] \leqslant 4\cdot A2^{-k} \end{array} \right\}.$$

▶ **Lemma 54.** *For all t, for small enough ε, there is a permutation-symmetric, bounded tiling body D with surface area $A = O(t/\sqrt{\log t})$ such that*

$$\Pr_{\mathbf{x},\mathbf{\Delta}_\varepsilon}\left[\textit{At least one of the conditions of Claim 22 fail for } \mathbf{x} \textit{ and } \mathbf{x}+\mathbf{\Delta}_\varepsilon\right] \lesssim A\cdot\frac{\varepsilon}{n}.$$

5.2.2 Decisive boxes

In this section, we use Lemma 54 to devise a symmetric strategy for the players in the t-fold repeated game. Take small enough ε so such Lemma 54 holds and assume that $k \overset{def}{=} 1/\varepsilon$ is an integer. Let D be the permutation-symmetric tiling body from Lemma 54. It will be convenient for us to think of challenges to the players as $C_n^t = \left\{\frac{i}{n} \,\middle|\, i = 0,1,\ldots,n-1\right\}$. Partition $[0,1)^t$ into the boxes $B_{\vec{a}} = \prod_{i=1}^{t}\left[\frac{a_i}{n}, \frac{a_i}{n}+\frac{1}{n}\right)$ for $\vec{a} \in \{0,1,\ldots,n-1\}^t$; it will be convenient for us identify a challenge of a player $\mathbf{x}'$ with the box it belongs to, i.e. with $B_{\vec{a}}$ for $\vec{a} = n\mathbf{x}'$. Consider the way D further partitions the boxes $B_{\vec{a}}$.

▶ **Definition 55.** *We say a box $B_{\vec{a}}$ is decisive if there exists $z \in \mathbb{Z}^n$ such that $\mu(B_{\vec{a}}\cap(D+z)) \geqslant \frac{2}{3}\mu(B_{\vec{a}})$. Otherwise, we say $B_{\vec{a}}$ is indecisive.*

We show that almost all boxes are decisive:

▶ **Lemma 56.** *The number of indecisive boxes is $O(An^{t-1})$.*

Proof. Define $\Phi = \sum_{z\in\mathbb{Z}^t}\sum_{\vec{a}\in\{0,1,\ldots,n-1\}^t} \mathsf{area}(\partial(D+z)\cap\mathsf{interior}(B_{\vec{a}}))$. By considering the surface area of D, we will show that $\Phi \leqslant A$, and we will lower bound Φ as a function of the number of the indecisive boxes, from which we will get the result. Let B be such that $D \subseteq [-B,B]^t$, and take m large enough.

5.2.2.1 The upper bound

For $\vec{a} \in \{0,1,\ldots,mn-1\}^t$, we define the box $B_{\vec{a}}$ as above, and define

$$\Phi_m = \sum_{z\in\mathbb{Z}^t}\sum_{\vec{a}\in\{0,1,\ldots,mn-1\}^t} \mathsf{area}(\partial(D+z)\cap\mathsf{interior}(B_{\vec{a}})).$$

On the one hand, we clearly have that $\Phi_m = m^t\Phi$, and we next upper bound Φ_m. Since $D \subseteq [-B,B]^t$, we have that

$$\begin{aligned}\Phi_m &= \sum_{z\in\{-B,-B+1,\ldots,B+m\}^t}\sum_{\vec{a}\in\{0,1,\ldots,mn-1\}^t} \mathsf{area}(\partial(D+z)\cap\mathsf{interior}(B_{\vec{a}}))\\ &\leqslant \sum_{z\in\{-B,-B+1,\ldots,B+m\}^t} \mathsf{area}(\partial(D+z))\\ &= (m+2B+1)^t\mathsf{area}(\partial D)\\ &\leqslant (m+2B+1)^t A.\end{aligned}$$

Combining the upper and lower bound we get $\Phi \leqslant \left(1+\frac{2B+1}{m}\right)^t A$, and sending m to infinity gets that $\Phi \leqslant A$.

5.2.2.2 The lower bound

Interchanging the order of summation, we write

$$\Phi = \sum_{\vec{a}\in\{0,1,\ldots,n-1\}^t}\sum_{z\in\mathbb{Z}^t}\mathsf{area}(\partial(D+z)\cap\mathsf{interior}(B_{\vec{a}})),$$

and we show that if the box $B_{\vec{a}}$ is indecisive, then the innermost sum is at least $\Omega(1/n^{t-1})$. Indeed, if $B_{\vec{a}}$ is indecisive, then $\mu(B_{\vec{a}}\cap(D+z))\leqslant\frac{2}{3}\mu(B_{\vec{a}})$ for all $\vec{a}$. Thus, we may find $P\subseteq\mathbb{Z}^n$ such that for $H=B_{\vec{a}}\cap\bigcup_{z\in P}(D+z)$ we have that $\frac{1}{6}\mu(B_{\vec{a}})\leqslant\mu(H)\leqslant\frac{5}{6}\mu(B_{\vec{a}})$. We now scale and translate H, i.e. take $H'=nH-\vec{a}$, so that the above translates to $H'\subseteq[0,1]^n$ such that $\frac{1}{6}\leqslant\mu(H')\leqslant\frac{5}{6}$, and hence by Fact 52 $\mathsf{area}(\partial H'\cap\mathsf{interior}([0,1]^n))\geqslant\Omega(1)$. Removing the scaling, we get that $\mathsf{area}(\partial H\cap\mathsf{interior}(B_{\vec{a}}))\geqslant\Omega(n^{1-t})$. Therefore, we get that

$$\Phi\geqslant\sum_{\substack{\vec{a}\in\{0,1,\ldots,n-1\}^t\\ B_{\vec{a}}\text{ indecisive}}}\Omega(n^{1-t})=\Omega(n^{1-t}\cdot\#\{\text{indecisive boxes}\})$$

Combining the upper and lower bound on Φ, we get that the number of indecisive boxes is at most $O(An^{t-1})$. ◀

Next, we show that if $B_{\vec{a}}$ is a typical decisive box, and $\mathbf{\Delta}_1\in_R\{0,\pm1/n\}$ is chosen randomly as above, then $B_{\vec{a}+\mathbf{\Delta}_1}$ is very likely to be somewhat decisive, and furthermore with the same cell of D.

▶ **Lemma 57.** *It holds that*

$$\Pr_{\substack{\mathbf{\Delta}_1\\ \vec{\mathbf{a}}\in\{0,1,\ldots,n-1\}^t}}\left[\exists z\in\mathbb{Z}^n,\mu(B_{\vec{\mathbf{a}}}\cap(D+z))\geqslant\frac{2}{3}\mu(B_{\vec{a}}),\mu(B_{\vec{\mathbf{a}}+n\mathbf{\Delta}_1}\cap(D+z))>\frac{1}{2}\mu(B_{\vec{\mathbf{a}}+n\mathbf{\Delta}_1})\right]$$
$$\geqslant 1-O\left(\frac{A}{n}\right).\tag{17}$$

Proof. Choose a random $\vec{\mathbf{a}}$, take a random $\mathbf{x}\in B_{\vec{\mathbf{a}}}$, and let $\mathbf{y}=\mathbf{x}\pmod{D}$. Note that as the distribution of $\mathbf{x}$ is uniform over $[0,1]^n$ and the distribution of $\mathbf{y}$ is uniform over D. Let $E_1(\vec{\mathbf{a}},\mathbf{x},\mathbf{\Delta}_1)$ be the event that $\mathbf{y}$ and $\mathbf{y}+\mathbf{\Delta}_1$ are in different cells of D. Then by the union bound and the choice of D

$$\begin{aligned}\Pr_{\vec{\mathbf{a}},\mathbf{x},\mathbf{\Delta}_1}[E_1]&=\Pr_{\vec{\mathbf{a}},\mathbf{x},\mathbf{\Delta}_\varepsilon}[\mathbf{y},\mathbf{y}+k\mathbf{\Delta}_\varepsilon\text{ in different cells of }D]\\&\leqslant\sum_{j=0}^{k-1}\Pr_{\mathbf{y},\mathbf{\Delta}_\varepsilon}[\mathbf{y}+j\mathbf{\Delta}_\varepsilon,\mathbf{y}+(j+1)\mathbf{\Delta}_\varepsilon\text{ in different cells of }D]\\&=\sum_{j=0}^{k-1}\Pr_{\mathbf{w}\in D,\mathbf{\Delta}_\varepsilon}[\mathbf{w},\mathbf{w}+\mathbf{\Delta}_\varepsilon\text{ in different cells of }D]\\&\leqslant\sum_{j=0}^{k-1}C\cdot A\cdot\frac{\varepsilon}{n}=C\frac{A}{n}.\end{aligned}$$

Let $E_2(\vec{\mathbf{a}})$ be the event that $B_{\vec{\mathbf{a}}}$ is decisive, and if $E_2(\vec{\mathbf{a}})$ holds let $\mathbf{z}\in\mathbb{Z}^n$ be such that $\mu(B_{\vec{\mathbf{a}}}\cap(D+\mathbf{z}))\geqslant\frac{2}{3}\mu(B_{\vec{\mathbf{a}}})$. Then by Lemma 56 $\Pr[E_2(\vec{\mathbf{a}})]\geqslant 1-O(A/n)$. Denote

$$p_{\vec{a},\Delta_1}=\Pr_{\mathbf{x},\mathbf{a},\mathbf{\Delta}_1}[E_1(\vec{\mathbf{a}},\mathbf{x},\mathbf{\Delta}_1)\mid\vec{\mathbf{a}}=\vec{a},\mathbf{\Delta}_1=\Delta_1].$$

The expectation of $p_{\vec{\mathbf{a}},\boldsymbol{\Delta}_1}$ is the probability of $E_1(\vec{\mathbf{a}},\mathbf{x},\boldsymbol{\Delta}_1)$, so

$$\begin{aligned}\Pr_{\vec{\mathbf{a}},\boldsymbol{\Delta}_1}\left[E_2(\vec{\mathbf{a}})\wedge p_{\vec{\mathbf{a}},\boldsymbol{\Delta}_1}\leqslant\frac{1}{10}\right]&\geqslant 1-\Pr_{\vec{\mathbf{a}}}\left[\overline{E_2(\vec{\mathbf{a}})}\right]-\Pr_{\vec{\mathbf{a}},\boldsymbol{\Delta}_1}\left[p_{\vec{\mathbf{a}},\boldsymbol{\Delta}_1}>\frac{1}{10}\right]\\&\geqslant 1-O\left(\frac{A}{n}\right)-\frac{\Pr_{\vec{\mathbf{a}},\mathbf{x},\boldsymbol{\Delta}_1}\left[E_1(\vec{\mathbf{a}},\mathbf{x},\boldsymbol{\Delta}_1)\right]}{1/10},\end{aligned}$$

which is at least $1-O\left(\frac{A}{n}\right)$. To finish the proof, we show that for every $\vec{a}$, Δ_1 such that $E_2(\vec{a})$ holds and $p_{\vec{a},\Delta_1}\leqslant\frac{1}{10}$, we have the the event on the left hand side of (17) holds.

Indeed, fix such $\vec{a}$, Δ_1. Then there is a unique $z\in\mathbb{Z}^n$ such that $\mu(B_{\vec{a}}\cap(D+z))=\Pr_{\mathbf{x}\in B_{\vec{a}}}[\mathbf{x}\in(D+z)]$ is at least $\frac{2}{3}\mu(B_{\vec{a}})$. Note that if $\mathbf{y}$, $\mathbf{y}+\Delta_1$ are in the same cell of D, then $\mathbf{x}$, $\mathbf{x}+\Delta_1$ are in the same cell of D, so

$$\begin{aligned}\frac{\mu(B_{\vec{a}+n\Delta_1}\cap(D+z))}{\mu(B_{\vec{a}+n\Delta_1})}&=\frac{\mu(B_{\vec{a}+n\Delta_1}\cap(D+z))}{\mu(B_{\vec{a}})}\\&=\Pr_{\mathbf{x}\in B_{\vec{a}}}\left[\mathbf{x}+\Delta_1\in(D+z)\right]\\&\geqslant\Pr_{\mathbf{x}\in B_{\vec{a}}}\left[\mathbf{x}\in(D+z),\mathbf{x}+\Delta_1\in(D+z)\right]\\&\geqslant\Pr_{\mathbf{x}\in B_{\vec{a}}}\left[\mathbf{x}\in(D+z)\text{ and }\mathbf{y},\mathbf{y}+\Delta_1\text{ in the same cell of }D\right]\\&\geqslant\Pr_{\mathbf{x}\in B_{\vec{a}}}\left[\mathbf{y},\mathbf{y}+\Delta_1\text{ in the same cell of }D\right]-\Pr_{\mathbf{x}\in B_{\vec{a}}}\left[\mathbf{x}\notin(D+z)\right]\\&=1-p_{\vec{a},\Delta_1}-\Pr_{\mathbf{x}\in B_{\vec{a}}}\left[\mathbf{x}\notin(D+z)\right]\\&\geqslant 1-\frac{1}{10}-\frac{1}{3}>\frac{1}{2}.\end{aligned}$$

◀

5.2.3 Proof of Theorem 8

In this section, we prove Theorem 8. For that, we show that the success probability of the following players' strategy is at least $1-O(A/n)$.

1. On challenge $x'\in C_n^t$, consider the box that x' belongs to, i.e. $B_{\vec{a}}$ for $\vec{a}=nx'$.
2. Check if there is $z\in\mathbb{Z}^t$ such that $\mu(B_{\vec{a}}\cap(D+z))>\frac{1}{2}\mu(B_{\vec{a}})$, and note that it is unique if such point exists. If there is no such z, abort. We refer to z as the chosen lattice point of the player.
3. Output $z+nx' \pmod 2$.

First, we argue that this strategy is symmetric. Indeed, the effect of permuting the entries of x' by $\pi\in S_t$ is that a,z above also get permuted by π, and therefore the output also gets permuted by π. Next, we analyze the success probability of this strategy.

Note the following equivalent way of picking challenges $(\mathbf{x}',\mathbf{y}')$: sample uniformly $\vec{\mathbf{a}}\in\{0,1,\ldots,n-1\}^t$, set $\mathbf{x}'=\vec{\mathbf{a}}/n$, sample $\boldsymbol{\Delta}_1$ Bernoulli as above and set $\mathbf{y}'=\mathbf{x}'+\boldsymbol{\Delta}_1 \pmod 1$. Denote the box of $\mathbf{x}'$ by $B_{\vec{a}(\mathbf{x}')}$, and consider the event E defined in Lemma 57. We show that whenever the event E holds, the players are successful with the above strategy, and as the probability of E is at least $1-O(A/n)$, the proof would be concluded.

Fix $\vec{a},\Delta_1$ such that E holds, and let $z\in\mathbb{Z}^t$ be the (unique) point such that $\mu(B_{\vec{a}}\cap(D+z))\geqslant\frac{2}{3}\mu(B_{\vec{a}})$, $\mu(B_{\vec{a}+n\Delta_1}\cap(D+z))>\frac{1}{2}\mu(B_{\vec{a}+n\Delta_1})$. The first condition implies that the x'-player does not abort and their chosen lattice point is z, and we next show that the y'-player does not abort as well. Note that the box of y' is $B_{\vec{a}(y')}$ for $\vec{a}(y')=\vec{a}+n\Delta_1 \pmod 1$, and write $\vec{a}+n\Delta_1=\vec{a}(y')+w$ for $w\in\mathbb{Z}^t$. Thus,

$$\mu(B_{\vec{a}(y')}\cap(D+z-w))=\mu(B_{\vec{a}(y')+w}\cap(D+z))=\mu(B_{\vec{a}+n\Delta_1}\cap(D+z))>\frac{1}{2}\mu(B_{\vec{a}+n\Delta_1}),$$

which is equal to $\frac{1}{2}\mu(B_{\vec{a}(y')})$, so the y'-player also does not abort and their chosen lattice point is $z-w$. We now analyze the answers of the players on each coordinate.

- If i is a coordinate such that $y'_i \neq x'_i$, then we may write $y'_i = x'_i + \Delta_1 + b$ for $b \in \{-1, 0, 1\}$ and $\Delta_1 \neq 0$. Then we get that $\vec{a}(y')_i = \vec{a}_i + n(\Delta_1)_i + nb$, so $w_i = -nb$. Thus, the answer of the x'-player is $z_i + nx'_i \pmod 2$, whereas the answer of the y'-player is

$$(z-w)_i + ny'_i = z_i + nb + nx'_i + n\Delta_1 + nb = z_i + nx'_i + n\Delta_1 + 2nb = z_i + n\mathbf{x}'_i + 1 \pmod 2,$$

where we used $2nb = 0 \pmod 2$, and $n\Delta_1 = 1 \pmod 2$ (as $\Delta_1 = \pm\frac{1}{n}$). Thus, the players are consistent on the ith coordinate.

- If i is a coordinate such that $y'_i = x'_i$, then in the above notations we have $w_i = 0$, $\Delta_i = 0$ and we get that the answers of the players are the same on the ith coordinate, so they are consistent on i. ◀

6 Open Problems

In this section, we propose several challenges for further investigation of symmetric parallel repetition.

Recall from the introduction that on general games a strong parallel repetition theorem still fails, even for symmetric repetition. A simple example is the union of many disjoint, odd cycle games. It would be interesting to understand for what instances of Max-Cut one has that a strong parallel holds with symmetric repetition, motivating the following problem.

▶ **Problem 1.** *For the Max-Cut problem, extend the family of graphs for which symmetric parallel repetition outperforms standard parallel repetition.*

Optimistically, one may hope that if symmetric parallel repetition would work for general enough class of graphs, then one would be able to reduce any graph to a graph in that class by mild preprocessing that doesn't affect the value of the game by much, and only then perform symmetric repetition. If possible, that would establish the equivalence of the Max-Cut Conjecture and UGC.

Secondly, there are well-known connections between parallel repetition and notions of mixing times and eigenvalues of the underlying graph; for example, a strong parallel repetition theorem is known to hold for expander graphs [31, 3], and more generally for graphs with low threshold rank [35], i.e. graphs with only constantly many eigenvalues close to 1. We expect there could be stronger relations between symmetric parallel repetition and higher order eigenvalues of $G^{\otimes_{\mathsf{sym}} k}$, the k-fold symmetric tensor product of G.

▶ **Problem 2.** *What is the relation between the performance of the k-fold symmetric parallel repetition of a given instance of Max-Cut G, and the first $k+1$ eigenvalues of G?*

Finally, we believe that solving the foam problem for special classes of bodies may be an interesting geometric question (albeit unrelated to the study of parallel repetition); a very natural class to study is the class of convex bodies.

References

1 Perfect sets are uncountable. [Online; accessed 15-April-2020]. URL: `https://mathcs.org/analysis/reals/topo/proofs/pfctuncb.html`.

2 Noga Alon and Bo'az Klartag. Economical toric spines via cheeger's inequality. *Journal of Topology and Analysis*, 1(02):101–111, 2009.

3 Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: extended abstract. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 21–28, 2008. `doi:10.1145/1374376.1374380`.

4 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

5 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

6 Boaz Barak, Moritz Hardt, Ishay Haviv, Anup Rao, Oded Regev, and David Steurer. Rounding parallel repetitions of unique games. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 374–383, 2008. doi:10.1109/FOCS.2008.55.

7 Boaz Barak, Pravesh K. Kothari, and David Steurer. Small-set expansion in shortcode graph and the 2-to-2 conjecture. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 9:1–9:12, 2019. doi:10.4230/LIPIcs.ITCS.2019.9.

8 Boaz Barak, Anup Rao, Ran Raz, Ricky Rosen, and Ronen Shaltiel. Strong parallel repetition theorem for free projection games. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*, pages 352–365, 2009. doi:10.1007/978-3-642-03685-9_27.

9 Amey Bhangale, Ramprasad Saptharishi, Girish Varma, and Rakesh Venkat. On fortification of projection games. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, pages 497–511, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.497.

10 Mark Braverman and Ankit Garg. Small value parallel repetition for general games. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 335–340, 2015. doi:10.1145/2746539.2746565.

11 Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. On non-optimally expanding sets in grassmann graphs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 940–951, 2018. doi:10.1145/3188745.3188806.

12 Irit Dinur, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Towards a proof of the 2-to-1 games conjecture? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 376–389, 2018. doi:10.1145/3188745.3188804.

13 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633, 2014. doi:10.1145/2591796.2591884.

14 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996. doi:10.1145/226643.226652.

15 Uriel Feige, Guy Kindler, and Ryan O'Donnell. Understanding parallel repetition requires understanding foams. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 179–192. IEEE, 2007.

16 Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theor. Comput. Sci.*, 134(2):545–557, 1994. doi:10.1016/0304-3975(94)90251-8.

17 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.

18 Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(1):141–172, 2009. doi:10.4086/toc.2009.v005a008.

19 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002*, page 25, 2002. doi:10.1109/CCC.2002.1004334.

20 Subhash Khot. Inapproximability of NP-complete problems, discrete fourier analysis, and geometry. In *Proceedings of the International Congress of Mathematicians 2010*, pages 2676–2697, 2010.

21 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007. doi:10.1137/S0097539705447372.

22 Subhash Khot, Dor Minzer, and Muli Safra. On independent sets, 2-to-2 games, and Grassmann graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 576–589, 2017. doi:10.1145/3055399.3055432.

23 Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 592–601, 2018. doi:10.1109/FOCS.2018.00062.

24 Guy Kindler, Anup Rao, Ryan O'Donnell, and Avi Wigderson. Spherical cubes: optimal foams from computational hardness amplification. *Commun. ACM*, 55(10):90–97, 2012. doi:10.1145/2347736.2347757.

25 Dana Moshkovitz. Parallel repetition from fortification. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 414–423, 2014. doi:10.1109/FOCS.2014.51.

26 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM Journal on Computing*, 40(6):1871–1891, 2011.

27 Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. doi:10.1137/S0097539795280895.

28 Ran Raz. A counterexample to strong parallel repetition. *SIAM Journal on Computing*, 40(3):771–777, 2011.

29 Antonio Ros. The isoperimetric problem. *Global Theory of Minimal Surfaces. Clay Math. Proc*, vol. 2:175–209, 2005.

30 Jean-François Sadoc and Nicolas Rivier. *Foams and emulsions*, volume 354. Springer Science & Business Media, 2013.

31 Shmuel Safra and Oded Schwartz. On parallel-repetition, Unique-Games and Max-Cut, 2007.

32 Luis Antonio Santaló Sors and Luis A Santaló. *Integral geometry and geometric probability*. Cambridge university press, 2004.

33 William Thomson. On the division of space with minimum partitional area. *Acta Math.*, 11:121–134, 1887. doi:10.1007/BF02612322.

34 Luca Trevisan. On Khot's unique games conjecture. *Bull. Amer. Math. Soc. (N.S.)*, 49(1):91–111, 2012. doi:10.1090/S0273-0979-2011-01361-1.

35 Madhur Tulsiani, John Wright, and Yuan Zhou. Optimal strong parallel repetition for projection games on low threshold rank graphs. In *International Colloquium on Automata, Languages, and Programming*, pages 1003–1014. Springer, 2014.

A Deferred proofs

A.1 Proof of Claim 30

We split the proof into two cases.

Case 1: $r_i \leqslant T/2$ for all i

In this case, $\min(r_i, T - r_i) = r_i$ for all i, and the sum on the RHS of (7) is just $(\sum_i |d_i|)/T$. We have

$$\begin{aligned}
\|p - q\|_1 = \sum_i \left|\frac{r_i + d_i}{T'} - \frac{r_i}{T}\right| &\leqslant \sum_i \left|\frac{r_i + d_i}{T} - \frac{r_i}{T}\right| + \sum_i \left|\frac{r_i + d_i}{T'} - \frac{r_i + d_i}{T}\right| \\
&= \sum_i \frac{|d_i|}{T} + \left|\frac{1}{T'} - \frac{1}{T}\right| \cdot \sum_i (r_i + d_i) \\
&= \sum_i \frac{|d_i|}{T} + \left|1 - \frac{T'}{T}\right| \\
&= \sum_i \frac{|d_i|}{T} + \frac{1}{T} \cdot \left|\sum_i d_i\right| \\
&\leqslant 2 \cdot \sum_i \frac{|d_i|}{T}.
\end{aligned}$$

Case 2: one of the r_i's is greater than $T/2$

Without loss of generality, $r_1 > T/2$. Denote by $S := \sum_{i>1} r_i = T - r_1$; $S' := \sum_{i>1}(r_i + d_i) = T' - r_1 - d_1$. In this case, the RHS of (7) is given by

$$\frac{|d_1| \cdot S}{r_1 \cdot T} + \sum_{i>1} \frac{|d_i|}{T}. \tag{18}$$

We will estimate $|p_1 - q_1|$ and $\sum_{i>1} |p_i - q_i|$ separately. First, note that $T' \geqslant T - \sum_j |d_j| \geqslant T/2$.

For $|p_1 - q_1|$, we have

$$\begin{aligned}
|p_1 - q_1| = \left|\frac{r_1}{T} - \frac{r_1 + d_1}{T'}\right| = \left|\frac{r_1 \cdot (S' - S) + d_1 \cdot S}{T \cdot T'}\right| &\leqslant 2\left|\frac{S' - S}{T}\right| + 2\left|\frac{d_1 \cdot S}{T \cdot r_1}\right| \\
&\leqslant \sum_{i>1} \frac{|d_i|}{T} + \frac{|d_1| \cdot S}{r_1 \cdot T}.
\end{aligned}$$

In the third transition, we used the fact that $T' \geqslant T/2 \geqslant r_1/2$.

For $\sum_{i>1} |p_i - q_i|$, by a similar calculation to the first case we have

$$\begin{aligned}
\sum_{i>1} |p_i - q_i| = \sum_{i>1} \left|\frac{r_i + d_i}{T'} - \frac{r_i}{T}\right| &\leqslant \sum_{i>1} \left|\frac{r_i + d_i}{T} - \frac{r_i}{T}\right| + \sum_{i>1} \left|\frac{r_i + d_i}{T'} - \frac{r_i + d_i}{T}\right| \\
&\leqslant \sum_{i>1} \frac{|d_i|}{T} + \left|\frac{1}{T'} - \frac{1}{T}\right| \cdot \sum_{i>1} r_i + d_i \\
&= \sum_{i>1} \frac{|d_i|}{T} + \left|\frac{1}{T'} - \frac{1}{T}\right| S',
\end{aligned}$$

and it is enough to bound $\left|\frac{1}{T'}-\frac{1}{T}\right|S'$ by constant times the expression in (18). We have

$$\left|\frac{1}{T'}-\frac{1}{T}\right|S' = \left|\frac{S'\cdot(S'-S)+S'\cdot d_1}{T'T}\right| \leqslant \left|\frac{(S'+d_1)\cdot(S'-S)}{T'T}\right| + \left|\frac{S\cdot d_1}{T'T}\right|$$
$$\leqslant \left|\frac{S'\cdot(S'-S)}{T'T}\right| + \left|\frac{d_1\cdot(S'-S)}{T'T}\right| + 2\cdot\left|\frac{S\cdot d_1}{T^2}\right|,$$

where in the last transition we used $T' \geqslant T/2 > 0$. We bound each term separately. For the first term, as $T' \geqslant T/2$, $|S'| \leqslant 2T$ (since $|d_i| \leqslant r_i$) we get

$$\left|\frac{S'\cdot(S'-S)}{T'T}\right| \leqslant 4\left|\frac{S'-S}{T}\right| \leqslant 4\sum_{i\geqslant 2}\frac{|d_i|}{T}.$$

For the second term, we have $|d_1| \leqslant r_1 \leqslant T$, $T' \geqslant T/2$ and so

$$\left|\frac{d_1\cdot(S'-S)}{T'T}\right| \leqslant 2\frac{|S'-S|}{T} \leqslant 2\sum_{i\geqslant 2}\frac{|d_i|}{T}.$$

For the third term, we have, as $T \geqslant r_1$, $\frac{S\cdot d_1}{T^2} \leqslant \frac{|d_1|}{r_1}\frac{S}{T}$.

A.2 Proof of Proposition 33

We will use the fact for points x_i in our domain, $g_j(x_i) \asymp \left(\frac{n}{\log n}\alpha_i\right)^3$. We consider two cases, based on the values of S and r.

Case 1: $\Pr_{\mathbf{x}_i}[r\cdot g_j(\mathbf{x}_i) > S] < 1/2$

We claim that for a sufficiently large constant $A > 0$,

$$\underbrace{\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\sqrt{z+\frac{1}{\boldsymbol{\alpha}_i^2}}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]}_{(I)} \lesssim \underbrace{\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\sqrt{z+An^2/\log^2 n}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]}_{(II)}.$$

To do that, we compare both sides to $\mathbb{E}_{\mathbf{x}_i}\left[\sqrt{z}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]$. For (I), we have

$$\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\left(\sqrt{z+1/\boldsymbol{\alpha}_i^2}-\sqrt{z}\right)\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]$$
$$\lesssim \mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\frac{1/\boldsymbol{\alpha}_i^2}{\sqrt{z+1/\boldsymbol{\alpha}_i^2}}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right].$$

Since $\boldsymbol{\alpha}_i \lesssim \log n/n$ always, we may further upper bound this by

$$\lesssim \mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\frac{1/\boldsymbol{\alpha}_i^2}{\sqrt{z+A/(\log n/n)^2}}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right] \lesssim \mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\frac{1/\boldsymbol{\alpha}_i^2}{\sqrt{z+An^2/\log^2 n}}\cdot\frac{r\left(\frac{n}{\log n}\boldsymbol{\alpha}_i\right)^3}{S+\varepsilon^{1.6}}\right],$$

where we used $\min(r\cdot g_j(\mathbf{x}_i),S) \leqslant rg_j(\mathbf{x}_i)$ and the asymptotic we have for g_j. Simplifying and using $\mathbb{E}_{\mathbf{x}_i}[\boldsymbol{\alpha}_i] \lesssim \log n/n$, we get that the last expression is equal to

$$\frac{n^2}{\log^2 n}\frac{1}{\sqrt{z+An^2/\log^2 n}}\cdot\frac{r}{S+\varepsilon^{1.6}}.$$

For (II), we have

$$\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\left(\sqrt{z+An^2/\log^2 n}-\sqrt{z}\right)\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]$$
$$\gtrsim \mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\frac{An^2/\log^2 n}{\sqrt{z+An^2/\log^2 n}}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right].$$

Restricting to the event E that $rg_j(\mathbf{x}_i)\leqslant S$ (that has probability at least 1/2 by assumption), we have that the last expression is at least

$$\gtrsim \mathbb{E}_{\mathbf{x}_i}\left[\frac{An^2/\log^2 n}{\sqrt{z+An^2/\log^2 n}}\cdot\frac{r\cdot g_j(\mathbf{x}_i)}{S+\varepsilon^{1.6}}\,\middle|\,E\right]\gtrsim\frac{An^2/\log^2 n}{\sqrt{z+An^2/\log^2 n}}\cdot\frac{r}{S+\varepsilon^{1.6}},$$

where the last inequality holds since $\mathbb{E}_{\alpha_i}\left[g_j(x_i)\mid E\right]\gtrsim 1$ (this is true for any event E with constant probability in our range of interest of $\mathbf{x}_i$'s). Combining the bounds for $(I),(II)$, we see that we may pick large enough A so that

$$\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\left(\sqrt{z+1/\alpha_i^2}-\sqrt{z}\right)\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]$$
$$\leqslant \mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\left(\sqrt{z+An^2/\log^2 n}-\sqrt{z}\right)\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right],$$

and hence $(I)\leqslant(II)$. Let A_1 be a large enough value of A so that this holds.

Case 2: $\Pr_{\mathbf{x}_i}\left[r\cdot g_j(\mathbf{x}_i)>S\right]\geqslant 1/2$

Using $\sqrt{a+b}\leqslant\sqrt{a}+\sqrt{b}$, we have

$$\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\sqrt{z+\frac{1}{\boldsymbol{\alpha}_i^2}}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]$$
$$\leqslant\underbrace{\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\sqrt{z}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]}_{(III)}+\underbrace{\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\frac{1}{\boldsymbol{\alpha}_i}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]}_{(IV)}.$$

Clearly, $(III)\leqslant\mathbb{E}_{\mathbf{x}_i}\left[\sqrt{z+A\frac{n^2}{\log^2 n}}\cdot\frac{\min(r\cdot g_j(\mathbf{x}_i),S)}{r\cdot g_j(\mathbf{x}_i)+S+\varepsilon^{1.6}}\right]$, and we upper bound (IV). Recall that $g_j(\mathbf{x}_i)\asymp\left(\frac{n}{\log n}\boldsymbol{\alpha}_i\right)^3$, so

$$(IV)\lesssim\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[\frac{1}{\boldsymbol{\alpha}_i}\cdot\frac{\min(r(n\boldsymbol{\alpha}_i/\log n)^3,S)}{B\cdot r(n\boldsymbol{\alpha}_i/\log n)^3+S+\varepsilon^{1.6}}\right],$$

for some absolute constant $B>0$. Writing the last expression as an integral, we note that $\boldsymbol{\alpha}_i$ is distributed uniformly on the interval $[0,\frac{\log n}{50n}+\varepsilon^{0.95}]$, so we get

$$(IV)\lesssim\left(\frac{n}{\log n}\right)\int_0^{\frac{\log n}{25n}}\frac{1}{t}\frac{\min(r(nt/\log n)^3,S)}{B\cdot r(nt/\log n)^3+S+\varepsilon^{1.6}}dt.$$

We break the range of integration into $R_1=\left[0,(S/r)^{1/3}\frac{\log n}{n}\right]$, and $R_2=\left[(S/r)^{1/3}\frac{\log n}{n},\frac{\log n}{25n}\right]$. On R_1 our expression is equal to

$$\left(\frac{n}{\log n}\right)^2\int_0^{\left(\frac{S}{r}\right)^{1/3}\frac{\log n}{n}}\frac{r(nt/\log n)^2}{B\cdot r(nt/\log n)^3+S+\varepsilon^{1.6}}dt\lesssim\left(\frac{n}{\log n}\right)^4\int_0^{\left(\frac{S}{r}\right)^{1/3}\frac{\log n}{n}}\frac{rt^2}{S}dt$$
$$\lesssim\frac{n}{\log n}.$$

On R_2 our expression is at most

$$\left(\frac{n}{\log n}\right)\int_{(\frac{S}{r})^{1/3}\frac{\log n}{n}}^{\frac{\log n}{25n}} \frac{1}{t}\frac{S}{B\cdot r(nt/\log n)^3}dt \lesssim \frac{S}{r}\left(\frac{\log n}{n}\right)^2 \int_{(\frac{S}{r})^{1/3}\frac{\log n}{n}}^{\frac{\log n}{25n}} \frac{1}{t^4}dt.$$

Computing the integral, we see it is at most $((\frac{S}{r})^{1/3}\frac{\log n}{n})^{-3}$, hence the overall expression is $\lesssim n/\log n$, and since $\mathbb{E}\left[1_{r\cdot g_j(x_i)>S}\right] \geqslant 1/2$ we conclude that there is $A_2 > 0$ such that

$$(IV) \leqslant A_2\frac{n}{\log n}\mathop{\mathbb{E}}_{\mathbf{x}_i}\left[1_{r\cdot g_j(\mathbf{x}_i)>S}\right].$$

The proposition is thus proven for $A = \max(A_1, A_2)$.

B From Noise Sensitivity to Surface Area

Let $D_{\vec{r}}$ be a family of tilings of $\mathbb{R}^n$ that are constructed from Lemma 24. I.e., the family $D_{\vec{r}}$ satisfies that the there is $A = O(n/\sqrt{\log n})$ such that for sufficiently small ε, we have that

$$\mathop{\mathbb{E}}_{\vec{r}}\left[\Pr_{\substack{\mathbf{x}\in D_{\vec{r}}\\ \mathbf{\Delta}\sim N(0,\varepsilon^2\cdot I_n)}}[\mathbf{x},\mathbf{x}+\mathbf{\Delta} \text{ fall in different cells of the tiling induced by } D_{\vec{r}}]\right] \leqslant A\varepsilon.$$

Let k_0 be the first integer such that this condition holds for any $0 < \varepsilon \leqslant 2^{-k_0}$. Thus, defining for each $k \geqslant k_0$ the set

$$G_k = \left\{\vec{r} \;\middle|\; \Pr_{\substack{\mathbf{x}\in D_{\vec{r}}\\ \mathbf{\Delta}\sim N(0,4^{-k}\cdot I_n)}}[\mathbf{x},\mathbf{x}+\mathbf{\Delta} \text{ lie in different cells of the tiling of } S_{\vec{r}}] \leqslant 2\cdot A2^{-k}\right\},$$

we have by Markov's inequality that $\Pr_{\vec{r}}[\vec{r}\in G_k] \geqslant \frac{1}{2}$.

▷ Claim 58. The sets G_k are monotone decreasing, i.e. for each k, $G_{k+1} \subseteq G_k$.

Proof. Fix $\vec{r}\in G_{k+1}$. Let $\Delta \sim N(0, 4^{-k-1}\cdot I_n)$, and note that $\Delta' = 2\cdot\Delta \sim N(0, 4^{-k}\cdot I_n)$. Thus,

$$\begin{aligned}
&\Pr_{\substack{\mathbf{x}\in D_{\vec{r}}\\ \mathbf{\Delta'}\sim N(0,4^{-k}\cdot I_n)}}[\mathbf{x},\mathbf{x}+\mathbf{\Delta'} \text{ in different cells}]\\
&\quad\leqslant \Pr_{\substack{\mathbf{x}\in D_{\vec{r}}\\ \mathbf{\Delta}\sim N(0,A4^{-k-1}\cdot I_n)}}[\mathbf{x},\mathbf{x}+\mathbf{\Delta} \text{ in different cells}]\\
&\quad+ \Pr_{\substack{\mathbf{x}\in D_{\vec{r}}\\ \mathbf{\Delta}\sim N(0,4^{-k-1}\cdot I_n)}}[\mathbf{x}+\mathbf{\Delta},\mathbf{x}+2\mathbf{\Delta} \text{ in different cells}]. \qquad (19)
\end{aligned}$$

First, we argue that the second probability on the right hand side is equal to the first one. To see that, denote $y = x + \Delta$ and observe that the points $y, y+\Delta$ lie in different cells of the tiling induced by $D_{\vec{r}}$ if and only if the points $y \pmod{D_{\vec{r}}}$, $y \pmod{D_{\vec{r}}} + \Delta$ lie in different cells. Additionally, note for any fixed Δ, the distribution of $\mathbf{y} \pmod{D_{\vec{r}}}$ when we take $\mathbf{x}\in_R D_{\vec{r}}$, is uniform over $D_{\vec{r}}$.

Therefore, the bound we get from (19) is (using the fact that $\vec{r}\in G_{k+1}$)

$$2\cdot \Pr_{\substack{\mathbf{x}\in D_{\vec{r}}\\ \mathbf{\Delta}\sim N(0,4^{-k-1}\cdot I_n)}}[\mathbf{x},\mathbf{x}+\mathbf{\Delta} \text{ in different cells}] \leqslant 2\cdot 2\cdot A2^{-(k+1)} = 2\cdot A2^{-k},$$

and so $\vec{r}\in G_k$. ◁

▷ Claim 59. It holds that $\Pr_{\vec{r}}\left[\vec{r} \in \bigcap_{k\geqslant k_0} G_k\right] \geqslant \frac{1}{2}$, and in particular $\bigcap_{k\geqslant k_0} G_k$ is not empty.

Proof. Define the sequence of functions $g_m(\vec{r}) = 1_{\vec{r}\in\bigcap_{k_0\leqslant k\leqslant m} G_k}$, and also $g = 1_{\vec{r}\in\bigcap_{k\geqslant k_0} G_k}$. Clearly, on each $\vec{r}$, the sequence $g_m(\vec{r})$ is monotonically decreasing to $g(\vec{r})$, and in other words we have monotone pointwise convergence of the non-negative functions g_m to g. Thus, by the monotone convergence theorem

$$\Pr_{\vec{r}}\left[\vec{r} \in \bigcap_{k\geqslant 0} G_k\right] = \mathop{\mathbb{E}}_{\vec{r}}\left[g(\vec{r})\right] = \mathop{\mathbb{E}}_{\vec{r}}\left[\lim_{k\to\infty} g_k(\vec{r})\right] = \lim_{k\to\infty}\mathop{\mathbb{E}}_{\vec{r}}\left[g_k(\vec{r})\right].$$

By the previous claim, $g_m = 1_{G_m}$, hence $\mathbb{E}_{\vec{r}}\left[g_m(\vec{r})\right] \geqslant \frac{1}{2}$ and in particular the limit above is at least $\frac{1}{2}$. ◁

Pick $\vec{r}^\star \in \bigcap_{k\geqslant k_0} G_k$, $\varepsilon = 2^{-k_0}$ and denote $D = D_{\vec{r}^\star}$ for the rest of the proof. Clearly D induces a tiling of the space $\mathbb{R}^n$, and next we will show that the surface area of D is $O(A) = O(n/\sqrt{\log n})$, as desired.

Towards this end, we will use Lemma 10 that tells us that the surface area of D is a constant multiple of

$$\frac{1}{\varepsilon}\mathop{\mathbb{E}}_{\substack{\mathbf{x}\in_R D\\ \mathbf{\Delta}\sim N(0,\varepsilon^2 I_n)}}\left[\left|(\mathbf{x},\mathbf{x}+\mathbf{\Delta})\cap\partial D\right|\right],$$

and we first observe that $(\mathbf{x},\mathbf{x}+\mathbf{\Delta})\cap\partial D$ is almost surely countable. [4]

▷ Claim 60. Let $\varepsilon > 0$ and sample $\mathbf{x} \in_R D$, $\mathbf{\Delta} \sim N(0,\varepsilon^2 I_n)$. Then with probability 1, $(\mathbf{x},\mathbf{x}+\mathbf{\Delta})\cap\partial D$ is finite or countable.

Proof. Recall that by Lemma 24, D is a countable union of semi-algebraic sets, say $B_1, B_2, \ldots$. Note that for each semi-algebraic set B_i, the probability that $(\mathbf{x},\mathbf{x}+\mathbf{\Delta})\cap\partial B_i$ is infinite is 0, hence by the union bound, with probability 1 all of these sets are finite, in which case $(\mathbf{x},\mathbf{x}+\mathbf{\Delta})\cap\partial D$ is finite or countable. ◁

For a parameter h, a point $x\in\mathbb{R}^n$ and a direction Δ, we say a point $y\in(x,x+\Delta)$ is h-isolated if

1. It holds that $y\in\partial D$.
2. The neighbourhood of radius h around y does not contain $x, x+\Delta$ or any point from ∂D (besides y).

Define the quantity $g_m(x,\Delta)$ to be the number of $2^{-m}\|\Delta\|_2$-isolated points in the interval $[x,x+\Delta]$.

▷ Claim 61. $g_m(x,\Delta)$ is an increasing sequence in m, and for any x,Δ for which Claim 60 holds, we have

$$\lim_{m\to\infty} g_m(x,\Delta) = \left|(x,x+\Delta)\cap\partial D\right|.$$

Proof. The monotonicity of $g_m(x,\Delta)$ in m, and $g_m(x,\Delta)\leqslant\left|(x,x+\Delta)\cap\partial D\right|$ are clear. We set $\ell = g_m(x,\Delta)$ and split the rest of the proof according to whether ℓ is finite or not.

[4] The diligent reader may note that here, we are only considering intersections of the surface with the open interval $(x,x+\Delta)$ as opposed to the closed interval. This does not make any difference, since the contribution of the endpoints is proportional to the measure of ∂D. Hence, if the measure of ∂D is 0 they endpoints contribute 0 to that expectation, and if the measure of ∂D is positive, then the expectation is infinite either way.

Case 1: ℓ is finite

In this case we argue that $g_m(x,\Delta) = |(x,x+\Delta)\cap\partial D|$ for large enough m. To see that, let $y_1,\ldots,y_\ell \in (x,x+\Delta)$ be all of the intersection points of $(x,x+\Delta)$ and ∂D, and take large enough m so that $2^{-m}\|\Delta\|_2$ is smaller than all of the distances $\|y_i - y_j\|_2$, $\|y_i - x\|_2$, $\|y_i - (x+\Delta)\|_2$ for all i and j.

Case 1: ℓ is infinite

Consider the set $S = [x,x+\Delta]\cap\partial D$, and note that it is a closed. By Claim 60, S is countable, and we argue that S must have an isolated point. Otherwise, S is a closed set and has no isolated point, i.e. it s a perfect set, but then it must be uncountable (e.g. see [1]). We thus conclude that S has an isolated point w_1; we may remove it from S, have that the resulting set is again closed and countable, so we may again find an isolated point. Repeating this argument, for any $v\in\mathbb{N}$ we may find a collection of isolated points $w_1,\ldots,w_v \in S$ that are all different from x and $x+\Delta$. As in case 1, we conclude that $g_m(x,\Delta)\geqslant v$ for large enough m, and since it holds for any v we conclude that $\lim_{m\to\infty} g_m(x,\Delta) = \infty$. ◁

By Lemma 10, we have that the surface area of D is at most a constant multiple of

$$\frac{1}{\varepsilon}\mathop{\mathbb{E}}_{\substack{\mathbf{x}\in_R D\\ \mathbf{\Delta}\sim N(0,\varepsilon^2 I_n)}}\left[|(\mathbf{x},\mathbf{x}+\mathbf{\Delta})\cap\partial D|\right] = \frac{1}{\varepsilon}\mathop{\mathbb{E}}_{\substack{\mathbf{x}\in_R D\\ \mathbf{\Delta}\sim N(0,\varepsilon^2 I_n)}}\left[\lim_{m\to\infty} g_m(\mathbf{x},\mathbf{\Delta})\right]$$
$$= \lim_{m\to\infty}\frac{1}{\varepsilon}\mathop{\mathbb{E}}_{\substack{\mathbf{x}\in_R D\\ \mathbf{\Delta}\sim N(0,\varepsilon^2 I_n)}}\left[g_m(\mathbf{x},\mathbf{\Delta})\right].$$

In the first transition we used Claims 61 and 60, and in the second one we used monotone convergence. Thus, if we assume that the surface area of D is larger than $c\cdot A$ for a sufficiently large absolute constant c, then we get that $\lim_{m\to\infty}\frac{1}{\varepsilon}\mathop{\mathbb{E}}_{\substack{x\in_R D\\ \Delta\sim N(0,\varepsilon^2 I_n)}}[g_m(x,\Delta)]\geqslant 10A$. In the rest of the proof we will reach a contradiction and thereby show that for sufficiently large absolute constant c, the surface area of D is at most cA, as required.

By properties of limits, we conclude there exists m such that

$$\mathop{\mathbb{E}}_{\substack{\mathbf{x}\in_R D\\ \mathbf{\Delta}\sim N(0,\varepsilon^2 I_n)}}\left[g_m(\mathbf{x},\mathbf{\Delta})\right]\geqslant 5A\varepsilon, \tag{20}$$

and we fix this m henceforth.

Take $0<\delta\leqslant 2^{-m}$, and consider the following experiment. Take $\mathbf{x}\in_R D$ uniformly at random, $\mathbf{\Delta}\sim N(0,\varepsilon^2 I_n)$ and take a uniformly random point $\mathbf{y}\in_R[\mathbf{x},\mathbf{x}+\mathbf{\Delta}]$. We consider the event E in which the points $\mathbf{y}$ and $\mathbf{y}+\delta\mathbf{\Delta}$ lie in different cells in the tiling induced by D.

▷ Claim 62. For any x,Δ we have that $\Pr_{\mathbf{y}}[E\mid x,\Delta]\geqslant\delta g_m(x,\Delta)$.

Proof. Let $\ell = g_m(x,\Delta)$, and let $z_1,\ldots,z_\ell$ be the $2^{-m}\|\Delta\|_2$-isolated points on the interval $(x,x+\Delta)$. For each j, let $I_j = (z_j-\delta\Delta, z_j)$, and note that as $\delta\leqslant 2^{-m}$ and the isolation of the points, we conclude that the intervals I_j are disjoint and contained in $(x,x+\Delta)$. Also, note that if we pick $y\in I_j$, then y and $y+\delta\Delta$ lie in different cells of the tiling induced by D; this holds since the interval between them contains exactly one point from ∂D (namely, the point z_j). Therefore,

$$\Pr_{\mathbf{y}}[E\mid x,\Delta]\geqslant\sum_{j=1}^{\ell}\Pr_{\mathbf{y}}[\mathbf{y}\in I_j\mid x,\Delta]\geqslant\sum_{j=1}^{\ell}\frac{\delta\|\Delta\|_2}{\|\Delta\|_2} = \delta\ell. \qquad ◁$$

▷ Claim 63. $\Pr_{\mathbf{x},\boldsymbol{\Delta},\mathbf{y}}[E] \leqslant 2A\delta\varepsilon$.

Proof. Consider $\mathbf{x}, \boldsymbol{\Delta}, \mathbf{y}$ the random variables in the definition of the event E. Let $\mathbf{z} = \mathbf{y} \pmod{D}$, and note that the points $\mathbf{y}$ and $\mathbf{y} + \delta\boldsymbol{\Delta}$ fall in different cells if and only if the points $\mathbf{z}$ and $\mathbf{z} + \delta\boldsymbol{\Delta}$ fall in different cells. Therefore, the probability of E is exactly the probability that $\mathbf{z}$, $\mathbf{z} + \delta\boldsymbol{\Delta}$ fall in different cells. Further, note that conditioned on $\boldsymbol{\Delta}$, the distribution of $\mathbf{z}$ is uniform over D, so

$$\begin{aligned}&\Pr_{\boldsymbol{\Delta}\sim N(0,\varepsilon^2 I_n)}[\mathbf{z}, \mathbf{z} + \delta\boldsymbol{\Delta} \text{ lie in different cells of } D]\\ &= \Pr_{\boldsymbol{\Delta}'\sim N(0,\delta^2\varepsilon^2 I_n)}[\mathbf{z}, \mathbf{z} + \boldsymbol{\Delta}' \text{ lie in different cells of } D],\end{aligned}$$

which is at most $2A\delta\varepsilon$ by the choice of D and the fact that $\delta\varepsilon \leqslant \varepsilon \leqslant 2^{-k_0}$. ◁

Combining the above claims we reach a contradiction:

$$2A\delta\varepsilon \geqslant \Pr_{\mathbf{x},\boldsymbol{\Delta},\mathbf{y}}[E] = \mathop{\mathbb{E}}_{\mathbf{x},\boldsymbol{\Delta}}\left[\Pr_{\mathbf{y}}[E \mid \mathbf{x}, \boldsymbol{\Delta}]\right] \geqslant \mathop{\mathbb{E}}_{\mathbf{x},\boldsymbol{\Delta}}[\delta g_m(\mathbf{x}, \boldsymbol{\Delta})] \geqslant \delta \cdot 5A\varepsilon,$$

and contradiction. The first transition is by Claim 62, the second transition is by conditional probability formula, the third transition is by Claim 63 and the final one is by equation (20).

On the Power and Limitations of Branch and Cut

Noah Fleming
University of Toronto, Canada
Simons Institute, Berkeley, CA, USA

Mika Göös
EPFL, Lausanne, Switzerland

Russell Impagliazzo
University of California, San Diego, CA, USA

Toniann Pitassi
University of Toronto, Canada
IAS, Princeton, NJ, USA

Robert Robere
McGill University, Montréal, Canada

Li-Yang Tan
Standford University, CA, USA

Avi Wigderson
IAS, Princeton, NJ, USA

Abstract

The Stabbing Planes proof system [8] was introduced to model the reasoning carried out in practical mixed integer programming solvers. As a proof system, it is powerful enough to simulate Cutting Planes and to refute the Tseitin formulas – certain unsatisfiable systems of linear equations mod2 – which are canonical hard examples for many algebraic proof systems. In a recent (and surprising) result, Dadush and Tiwari [25] showed that these short refutations of the Tseitin formulas could be translated into quasi-polynomial size and depth Cutting Planes proofs, refuting a long-standing conjecture. This translation raises several interesting questions. First, whether all Stabbing Planes proofs can be efficiently simulated by Cutting Planes. This would allow for the substantial analysis done on the Cutting Planes system to be lifted to practical mixed integer programming solvers. Second, whether the quasi-polynomial depth of these proofs is inherent to Cutting Planes.

In this paper we make progress towards answering both of these questions. First, we show that *any* Stabbing Planes proof with bounded coefficients (SP^*) can be translated into Cutting Planes. As a consequence of the known lower bounds for Cutting Planes, this establishes the first exponential lower bounds on SP^*. Using this translation, we extend the result of Dadush and Tiwari to show that Cutting Planes has short refutations of any unsatisfiable system of linear equations over a finite field. Like the Cutting Planes proofs of Dadush and Tiwari, our refutations also incur a quasi-polynomial blow-up in depth, and we conjecture that this is inherent. As a step towards this conjecture, we develop a new *geometric* technique for proving lower bounds on the depth of Cutting Planes proofs. This allows us to establish the first lower bounds on the depth of *Semantic* Cutting Planes proofs of the Tseitin formulas.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases Proof Complexity, Integer Programming, Cutting Planes, Branch and Cut, Stabbing Planes

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.6

Funding *Russell Impagliazzo*: Research supported by NSF and the Simons Foundation.
Toniann Pitassi: Research supported by NSERC, NSF Grant No. CCF-1900460 and the IAS school of mathematics.
Robert Robere: Research supported by the Charles Simonyi Endowment, and indirectly supported by the National Science Foundation Grant No. CCF-1900460.
Li-Yang Tan: Research supported by NSF grant CCF-192179 and NSF CAREER award CCF-1942123.
Avi Wigderson: Research partially supported by NSF grant CCF-1900460.

Acknowledgements N.F. would like to thank Albert Atserias for some corrections to an earlier version of this paper.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 6; pp. 6:1–6:30
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

An effective method for analyzing classes of algorithms is to formalize the techniques used by the class into a *formal proof system*, and then analyze the formal proof system instead. By doing this, theorists are able to hide many of the practical details of implementing these algorithms, while preserving the class of methods that the algorithms can feasibly employ. Indeed, this approach has been applied to study many different families of algorithms, such as

- *Conflict-driven clause-learning* algorithms for SAT [41, 49, 61], which can be formalized using *resolution* proofs [27].
- Optimization algorithms using *semidefinite programming* [34, 51], which can often be formalized using *Sums-of-Squares proofs* [6, 38].
- The classic *cutting planes* algorithms for integer programming [18,35], which are formalized by *cutting planes proofs* [18, 19, 23].

In the present work, we continue the study of formal proof systems corresponding to modern integer programming algorithms. Recall that in the integer programming problem, we are given a polytope $P \subseteq \mathbb{R}^n$ and a vector $c \in \mathbb{R}^n$, and our goal is to find a point $x \in P \cap \mathbb{Z}^n$ maximizing $c \cdot x$. The classic approach to solving this problem – pioneered by Gomory [35] – is to add[1] *cutting planes* to P. A *cutting plane* for P is any inequality of the form $ax \leq \lfloor b \rfloor$, where a is an integral vector, b is rational, and *every* point of P is satisfied by $ax \leq b$. By the integrality of a, it follows that cutting planes *preserve* the integral points of P, while potentially *removing* non-integral points from P. The cutting planes algorithms then proceed by heuristically choosing "good" cutting planes to add to P to try and locate the integral hull of P as quickly as possible.

As mentioned above, these algorithms can be naturally formalized into a proof system – the *Cutting Planes proof system*, denoted CP – as follows [23]. Initially, we are given a polytope P, presented as a list of integer-linear inequalities $\{a_i x \leq b_i\}$. From these inequalities we can then deduce new inequalities using two deduction rules:

- *Linear Combination.* From inequalities $ax \leq b, cx \leq d$, deduce any non-negative linear combination of these two inequalities with integer coefficients.
- *Division Rule.* From an inequality $ax \leq b$, if $d \in \mathbb{Z}$ with $d \geq 0$ divides all entries of a then deduce $(a/d)x \leq \lfloor b/d \rfloor$.

A Cutting Planes *refutation* of P is a proof of the trivially false inequality $1 \leq 0$ from the inequalities in P; clearly, such a refutation is possible only if P does not contain any integral points. While Cutting Planes has grown to be an influential proof system in propositional proof complexity, the original cutting planes algorithms suffered from numerical instabilities, as well as difficulties in finding good heuristics for the next cutting planes to add [35].

The modern algorithms in integer programming improve on the classical cutting planes method by combining them with a second technique, known as *branch-and-bound*, resulting in a family of optimization algorithms broadly referred to as *branch-and-cut algorithms*. These algorithms search for integer solutions in a polytope P by recursively repeating the following two procedures: First, P is split into smaller polytopes $P_1, \ldots, P_k$ such that $P \cap \mathbb{Z}^n \subseteq \bigcup_{i \in [k]} P_i$ (i.e. *branching*). Next, cutting planes deductions are made in order to further refine the branched polytopes (i.e. *cutting*). In practice, branching is usually performed by selecting a variable x_i and branching on all possible values of x_i; that is, recursing on $P \cap \{x_i = t\}$ for each feasible integer value t. More complicated branching

[1] Throughout, we will say that a cutting plane, or an inequality is *added* to a polytope P to mean that it is added to the set of inequalities defining P.

schemes have also been considered, such as branching on the hamming weight of subsets of variables [31], branching using basis-reduction techniques [1,2,45], and more general linear inequalities [42,47,50].

However, while these branch-and-cut algorithms are much more efficient in practice than the classical cutting planes methods, they are no longer naturally modelled by Cutting Planes proofs. So, in order to model these solvers as proof systems, Beame et al. [8] introduced the *Stabbing Planes* proof system. Given a polytope P containing no integral points, a *Stabbing Planes* refutation of P proceeds as follows. We begin by choosing an integral vector a, an integer b, and replacing P with the two polytopes $P \cap \{ax \leq b-1\}$ and $P \cap \{ax \geq b\}$. Then, we recurse on these two polytopes, continuing until all descendant polytopes are empty (that is, they do not even contain any *real* solutions). The majority of branching schemes used in practical branch-and-cut algorithms (including all of the concrete schemes mentioned above) are examples of this general branching rule.

It is now an interesting question how the two proof systems – Cutting Planes and Stabbing Planes – are related. By contrasting the two systems we see at least three major differences:

- *Top-down vs. Bottom-up.* Stabbing Planes is a *top-down* proof system, formed by performing queries on the polytope and recursing; while Cutting Planes is a *bottom-up* proof system, formed by deducing new inequalities from old ones.
- *Polytopes vs. Halfspaces.* Individual "lines" in a Stabbing Planes proof are *polytopes*, while individual "lines" in a Cutting Planes proof are *halfspaces*.
- *Tree-like vs. DAG-like.* The graphs underlying Stabbing Planes proofs are trees, while the graphs underlying Cutting Planes proofs are general DAGs: intuitively, this means that Cutting Planes proofs can "re-use" their intermediate steps, while Stabbing Planes proofs cannot.

When taken together, these facts suggest that Stabbing Planes and Cutting Planes could be incomparable in power, as polytopes are more expressive than halfspaces, while DAG-like proofs offer the power of line-reuse. Going against this natural intuition, Beame et al. proved that Stabbing Planes *can* actually efficiently simulate Cutting Planes [8] (see Figure 1). Furthermore, they proved that Stabbing Planes is *equivalent* to the proof system *tree-like* $\mathsf{R(CP)}$, denoted $\mathsf{treeR(CP)}$, which was introduced by Krajíček [44], and whose relationship to Cutting Planes was previously unknown.

This leaves the converse problem – of whether Stabbing Planes can also be simulated by Cutting Planes – as an intriguing open question. Beame et al. conjectured that such a simulation was impossible, and furthermore that the *Tseitin formulas* provided a separation between these systems [8]. For any graph G and any $\{0,1\}$-labelling ℓ of the vertices of G, the *Tseitin formula* of (G, ℓ) is the following system of $\mathbb{F}_2$-linear equations: for each edge e we introduce a variable x_e, and for each vertex v we have an equation

$$\bigoplus_{u : uv \in E} x_{uv} = \ell(v)$$

asserting that the sum of the edge variables incident with v must agree with its label $\ell(v)$ (note such a system is unsatisfiable as long as $\sum_v \ell(v)$ is odd). On the one hand, Beame et al. proved that there are *quasi-polynomial size* Stabbing Planes refutations of the Tseitin formulas [8]. On the other hand, Tseitin formulas had long been conjectured to be exponentially hard for Cutting Planes [23], as they form one of the canonical families of hard examples for algebraic and semi-algebraic proof systems, including Nullstellensatz [37], Polynomial Calculus [17], and Sum-of-Squares [38,59].

In a recent breakthrough, the long-standing conjecture that Tseitin was exponentially hard for Cutting Planes was *refuted* by Dadush and Tiwari [25], who gave *quasi-polynomial size* Cutting Planes refutations of Tseitin instances. Moreover, to prove their result, Dadush and Tiwari showed how to *translate* the quasipolynomial-size Stabbing Planes refutations of Tseitin into Cutting Planes refutations. This translation result is interesting for several reasons. First, it brings up the possibility that Cutting Planes *can actually* simulate Stabbing Planes. If possible, such a simulation would allow the significant analysis done on the Cutting Planes system to be lifted directly to branch-and-cut solvers. In particular, this would mean that the known exponential-size lower bounds for Cutting Planes refutations would immediately imply the first exponential lower bounds for these algorithms for arbitrary branching heuristics. Second, the translation converts *shallow* Stabbing Planes proofs into *very deep* Cutting Planes proofs: the Stabbing Planes refutation of Tseitin has depth $O(\log^2 n)$ and quasi-polynomial size, while the Cutting Planes refutation has quasipolynomial size *and* depth. This is quite unusual since simulations between proof systems typically preserve the structure of the proofs, and thus brings up the possibility that the Tseitin formulas yield a *supercritical* size/depth tradeoff – formulas with short proofs, requiring *superlinear* depth. For contrast: another simulation from the literature which emphatically does *not* preserve the structure of proofs is the simulation of *bounded-size* resolution by *bounded-width* resolution by Ben-Sasson and Wigderson [10]. In this setting, it is known that this simulation is tight [14], and even that there exist formulas refutable in resolution width w requiring maximal size $n^{\Omega(w)}$ [5]. Furthermore, under the additional assumption that the proofs are *tree-like*, Razborov [56] proved a supercritical trade-off between width and size.

1.1 Our Results

A New Characterization of Cutting Planes

Our first main result gives a *characterization* of Cutting Planes proofs as a natural subsystem of Stabbling Planes that we call *Facelike* Stabbing Planes. A Stabbing Planes query is *facelike* if one of the sets $P \cap \{ax \leq b-1\}$ or $P \cap \{ax \geq b\}$ is either empty or is a face of the polytope P, and a Stabbing Planes proof is said to be facelike if it only uses facelike queries. Our main result is the following theorem.

▶ **Theorem 1.** *The proof systems* CP *and Facelike* SP *are polynomially equivalent.*

The proof of this theorem is inspired by Dadush and Tiwari's upper bound for the Tseitin formulas. Indeed, the key tool underlying both their proof and ours is a lemma due to Schrijver [60] which allows us to simulate CP refutations of faces of a polytope, when beginning from P itself.

Using this equivalence we prove the following surprising simulation (see Figure 1), stating that Stabbing Planes proofs with relatively small coefficients (quasi-polynomially bounded in magnitude) can be quasi-polynomially simulated by Cutting Planes.

▶ **Theorem 2.** *Let F be any unsatisfiable CNF formula on n variables, and suppose that there is a* SP *refutation of F in size s and maximum coefficient size c. Then there is a* CP *refutation of F in size $s(cn)^{\log s}$.*

In fact, we prove a more general result (Theorem 16) which holds for arbitrary polytopes $P \in \mathbb{R}^n$, rather than only for CNF formulas, which degrades with the *diameter* of P. This should be contrasted with the work of Dadush and Tiwari [25], who show that any SP proof of size s of a polytope with diameter d can be assumed to have coefficients of size $(nd)^{O(n^2)}$.

As a second application of Theorem 1, we generalize Dadush and Tiwari upper bound for Tseitin to show that Cutting Planes can refute any unsatisfiable system of linear equations over a finite field. This follows by showing that, like Tseitin, we can refute such systems of linear equations in quasi-polynomial-size Facelike SP.

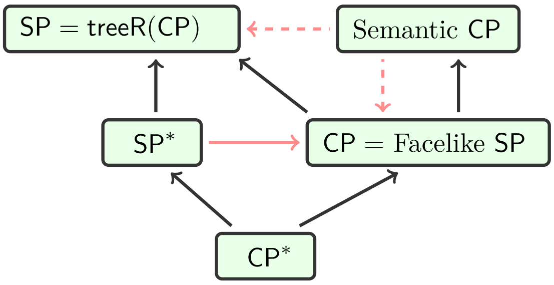


Figure 1 Known relationships between proof systems considered in this paper. A solid black (red) arrow from proof system P_1 to P_2 indicates that P_2 can polynomially (quasi-polynomially) simulate P_1. A dashed arrow indicates that this simulation cannot be done.

▶ **Theorem 3.** *Let F be the CNF encoding of an unsatisfiable system of m linear equations over a finite field. There is a* CP *refutation of F of size $|F|^{O(\log m)}$.*

This should be contrasted with the work of Filmus, Hrubeš, and Lauria [30], which gives several unsatisfiable systems of linear equations over $\mathbb{R}$ that require *exponential size* refutations in Cutting Planes (see Figure 1).

Lower Bounds

An important open problem is to prove superpolynomial size lower bounds for Stabbing Planes proofs. We make significant progress toward this goal by proving the first superpolynomial lower bounds on the size of low-weight Stabbing Planes proofs. Let SP* denote the family of Stabbing Planes proofs in which each coefficient has at most quasipolynomial ($n^{\log^{O(1)} n}$) magnitude.

▶ **Theorem 4.** *There exists a family of unsatisfiable CNF formulas $\{F_n\}$ such that any* SP* *refutation of F requires size at least $2^{n^{\varepsilon}}$ for constant $\varepsilon > 0$.*

Our proof follows in a straightforward manner from Theorem 2 together with known Cutting Planes lower bounds. We view this as a step toward proving SP lower bounds (with no restrictions on the weight). Indeed, lower bounds for CP* (low-weight Cutting Planes) [15] were first established, and led to (unrestricted) CP lower bounds [54].

Our second lower bound is a new linear depth lower bound for *semantic* Cutting Planes proofs. (In a semantic Cutting Planes proof the deduction rules for CP are replaced by a simple and much stronger *semantic deduction rule*).

▶ **Theorem 5.** *For all sufficiently large n there is a graph G on n vertices and a labelling ℓ such that the Tseitin formula for (G, ℓ) requires $\Omega(n)$ depth to refute in Semantic Cutting Planes.*

We note that depth lower bounds for Semantic Cutting Planes have already established via communication complexity arguments. However, since Tseitin formulas have short communication protocols, our depth bound for semantic Cutting Planes proofs of Tseitin is new.

Theorem 5 is established via a new technique for proving lower bounds on the depth of semantic Cutting Planes proofs. Our technique is inspired by the result of Buresh-Oppenheim et al. [16], who proved lower bounds on the depth of Cutting Planes refutations of Tseitin by studying the *Chátal rank* of the associated polytope P. Letting $P^{(d)}$ be the polytope composed of all inequalities which can be derived in depth d in Cutting Planes. The Chátal rank of P is the minimum d such that $P^{(d)} = \emptyset$. Thus, in order to establish a depth lower bound of depth d, one would like to show the existence of a point $p \in P^{(d)}$. To do so, they give a sufficient criterion for a point p to be in $P^{(i)}$ in terms of the points in $P^{(i-1)}$. This criterion relies on a careful analysis of the specific rules of Cutting Planes, and is no longer sufficient for semantic CP. Instead, we develop an analogous criterion for semantic CP by using novel *geometric* argument (Lemma 28) which we believe will be of independent interest.

Our main motivation behind this depth bound is as a step towards proving a *supercritical* tradeoff in CP for Tseitin formulas. A supercritical tradeoff for CP, roughly speaking, states that small size CP proofs must sometimes necessarily be very deep – that is, beyond the trivial depth upper bound of $O(n)$ [11,56]. (Observe that Dadush and Tiwari's quasipolynomial-size CP refutations of Tseitin are quasipolynomially deep; this is preserved by our simulation of Facelike Stabbing Planes by Cutting Planes in Theorem 1.) Establishing supercritical tradeoffs is a major challenge, both because hard examples witnessing such a tradeoff are rare, and because current methods seem to fail beyond the critical regime. In fact, to date the only supercritical tradeoffs between size and depth for known proof systems are due to Razborov, under the additional assumption that the proofs have *bounded width.* Namely, Razborov exhibited a supercritical size-depth tradeoff for bounded width tree-like resolution [56], and then extended this result to CP proofs in which each inequality has a bounded number of distinct variables [57].

How could one prove a supercritical depth lower bound for Cutting Planes? All prior depth lower bounds for Cutting Planes proceed by either reducing to communication complexity, or by using so-called *protection lemmas* (e.g. [16]). Since communication complexity is always at most n, it will be useless for proving supercritical lower bounds directly. It therefore stands to reason that we should focus on improving the known lower bounds using protection lemmas and, indeed, our proof of Theorem 5 is a novel geometric argument which generalizes the top-down "protection lemma" approach [16] for syntactic CP. At this point in time we are currently unable to use protection lemma techniques to prove size-depth tradeoffs, so, we leave this as an open problem.

▶ **Conjecture 6.** *There exists a family of unsatisfiable formulas $\{F_n\}$ such that F_n has quasipolynomial-size* CP *proofs, but any quasipolynomial-size proof requires superlinear depth.*

1.2 Related Work

Lower Bounds on SP and treeR(CP)

Several lower bounds on subsystems of SP and treeR(CP) have already been established. Krajíček [44] proved exponential lower bounds on the size of R(CP) proofs in which both the *width* of the clauses and the magnitude of the coefficients of each line in the proof are bounded. Concretely, let these bounds be w and c respectively. The lower bound that he obtains is $2^{n^{\Omega(1)}}/c^{w \log^2 n}$. Kojevnikov [43] removed the dependence on the coefficient size for treeR(CP) proofs, obtaining a bound of $\exp(\Omega(\sqrt{n/w} \log n))$. Beame et al. [8] provide a size-preserving simulation of Stabbing Planes by treeR(CP) which translates a depth d Stabbing Planes proof into a width d treeR(CP) proof, and therefore this implies lower bounds on the size of SP proofs of depth $o(n/\log n)$. Beame et al. [8] exhibit a function for which there are no SP refutations of depth $o(n/\log^2 n)$ via a reduction to the communication complexity of the CNF search problem.

Supercritical Tradeoffs

Besides the work of Razborov [56], a number of supercritical tradeoffs have been observed in proof complexity. Perhaps most relevant for our work, Razborov [57] proved a supercritical tradeoff for Cutting Planes proofs under the assumption that each inequality has a bounded number of distinct variables (mimicking the bound on the width of each clause in the supercritical tradeoff of [56]).

A number of supercritical tradeoffs are also known between proof width and proof *space*. Beame et al. [7] and Beck et al. [9] exhibited formulas which admit polynomial size refutations in Resolution and the Polynomial Calculus respectively, and such that any refutation of sub-linear space necessitates a superpolynomial blow-up in size. Recently, Berkholz and Nordström [11] gave a supercritical trade-off between width and space for Resolution.

Depth in Cutting Planes and Stabbing Planes

It is widely known (and easy to prove) that any unsatisfiable family of CNF formulas can be refuted by exponential size and *linear* depth Cutting Planes. It is also known that neither Cutting Planes nor Stabbing Planes can be *balanced*, in the sense that a depth-d proof can always be transformed into a size $2^{O(d)}$ proof [8, 16]. This differentiates both of these proof systems from more powerful proof systems like Frege, for which it is well-known how to balance arbitrary proofs [22]. Furthermore, even though both the Tseitin principles and systems of linear equations in finite fields can be proved in both quasipolynomial-size *and* $O(\log^2 n)$ depth in Facelike SP, the simulation of Facelike SP by CP *cannot* preserve both size and depth, as the Tseitin principles are known to require depth $\Theta(n)$ to refute in CP [16].

We first recall the known depth lower bound techniques for Cutting Planes, semantic Cutting Planes, and Stabbing Planes proofs. In all of these proof systems, arguably the primary method for proving depth lower bounds is by reducing to *real communication complexity* [8, 40]; however, communication complexity is always trivially upper bounded by n, and it is far from clear how to use the assumption on the size of the proof to boost this to superlinear.

A second class of methods have been developed for *syntactic* Cutting Planes, which lower bound *rank measures* of a polytope, such as the Chvátal rank. In this setting, lower bounds are typically proven using so-called *protection lemmas* [16], which seems much more amenable to applying a small-size assumption on the proof. We also remark that for many formulas (such as the Tseitin formulas!) it is known how to achieve $\Omega(n)$-depth lower bounds in Cutting Planes via protection lemmas, while proving even $\omega(\log n)$ lower bounds via communication complexity is impossible, due to a known folklore upper bound.

The first lower bound on the Chvátal rank was established by Chvátal et al. [20], who proved a linear bound for a number of polytopes in $[0, 1]^n$. Much later, Pokutta and Schulz [53] characterized the polytopes $P \subseteq [0, 1]$ with $P \cap \mathbb{Z}^n = \emptyset$ which have Chvátal rank exactly n. However, unlike most other cutting planes procedures, the Chvátal rank is not of polytopes $P \cap [0, 1]^n$ with $P \cap \mathbb{Z}^n = \emptyset$ is not upper bounded by n. Eisenbrand and Schulz [29] showed that the Chvátal rank of any polytope $P \subseteq [0, 1]^n$ is at most $O(n^2 \log n)$ and gave examples where it is $\Omega(n)$; a nearly-matching quadratic lower bound was later established by Rothvoß and Sanita [58]. For CNF formulas, the Chvátal rank is (trivially) at most n. Buresh-Oppenheim et al. [16] gave the first lower bounds on the Chvátal rank a number of CNF formulas, including an $\Omega(n)$ lower bound for the Tseitin formulas.

The rank of a number of generalizations of Cutting Planes has been studied as well. However, none of these appear to capture the strength of semantic Cutting Planes. Indeed, semantic Cutting Planes is able to refute Knapsack in a single cut, and therefore is known not to be polynomially verifiable unless $\mathsf{P} = \mathsf{NP}$ [30]. Lower bounds on the rank when using split cuts and mixed integer cuts, instead of CG cuts, was established in [24]. Pokutta and Schulz [52] obtained $\Omega(n/\log n)$ rank lower bounds on the complete tautology (which includes every clause of width n) for the broad class of *admissible cutting planes*, which includes syntactic Cutting Planes, split cuts, and many of the lift-and-project operators. Bodur et al. [13] studied the relationship between rank and integrality gaps for another broad generalization of Cutting Planes known as *aggregate cuts*.

2 Preliminaries

We first recall the definitions of some key proof systems.

Resolution

Fix an unsatisfiable CNF formula F over variables $x_1, \ldots, x_n$. A *Resolution refutation* P of F is a sequence of clauses $\{C_i\}_{i \in [s]}$ ending in the empty clause $C_s = \emptyset$ such that each C_i is in F or is derived from earlier clauses C_j, C_k with $j, k < i$ using one of the following rules:

- *Resolution.* $C_i = (C_j \setminus \{\ell_k\}) \cup (C_k \setminus \{\overline{\ell}_k\})$ where $\ell_k \in C_j$, $\overline{\ell}_k \in C_k$ is a literal.
- *Weakening.* $C_i \supseteq C_j$.

The *size* of the resolution proof is s, the number of clauses. It is useful to visualize the refutation P as a directed acyclic graph; with this in mind the *depth* of the proof (denoted $\mathsf{depth}_{\mathsf{Res}}(P)$) is the length of the longest path in the proof DAG. The *resolution depth* $\mathsf{depth}_{\mathsf{Res}}(F)$ of F is the minimal depth of any resolution refutation of F.

Cutting Planes and Semantic Cutting Planes

A *Cutting Planes* (CP) *proof* of an inequality $cx \geq d$ from a system of linear inequalities P is given by a sequence of inequalities

$$a_1 x \geq b_1, a_2 x \geq b_2, \ldots, a_s x \geq b_s$$

such that $a_s = c$, $b_s = d$, and each inequality $a_i x \geq b_i$ is either in P or is deduced from earlier inequalities in the sequence by applying one of the two rules *Linear Combination* or *Division Rule* described at the beginning of Section 1. We will usually be interested in the case that the list of inequalities P defines a polytope.

An alternative characterization of Cutting Planes uses *Chvátal-Gomory cuts* (or just *CG cuts*) [18, 23]. Let P be a polytope. A hyperplane $ax = b$ is *supporting* for P if $b = \max\{ax : x \in P\}$, and if $ax = b$ is a supporting hyperplane then the set $P \cap \{x \in \mathbb{R}^n : ax = b\}$ is called a *face* of P. An inequality $ax \leq b$ is *valid* for P if every point of P satisfies the inequality and $ax = b$ is a supporting hyperplane of P.

▶ **Definition 7.** *Let $P \subseteq \mathbb{R}^n$ be a polytope, and let $ax \geq b$ be any valid inequality for P such that all coefficients of a are relatively prime integers. The halfspace $\{x \in \mathbb{R}^n : ax \geq \lceil b \rceil\}$ is called a* CG cut *for P. (We will sometimes abuse notation and refer to the inequality $ax \geq \lceil b \rceil$ also as a CG cut.)*

If $ax \geq \lceil b \rceil$ is a CG cut for the polytope P, then we can derive $ax \geq \lceil b \rceil$ from P in $O(n)$ steps of Cutting Planes by Farkas Lemma (note that the inequality $ax \geq b$ is valid for P by definition, so we can deduce $ax \geq b$ as a linear combination of the inequalities of P and then apply the division rule). If P is a polytope and H is a CG cut, then we will write $P \vdash P \cap H$, and say that $P \cap H$ is *derived* from P.

Given a CNF formula F, we can translate F into a system of linear inequalities in the following natural way. First, for each variable x_i in F add the inequality $0 \leq x_i \leq 1$. If $C = \bigvee_{i \in P} x_i \vee \bigvee_{i \in N} \neg x_i$ is a clause in F, then we add the inequality

$$\sum_{i \in P} x_i + \sum_{i \in N} (1 - x_i) \geq 1.$$

It is straightforward to see that the resulting system of inequalities will have no integral solutions if and only if the original formula F is unsatisfiable. With this translation we consider Cutting Planes refutations (defined in the introduction) of F to be refutations of the translation of F to linear inequalities.

The *semantic Cutting Planes* proof system (denoted sCP or Semantic CP) is a strengthening of Cutting Planes proofs to allow *any deduction* that is sound over Boolean points [15]. Like Cutting Planes, an sCP proof is given by a sequence of halfspaces $\{a_i x \geq c_i\}_{i \in [s]}$, but now we can use the following very powerful *semantic deduction rule*:

- *Semantic Deduction.* From $a_j x \geq c_j$ and $a_k x \geq c_k$ deduce $a_i x \geq c_i$ if every $\{0,1\}$ assignment satisfying both $a_j x \geq c_j$ and $a_k x \geq c_k$ also satisfies $a_i x \geq c_i$.

Filmus et al. [30] showed that sCP is extremely strong: there are instances for which any refutation in CP requires exponential size, and yet these instances admit polynomial-size refutations in semantic sCP.

The size of a Cutting Planes proof is the number of lines (it is known that for unsatisfiable CNF formulas that this measure is polynomially related to the length of the bit-encoding of the proof [23]). As with Resolution, it is natural to arrange Cutting Planes proofs into a proof DAG. With this in mind we analogously define $\mathsf{depth}_{\mathsf{CP}}(F)$ and $\mathsf{depth}_{\mathsf{sCP}}(F)$ to be the smallest depth of any (semantic) Cutting Planes proof of F.

It is known that *any* system of linear inequalities in the unit cube has CP depth at most $O(n^2 \log n)$, and moreover there are examples requiring CP-depth more than n [29]. However for unsatisfiable CNF formulas, the CP-depth is at most n [12].

Stabbing Planes

Let F be an unsatisfiable system of linear inequalities. A *Stabbing Planes (*SP*) refutation* of F is a directed binary tree, T, where each edge is labelled with a linear integral inequality satisfying the following *consistency conditions*:

- *Internal Nodes.* For any internal node u of T, if the right outgoing edge of u is labelled with $ax \geq b$, then the left outgoing edge is labelled with its *integer negation* $ax \leq b - 1$.
- *Leaves.* Each leaf node v of T is labelled with a non-negative linear combination of inequalities in F with inequalities along the path leading to v that yields $0 \geq 1$.

For an internal node u of T, the pair of inequalities $(ax \leq b-1, ax \geq b)$ is called the *query* corresponding to the node. Every node of T has a polytope P associated with it, where P is the polytope defined by the intersection of the inequalities in F together with the inequalities labelling the path from the root to this node. We will say that the polytope P *corresponds* to this node. The *slab* corresponding to the query is $\{x^* \in \mathbb{R}^n \mid b - 1 < ax^* < b\}$, which is the set of points ruled out by this query. The *width* of the slab is the minimum distance between $ax \leq b - 1$ and $ax \geq b$, which is $1/\|a\|_2$. The *size* of a refutation is the bit-length needed to encode a description of the entire proof tree, which, for CNF formulas as well as sufficiently bounded systems of inequalities, is polynomially equivalent to the number of queries in the refutation [25]. As well, the *depth* of the refutation is the depth of the binary tree. The proof system SP* is the subsystem of Stabbing Planes obtained by restricting all coefficients of the proofs to have magnitude at most quasipolynomial ($n^{\log^{O(1)} n}$) in the number of input variables.

The Stabbing Planes proof system was introduced by Beame et al. [8] as a generalization of Cutting Planes that more closely modelled query algorithms and branch-and-bound solvers. Beame et al. proved that SP is equivalent to the proof system TreeR(CP) introduced by Krajíček [44] which can be thought of as a generalization of Resolution where the literals are replaced with integer-linear inequalities.

3 Translating Stabbing Planes into Cutting Planes

3.1 Equivalence of CP with Subsystems of SP

In this section we prove Theorem 1, restated below, which characterizes Cutting Planes as a non-trivial subsystem of Stabbing Planes.

▶ **Theorem 8** (Theorem 1). *The proof systems* CP *and Facelike* SP *are polynomially equivalent.*

We begin by formally defining Facelike SP.

▶ **Definition 9.** *A Stabbing Planes query $(ax \leq b-1, ax \geq b)$ at a node P is* facelike *if one of the sets $P \cap \{x \in \mathbb{R}^n : ax \leq b-1\}$, $P \cap \{x \in \mathbb{R}^n : ax \geq b\}$ is empty or a face of P (see Figure 2b). An* SP *refutation is facelike if every query in the refutation is facelike.*

Enroute to proving Theorem 1, it will be convenient to introduce the following further restriction of Facelike Stabbing Planes.

▶ **Definition 10.** *A Stabbing Planes query $(ax \leq b-1, ax \geq b)$ at a node corresponding to a polytope P is* pathlike *if at least one of $P \cap \{x \in \mathbb{R}^n : ax \leq b-1\}$ and $P \cap \{x \in \mathbb{R}^n : ax \geq b\}$ is empty (see Figure 2a). A Pathlike* SP *refutation is one in which every query is pathlike.*

The name "pathlike" stems from the fact that the underlying graph of a pathlike Stabbing Planes proof is a path, since at most one child of every node has any children (see Figure 2). In fact, we have already seen (nontrivial) pathlike SP queries under another name: Chvátal-Gomory cuts.

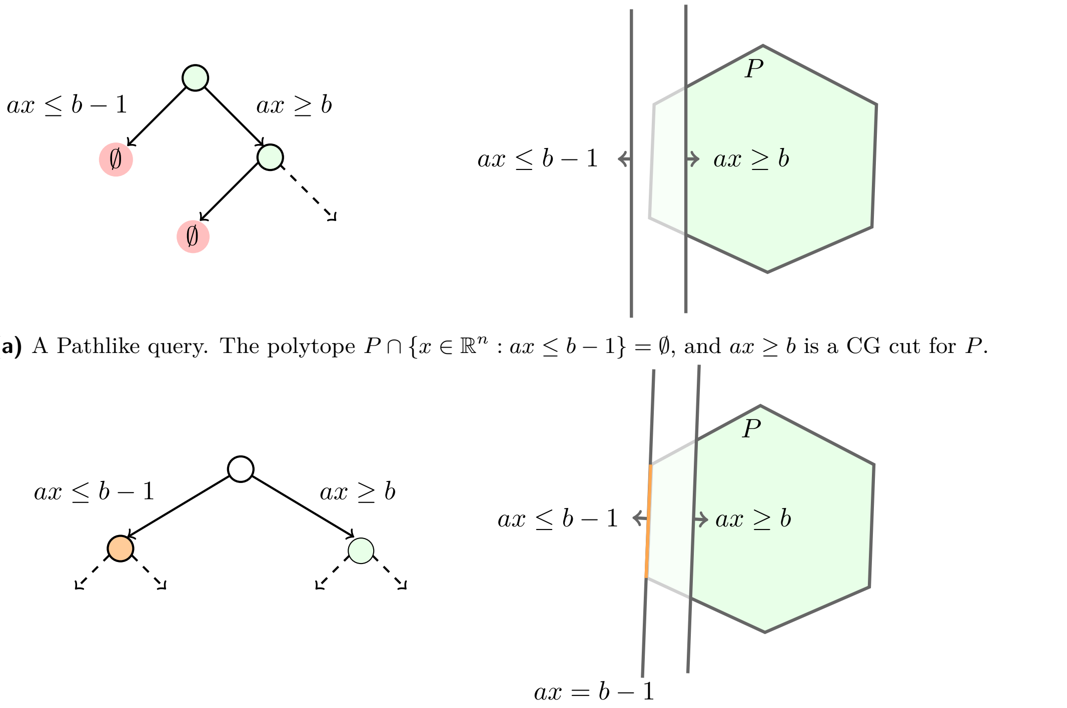
▶ **Lemma 11.** *Let P be a polytope and let $(ax \leq b-1, ax \geq b)$ be a pathlike Stabbing Planes query for P. Assume w.l.o.g. that $P \cap \{x \in \mathbb{R}^n : ax \leq b-1\} = \emptyset$ and that $P \cap \{x \in \mathbb{R}^n : ax \geq b\} \subsetneq P$. Then $ax \geq b$ is a CG cut for P.*

Proof. Since $ax \geq b$ is falsified by some point in P, it follows that there exists some $0 < \varepsilon < 1$ such that $ax \geq b - \varepsilon$ is valid for P – note that $\varepsilon < 1$ since otherwise $ax \leq b-1$ would not have empty intersection with P. This immediately implies that $ax \geq b$ is a CG cut for P. ◀

With this observation we can easily prove that Pathlike SP is equivalent to CP. Throughout the remainder of the section, for readability, we will use the abbreviation $P \cap \{ax \geq b\}$ for $P \cap \{x \in \mathbb{R}^n : ax \geq b\}$, for any polytope P and linear inequality $ax \geq b$.

▶ **Lemma 12.** *Pathlike* SP *is polynomially equivalent to* CP*.*

Proof. First, let $a_1x \geq b_1, a_2x \geq b_2, \ldots, a_sx \geq b_s$ be a CP refutation of an unsatisfiable system of linear inequalities $Ax \geq b$. Consider the sequence of polytopes $P_0 = \{Ax \geq b\}$ and $P_i = P_{i-1} \cap \{a_ix \geq b_i\}$. By inspecting the rules of CP, it can observed that $P_i \cap \{a_ix \leq b_i - 1\} = \emptyset$ and thus P_{i+1} can be deduced using one pathlike SP query from P_i for all $0 \leq i \leq s$.



(a) A Pathlike query. The polytope $P \cap \{x \in \mathbb{R}^n : ax \leq b-1\} = \emptyset$, and $ax \geq b$ is a CG cut for P.

(b) A Facelike query. The polytope $P \cap \{x \in \mathbb{R}^n : ax \leq b-1\} = P \cap \{x \in \mathbb{R}^n : ax = b-1\}$ is a face of P.

Figure 2 Pathlike and Facelike SP queries on a polytope P. On the left are the proofs and on the right are the corresponding effects on the polytope.

Conversely, let P be any polytope and let $(ax \leq b-1, ax \geq b)$ be any pathlike SP query to P (so, suppose w.l.o.g. that the halfspace defined by $ax \leq b-1$ has empty intersection with P). By Lemma 11, $ax \geq b$ is a CG cut for P, and so can be deduced in Cutting Planes from the inequalities defining P in length $O(n)$ (cf. Section 2). Applying this to each query in the Pathlike SP proof yields the theorem. ◀

Next, we show how to simulate Facelike SP proofs by Pathlike SP proofs of comparable size. The proof of Lemma 14 is inspired by Dadush and Tiwari [25], and will use the following lemma due to Schrijver [60] (although, we use the form appearing in [23]). Recall that we write $P \vdash P'$ for polytopes P, P' to mean that P' can be obtained from P by adding a single CG cut to P.

▶ **Lemma 13** (Lemma 2 in [23]). *Let P be a polytope defined by a system of integer linear inequalities and let F be a face of P. If $F \vdash F'$ then there is a polytope P' such that $P \vdash P'$ and $P' \cap F \subseteq F'$.*

▶ **Lemma 14.** *Facelike* SP *is polynomially equivalent to Pathlike* SP.

Proof. That Facelike SP simulates Pathlike SP follows by the fact that any Pathlike SP query is a valid query in Facelike SP. For the other direction, consider an SP refutation π of size t. We describe a recursive algorithm for generating a Pathlike SP proof from π. The next claim will enable our recursive case.

Claim. Let P be a polytope and suppose $ax \geq b$ is valid for P. Assume that $P \cap \{ax = b\}$ has a Pathlike SP refutation using s queries. Then $P \cap \{ax \geq b+1\}$ can be derived from P in Pathlike SP using $s+1$ queries.

Proof of Claim. Since $ax \geq b$ is valid for P it follows that $F = P \cap \{ax = b\}$ is a face of P by definition. Consider the Pathlike SP refutation $F_0, F_1, \dots F_s = \emptyset$, where the ith polytope F_i for $i < s$ is obtained from F_{i-1} by applying a pathlike SP query and proceeding to the non-empty child. Without loss of generality we may assume that $F_i \subsetneq F_{i-1}$ for all i, and so applying Lemma 11 we have that $F_{i-1} \vdash F_i$ for all i. Thus, by applying Lemma 13 repeatedly, we get a sequence of polytopes $P = P_0 \vdash P_1 \vdash \cdots \vdash P_s$ such that $P_i \cap F = P_i \cap \{ax = b\} \subseteq F_i$. This means that $P_s \cap \{ax = b\} \subseteq F_s = \emptyset$, and so $(ax \leq b, ax \geq b+1)$ is Pathlike SP query for P_s. This means that $P_s \vdash P_s \cap \{ax \geq b+1\} \subseteq P \cap \{ax \geq b+1\}$. Since any CG cut can be implemented as a Pathlike SP query the claim follows by applying the s CG cuts as pathlike queries, followed by the query $(ax \leq b, ax \geq b+1)$. ◀

We generate a Pathlike SP refutation by the following recursive algorithm, which performs an *in-order* traversal of π. At each step of the recursion (corresponding to a node in π) we maintain the current polytope P we are visiting and a Pathlike SP proof Π – initially, P is the initial polytope and $\Pi = \emptyset$. We maintain the invariant that when we finish the recursive step at node P, the Pathlike SP refutation Π is a refutation of P. The algorithm is described next:

1. Let $(ax \leq b-1,\ ax \geq b)$ be the current query and suppose that $ax \geq b-1$ is valid for P.
2. Recursively refute $P \cap \{ax \leq b-1\} = P \cap \{ax = b-1\}$, obtaining a Pathlike SP refutation Π with t queries.
3. Apply the above Claim to deduce $P \cap \{ax \geq b\}$ from P in $t+1$ queries.
4. Refute $P \cap \{ax \geq b\}$ by using the SP refutation for the right child.

Correctness follows immediately from the Claim, and also since the size of the resulting proof is the same as the size of the SP refutation. ◀

Theorem 1 then follows by combining Lemma 12 with Lemma 14.

3.2 Simulating SP* by CP

In this section we prove Theorem 2, restated below for convenience.

▶ **Theorem 15** (Theorem 2). *Let F be any unsatisfiable CNF formula on n variables, and suppose that there is a* SP *refutation of F in size s and maximum coefficient size c. Then there is a* CP *refutation of F in size $s(cn)^{\log s}$.*

To prove this theorem, we will show that *any* low coefficient SP proof can be converted into a Facelike SP proof with only a quasi-polynomial loss. If P is a polytope let $d(P)$ denote the *diameter* of P, which is the maximum Euclidean distance between any two points in P. Theorem 2 follows immediately from the following theorem.

▶ **Theorem 16.** *Let P be a polytope and suppose there is an* SP *refutation of P with size s and maximum coefficient size c. Then there is a Facelike* SP *refutation of P in size*

$$s(c \cdot d(P)\sqrt{n})^{\log s}.$$

Proof. The theorem is by induction on s. Clearly, if $s = 1$ then the tree is a single leaf and the theorem is vacuously true.

We proceed to the induction step. Let P be the initial polytope and π be the SP proof. Consider the first query $(ax \leq b,\ ax \geq b+1)$ made by the proof, and let π_L be the SP proof rooted at the left child (corresponding to $ax \leq b$) and let π_R be the SP proof rooted at the right child. Let P_L denote the polytope at the left child and P_R denote the polytope at the right child. By induction, let π'_L and π'_R be the Facelike SP refutations for P_L and P_R guaranteed by the statement of the theorem.

Suppose w.l.o.g. that $|\pi_L| \leq |\pi|/2$. Let b_0 be the largest integer such that $ax \geq b_0$ is satisfied for any point in P. The plan is to replace the first query $(ax \leq b,\ ax \geq b+1)$ with a sequence of queries $q_0, q_1, \ldots, q_{t-1}$ such that

- For each i, $q_i = (ax \leq b_0 + i,\ ax \geq b_0 + i + 1)$.
- The query q_0 is the root of the tree and q_i is attached to the right child of q_{i-1} for $i \geq 1$.
- $q_{t-1} = (ax \leq b,\ ax \geq b+1)$.

After doing this replacement, instead of having two child polytopes P_L, P_R below the top query, we have $t+1$ polytopes $P_0, P_1, \ldots, P_{t+1}$ where $P_i = P \cap \{ax = b_0 + i\}$ and $P_{t+1} = P_R$. To finish the construction, for each $i \leq t$ use the proof π'_L to refute P_i and the proof π'_R to refute P_{t+1}.

We need to prove three statements: this new proof is a valid refutation of P, the new proof is facelike, and that the size bound is satisfied.

First, it is easy to see that this is a valid proof, since for each $i \leq t$ the polytope $P_i \subseteq P_L$ and $P_{t+1} \subseteq P_R$ – thus, the refutations π'_L and π'_R can be used to refute the respective polytopes.

Second, to see that the proof is facelike, first observe that all the queries in the subtrees π'_L, π'_R are facelike queries by the inductive hypothesis. So, we only need to verify that the new queries at the top of the proof are facelike queries, which can easily be shown by a quick induction. First, observe that the query q_0 is a facelike query, since b_0 was chosen so that $ax \geq b_0$ is valid for the polytope P. By induction, the query $q_i = (ax \leq b_0+i,\ ax \geq b_0+i+1)$ is a facelike query since the polytope P_i associated with that query is $P \cap \{ax \geq b_0 + i\}$ by definition. Thus $ax \geq b_0 + i$ is valid for the polytope at the query.

Finally, we need to prove the size upper bound. Let s be the size of the original proof, s_L be the size of π_L and s_R be the size of π_R. Observe that the size of the new proof is given by the recurrence relation

$$f(s) = t \cdot f(s_L) + f(s_R).$$

where $f(1) = 1$. Since the queries $q_0, q_1, \ldots, q_{t-1}$ cover the polytope P_L with slabs of width $1/\|a\|_2$, it follows that

$$t \leq d(P_L)\|a\|_2 \leq d(P)\sqrt{n}\|a\|_\infty = d(P)c\sqrt{n}$$

where we have used that the maximum coefficient size in the proof is c. Thus, by induction, the previous inequality, and the assumption that $s_L \leq s/2$, we can conclude that the size of the proof is

$$\begin{aligned} f(s) &\leq s_L(c \cdot d(P)\sqrt{n})(c \cdot d(P_L)\sqrt{n})^{\log s_L} + s_R(c \cdot d(P_R)\sqrt{n}+)^{\log s_R} \\ &\leq s_L(c \cdot d(P)\sqrt{n})(c \cdot d(P)\sqrt{n})^{\log(s/2)} + s_R(c \cdot d(P)\sqrt{n})^{\log s} \\ &\leq s_L(c \cdot d(P)\sqrt{n})^{\log s} + s_R(c \cdot d(P)\sqrt{n})^{\log s} \\ &= s(c \cdot d(P)\sqrt{n})^{\log s}. \end{aligned}$$ ◀

Theorem 2 follows immediately, since for any CNF formula F the encoding of F as a system of linear inequalities is contained in the n-dimensional cube $[0,1]^n$, which has diameter $\sqrt{n}$. We may also immediately conclude Theorem 4 by applying the known lower bounds on the size of Cutting Planes proofs [32,33,39,54].

As a consequence of Theorem 2 and the non-automatability of Cutting Planes [36], we can conclude that SP^* proofs cannot be found efficiently assuming $\mathsf{P} \neq \mathsf{NP}$.

▶ **Corollary 17.** SP^* *is not automatable unless* $\mathsf{P} \neq \mathsf{NP}$.

This follows by observing that the argument in [36] does not require large coefficients.

4 Refutations of Linear Equations over a Finite Field

In this section we prove Theorem 3. To do so, we will extend the approach used by Beame et al. [8] to prove quasi-polynomial upper bounds on the Tseitin formulas to work on any unsatisfiable set of linear equations over any finite field.

If $ax = b$ is a linear equation we say the *width* of the equation is the number of non-zero variables occurring in it. Any width-d linear equation over a finite field of size q, denoted $\mathbb{F}_q$, can be represented by a CNF formula with q^{d-1} width-d clauses – one ruling out each falsifying assignment. For a width-d system of m linear equations F over $\mathbb{F}_q$, we will denote by $|F| := mq^{d-1}$ the size of the CNF formula encoding F.

▶ **Theorem 18.** *Let* $F = \{f_1 = b_1, \dots, f_m = b_m\}$ *be a width-d, unsatisfiable set of linear equations over* $\mathbb{F}_q$. *There is an* SP *refutation of (the CNF encoding of)* F *in size* $(mqd)^{O(\log m)}q^d = |F|^{O(\log m)}$.

First we sketch the idea for $\mathbb{F}_2$, i.e., a system of XOR equations. In this case the SP proof corresponds to a branch decomposition procedure which is commonly used to solve SAT (see e.g. [3,26,28,46]). View the system F as a hypergraph over n vertices (corresponding to the variables) and with a d-edge for each equation. Partition the set of hyperedges into two sets $E = E_1 \cup E_2$ of roughly the same size, and consider the *cut* of vertices that belong to both an edge in E_1 and in E_2. Using the SP rule we branch on all possible values of the sum of the cut variables in order to isolate E_1 and E_2. Once we know this sum, we are guaranteed that either E_1 is unsatisfiable or E_2 is unsatisfiable depending on the parity of the of the sum of the cut variables. This allows us to recursively continue on the side of the cut (E_1 or E_2) that is unsatisfiable. Since there are n Boolean variables, each cut corresponds to at most $n+1$ possibilities for the sum, and if we maintain that the partition of the hyper edges defining the cut is balanced, then we will recurse at most $O(\log m)$ times. This gives rise to a tree decomposition of fanout $O(n)$ and height $O(\log n)$.

Over a finite field of size q the proof will proceed in much the same way. Instead of a subgraph, at each step we will maintain a subset of the equations $I \subseteq [m]$ such that $\{f_i = b_i\}_{i \in I}$ must contain a constraint that is violated by the SP queries made so far. We partition I into two sets I_1 and I_2 of roughly equal size and query the values a and b of $\sum_{i \in I_1} f_i$ and $\sum_{i \in I_2} f_i$. Because F is unsatisfiable, at least one of $a - \sum_{i \in I_1} b_i \not\equiv 0$ or $b - \sum_{i \in I_2} b_i \not\equiv 0$, meaning that that it is unsatisfiable, and we recurse on it.

In the following, we will let z stand for a vector of $\mathbb{F}_q$-valued variables z_i. When we discuss any form $f := az$ where $a \in \mathbb{F}_q^n$ and z is a vector of n variables z_i, we will implicitly associate it with the linear form $\sum_{i \in [n]} a_i (\sum_{j \in [\log q]} x_{i,j})$ where $x_{i,j}$ are the $\log q$ many Boolean variables encoding z_i in the CNF encoding of F.

Proof of Theorem 18. Let $F = \{f_1 = b_1, \dots, f_m = b_m\}$ be a system of unsatisfiable linear equations over $\mathbb{F}_q$, where each $f_i = a_i z$ for $a_i \in \mathbb{F}_q^n$, and $b_i \in \mathbb{F}_q$. Because F is unsatisfiable, there exists a $\mathbb{F}_q$ linear combination of the equations in F witnessing this; formally, there exists $\alpha \in \mathbb{F}_q^n$ such that $\sum_{i\in[m]} \alpha_i f_i \equiv 0 \mod q$, but $\sum_{i\in[m]} \alpha_i b_i \not\equiv 0 \mod q$.

Stabbing Planes will implement the following binary search procedure for a violated equation; we describe the procedure first, and then describe how to implement it in Stabbing Planes. In each round we maintain a subset $I \subseteq [m]$ and an integer k_I representing the value of $\sum_{i\in I} \alpha_i f_i$. Over the algorithm, we maintain the invariant that $k_I - \sum_{i\in I} b_i \not\equiv 0 \bmod q$, which implies that there must be a contradiction to F inside of the constraints $\{f_i = b_i\}_{i\in I}$.

Initially, $I = [m]$ and we obtain k_I by querying the value of the sum $\sum_{i\in[m]} \alpha_i f_i$. If $k_I \not\equiv 0 \mod q$ then this contradicts the fact that $\sum_{i\in I} \alpha_i f_i \equiv 0 \mod q$; thus, the invariant holds. Next, perform the following algorithm.

1. Choose a balanced partition $I = I_1 \cup I_2$ (so that $||I_1| - |I_2|| \leq 1$).
2. Query the value of $\sum_{i\in I_1} \alpha_i f_i$ and $\sum_{i\in I_2} \alpha_i f_i$; denote these values by a and b respectively.
3. If $a - \sum_{i\in I_1} \alpha_i b_i \not\equiv 0 \bmod q$ then recurse on I_1 with $k_{I_1} := a$. Otherwise, if $b - \sum_{i\in I_2} \alpha_i b_i \not\equiv 0 \bmod q$ then recurse on I_2 with $k_{I_2} := b$.
4. Otherwise (if $a - \sum_{i\in I_1} \alpha_i b_i \equiv b - \sum_{i\in I_2} \alpha_i b_i \equiv 0 \bmod q$), then this contradicts the invariant:

$$\begin{aligned} 0 \not\equiv k_I - \sum_{i\in I} \alpha b_i &= \sum_{i\in I} \alpha_i (f_i - b_i) \\ &= \sum_{i\in I_1} \alpha_i (f_i - b_i) + \sum_{i\in I_2} \alpha_i (f_i - b_i) \\ &= (a - \sum_{i\in I_1} \alpha_i b_i) + (b - \sum_{i\in I_1} \alpha_i b_i) \equiv 0 \mod q. \end{aligned}$$

This recursion stops when $|I| = 1$, at which point we have an immediate contradiction between k_I and the single equation indexed by I.

It remains to implement this algorithm in SP. First, we need to show how to perform the queries in step 2. Querying the value of any sum $\sum_{i\in I} \alpha_i f_i$ can be done in a binary tree with at most q^2md leaves, one corresponding to every possible query outcome. Internally, this tree queries all possible integer values for this sum (e.g. $(\sum_{i\in I} \alpha_i f_i \leq 0, \sum_{i\in I} \alpha_i f_i \geq 1), (\sum_{i\in I} \alpha_i f_i \leq 1, \sum_{i\in I} \alpha_i f_i \geq 2), \dots$). For the leaf where we have deduced $\sum_{i\in[m]} \alpha_i f_i \leq 0$ we use the fact that each variable is non-negative to deduce that $\sum_{i\in[m]} \alpha_i f_i \geq 0$ as well. Note that q^2md is an upper bound on this sum because there are m equations, each containing at most d variables, each taking value at most $(q-1)$ [2]. Thus, step 2 can be completed in $(q^2md)^2$ queries.

Finally, we show how to derive refutations in the following cases: (i) when we deduced that $\sum_{i\in[m]} \alpha_i f_i \not\equiv 0 \mod q$ at the beginning, (ii) in step 4, (iii) when $|I| = 1$.

(i) Suppose that we received the value $a \not\equiv 0 \mod q$ from querying $\sum_{i\in[m]} \alpha_i f_i$. Note that every variable in $\sum_{i\in[m]} \alpha_i f_i$ is a multiple of q. Query

$$\Big(\sum_{i\in[m]} \alpha_i f_i / q \leq \lceil a/q \rceil - 1, \sum_{i\in[m]} \alpha_i f_i / q \geq \lceil a/q \rceil \Big).$$

At the leaf that deduces $\sum_{i\in[m]} \alpha_i f_i / q \leq \lceil a/q \rceil - 1$, we can derive $0 \geq 1$ as a non-negative linear combination of this inequality together with $\sum_{i\in[m]} \alpha_i f_i \geq a$. Similarly, at the other leaf $\sum_{i\in[m]} \alpha_i f_i / q \geq \lceil a/q \rceil$ can be combined with $\sum_{i\in[m]} \alpha_i f_i \leq a$ to derive $0 \geq 1$.

[2] Note that instead of querying the value of $\sum_{i\in I} \alpha_i f_i$ we could have queried $\sum_{i\in I} \alpha_i f_i$ (mod$q$) to decrease the number of leaves to qmd.

(ii) Suppose that $a - \sum_{i \in I_1} \alpha_i b_i \equiv b - \sum_{i \in I_2} \alpha_i b_i \equiv 0 \bmod q$. Then $0 \geq 1$ is derived by summing $\sum_{i \in I_1} \alpha_i f_i \geq a$, $\sum_{i \in I_2} \alpha_i f_i \geq b$ and $\sum_{i \in I} \alpha_i f_i \leq k_I$, all of which have already been deduced.

(iii) When $|I| = 1$ then we deduced that $a_I z = k_I$ for $k_I \not\equiv b_I \mod q$ and we would like to derive a contradiction using the axioms encoding $a_I z \equiv b_I$. These axioms are presented to SP as the linear-inequality encoding of a CNF formula, and while there are no integer solutions satisfying both these axioms and $a_I z = k_I$, there could in fact be *rational* solutions. To handle this, we simply force that each of the at most d variables in $a_I z$ takes an integer value by querying the value of each variable one by one. As there are at most d variables, each taking an integer value between 0 and $q - 1$, this can be done in a tree with at most q^d many leaves. At each leaf of this tree we deduce $0 \geq 1$ by a non-negative linear combination with the axioms, the integer-valued variables, and $a_I z \equiv b_I$.

The recursion terminates in at most $O(\log m)$ many rounds because the number of equations under consideration halves every time. Therefore, the size of this refutation is $(qmd)^{O(\log m)} q^d$. Note that by making each query in a balanced tree, this refutation can be carried out in depth $O(\log^2(mqd))$. ◀

Finally, we conclude Theorem 3.

Proof of Theorem 3. Observe that the SP refutation from Theorem 18 is facelike. Indeed, to perform step 2 we query $(\sum_{i \in I} \alpha_i f_i \leq t - 1, \sum_{i \in I} \alpha_i f_i \geq t)$ from $t = 1, \ldots, q^2 md$. For $t = 1$, the halfspace $\sum_{i \in I} \alpha_i f_i \geq 0$ is valid for the current polytope because the polytope belongs to the $[0,1]^n$ cube. For each subsequent query, $\sum_{i \in I} \alpha_i f_i \geq t - 1$ is valid because the previous query deduced $\sum_{i \in I} \alpha_i f_i \geq t - 1$. Similar arguments show that the remaining queries are also facelike. Thus, Lemma 14 completes the proof. ◀

We note that the CP refutations that result from Theorem 3 have a very particular structure: they are extremely long and narrow. Indeed, they have depth $n^{O(\log m)}$. We give a rough sketch of the argument: it is enough to show that most lines L_i in the CP refutation are derived using some previous line L_j with $j = O(i)$. This is because the final line would have depth proportional to the size of the proof. To see that the CP refutation satisfies this property, observe that for each node visited in the in-order traversal, the nodes in the right subproof π_R depend on the halfspace labelling the root, which in turn depends on the left subproof π_L.

5 Lower Bound on the Depth of Semantic CP Refutations

Our results from Section 3 suggest an interesting interplay between depth and size of Cutting Planes proofs. In particular, we note that there is a *trivial* depth n and exponential size refutation of any unsatisfiable CNF formula in Cutting Planes; however, it is easy to see that the Dadush–Tiwari proofs and our own quasipolynomial size CP proofs of Tseitin are also extremely deep (in particular, they are *superlinear*). Even in the stronger *Semantic* CP it is not clear that the depth of these proofs can be decreased. However, this does not hold for SP, which has quasi-polynomial size and poly-logarithmic depth refutations. This motivates Conjecture 6, regarding the existence of a “supercritical” trade-off between size and depth for Cutting Planes [11, 56]. The Tseitin formulas are a natural candidate for resolving this conjecture.

In this section we develop a new method for proving depth lower bounds which we believe should be more useful for resolving this conjecture. Our method works not only for CP but also for semantic CP. Using our technique, we establish the first linear lower bounds on the depth of Semantic CP refutations of the Tseitin formulas.

Lower bounds on the depth of *syntactic* CP refutations of Tseitin formulas were established by Buresh-Openheim et al. [16] using a rank-based argument. Our proof is inspired by their work, and so we describe it next. Briefly, their proof proceeds by considering a sequence of polytopes $P^{(0)} \supseteq \ldots \supseteq P^{(d)}$ where $P^{(i)}$ is the polytope defined by all inequalities that can be derived in depth i from the axioms in F. The goal is to show that $P^{(d)}$ is not empty. To do so, they show that a point $p \in P^{(i)}$ is also in $P^{(i+1)}$ if for every coordinate j such that $0 < p_j < 1$, there exists points $p^{(j,0)}, p^{(j,1)} \in P^{(i)}$ such that $p_k^{(j,b)} = b$ if $k = j$ and $p_k^{(j,b)} = p_k$ otherwise. The proof of this fact is syntactic: it relies on the careful analysis of the precise rules of CP.

When dealing with Semantic CP, we can no longer analyze a finite set of syntactic rules. Furthermore, it is not difficult to see that the aforementioned criterion for membership in $P^{(i+1)}$ is no longer sufficient for Semantic CP. We develop an analogous criterion for Semantic CP given later in this section. As well, we note that the definition of $P^{(i)}$ is not well-suited to studying the depth of bounded-size CP proofs like those in Conjecture 6 – there does not appear to be a useful way to limit $P^{(i)}$ to be a polytope derived by a bounded number of halfspaces. Therefore we develop our criterion in the language of lifting, which is more amenable to supercritical tradeoffs [11, 56].

Through this section we will work with the following *top-down* definition of Semantic CP.

▶ **Definition 19.** *Let F be an n-variate unsatisfiable CNF formula. An* sCP *refutation of F is a directed acyclic graph of fan-out ≤ 2 where each node v is labelled with a halfspace $H_v \subseteq \mathbb{R}^n$ (understood as a set of points satisfying a linear inequality) satisfying the following:*

1. Root. *There is a unique source node r labelled with the halfspace $H_v = \mathbb{R}^n$ (corresponding to the trivially true inequality $1 \geq 0$).*
2. Internal-Nodes. *For each non-leaf node u with children v, w, we have*
$$H_u \cap \{0,1\}^n \subseteq H_v \cup H_w.$$
3. Leaves. *Each sink node u is labeled with a unique clause $C \in F$ such that $H_v \cap \{0,1\}^n \subseteq C^{-1}(0)$.*

The above definition is obtained by taking a (standard) sCP proof and *reversing all inequalities*: now, a line is associated with the set of assignments *falsified* at that line, instead of the assignments *satisfying* the line.

To prove the lower bound we will need to find a long path in the proof. To find this path we will be taking a root-to-leaf walk down the proof while constructing a partial restriction $\rho \in \{0,1,*\}^n$ on the variables. For a partial restriction ρ, denote by $\mathsf{free}(\rho) := \rho^{-1}(*)$ and $\mathsf{fix}(\rho) := [n] \setminus \mathsf{free}(\rho)$. Let the *restriction* of H by ρ be the halfspace

$$H \restriction \rho := \{x \in \mathbb{R}^{\mathsf{free}(\rho)} : \exists \alpha \in H,\ \alpha_{\mathsf{fix}(\rho)} = \rho_{\mathsf{fix}(\rho)},\ \alpha_{\mathsf{free}(\rho)} = x\}.$$

It is important to note that $H \restriction \rho$ is itself a halfspace on the *free* coordinates of ρ.

One of our key invariants needed in the proof is the following.

▶ **Definition 20.** *A halfspace $H \subseteq \mathbb{R}^n$ is* good *if it contains the all-$\frac{1}{2}$ vector, that is, $(\frac{1}{2})^n = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}) \in H$.*

We will need two technical lemmas to prove the lower bounds. The first lemma shows that if a good halfspace H has its boolean points covered by halfspaces H_1, H_2, then one of the two covering halfspaces is also good modulo restricting a small set of coordinates.

▶ **Lemma 21.** *Let $H \subseteq \mathbb{R}^n$ be any good halfspace, and suppose $H \cap \{0,1\}^n \subseteq H_1 \cup H_2$ for halfspaces H_1, H_2. Then there is a restriction ρ and an $i = 1, 2$ such that $|\mathsf{fix}(\rho)| \leq 2$ and $H_i \restriction \rho$ is good.*

The second lemma shows that good halfspaces are *robust*, in the sense that we can restrict a good halfspace to another good halfspace while also satisfying any mod-2 equation.

▶ **Lemma 22.** *Let $n \geq 2$ and $H \subseteq \mathbb{R}^n$ be a good halfspace. For any $I \subseteq [n]$ with $|I| \geq 2$ and $b \in \{0,1\}$, there is a partial restriction $\rho \in \{0,1,*\}^n$ with $\mathsf{fix}(\rho) = I$ such that*

- $\bigoplus_{i \in I} \rho(x_i) = b$ *and*
- $H \restriction \rho \subseteq \mathbb{R}^{\mathsf{free}(\rho)}$ *is good.*

With these two lemmas one can already get an idea of how to construct a long path in the proof. Suppose we start at the root of the proof; the halfspace is $1 \geq 0$ (which is clearly good) and the restriction we maintain is $\rho = *^n$. We can use the first lemma to move from the current good halfspace to a good child halfspace while increasing the number of fixed coordinates by at most 2. However, we have no control over the two coordinates which are fixed by this move, and so we may fall in danger of falsifying an initial constraint. Roughly speaking, we will use the second lemma to satisfy constraints that are in danger of being falsified.

We delay the proofs of these technical lemmas to the end of the section, and first see how to prove the depth lower bounds.

5.1 Lifting Decision Tree Depth to Semantic CP Depth

As a warm-up, we show how to lift lower bounds on Resolution depth to Semantic CP depth by composing with a constant-width XOR gadget. If F is a CNF formula then we can create a new formula by replacing each variable z_i with an XOR of 4 new variables $x_{i,1}, \ldots, x_{i,4}$:

$$z_i := \mathsf{XOR}_4(x_{i,1}, \ldots, x_{i,4}) = x_{i,1} \oplus \cdots \oplus x_{i,4}.$$

We call z_i the *unlifted* variable associated with the output of the XOR_4 *gadget* applied to the i-th *block* of variables. Formally, let $\mathsf{XOR}_4^n : \{0,1\}^{4n} \to \{0,1\}^n$ be the application of XOR_4 to each 4-bit block of a $4n$-bit string. Let $F \circ \mathsf{XOR}_4^n$ denote the formula obtained by performing this substitution on F and transforming the result into a CNF formula in the obvious way.

The main result of this section is the following.

▶ **Theorem 23.** *For any unsatisfiable CNF formula F,*

$$\mathsf{depth}_{\mathsf{sCP}}(F \circ \mathsf{XOR}_4^n) \geq \frac{1}{2}\mathsf{depth}_{\mathsf{Res}}(F).$$

Key to our lower bound will be the following characterization of Resolution depth by *Prover-Adversary* games.

▶ **Definition 24.** *The* Prover–Adversary *game associated with an n-variate formula F is played between two competing players, Prover and Adversary. The game proceeds in rounds, where in each round the state of the game is recorded by a partial assignment $\rho \in \{0,1,*\}^n$ to the variables of F.*

Initially the state is the empty assignment $\rho = *^n$. Then, in each round, the Prover chooses an $i \in [n]$ with $\rho_i = *$, and the Adversary chooses $b \in \{0,1\}$. The state is updated by $\rho_i \leftarrow b$ and play continues. The game ends when the state ρ falsifies an axiom of F.

It is known [55] that $\mathsf{depth}_{\mathsf{Res}}(F)$ *is exactly the smallest d for which there is a Prover strategy that ends the game in d rounds, regardless of the strategy for the Adversary.*

The proof of Theorem 23 will follow by using an optimal Adversary strategy for F to construct a long path in the Semantic CP proof of $F \circ \mathsf{XOR}_4^n$. Crucially, we need to understand how halfspaces H transform under XOR_4^n:

$$\mathsf{XOR}_4^n(H) := \{z \in \{0,1\}^n : \exists x \in H \cap \{0,1\}^{4n}, \mathsf{XOR}_4^n(x) = z\}.$$

As we have already stated, we will maintain a partial assignment $\rho \in \{0,1,*\}^{4n}$ on the $4n$ *lifted* variables. However, in order to use the Adversary, we will need to convert ρ to a partial assignment on the n *unlifted* variables. To perform this conversion, for any $\rho \in \{0,1,*\}^{4n}$ define $\mathsf{XOR}_4^n(\rho) \in \{0,1,*\}^n$ as follows: for each block $i \in [n]$, define

$$\mathsf{XOR}_4^n(\rho)_i = \begin{cases} \mathsf{XOR}_4(\rho(x_{i,1}), \ldots, \rho(x_{i,4})) & \text{if } (i,j) \in \mathsf{fix}(\rho) \text{ for } j \in [4], \\ * & \text{otherwise.} \end{cases}$$

We are now ready to prove Theorem 23. Fix any Semantic CP refutation of $F \circ \mathsf{XOR}_4^n$, and suppose that there is a strategy for the Adversary in the Prover-Adversary game of F certifying that F requires depth d. Throughout the walk, we maintain a partial restriction $\rho \in \{0,1,*\}^{4n}$ to the lifted variables satisfying the following three invariants with respect to the current visited halfspace H.

- *Block Closed.* In every block either all variables in the block are fixed or all variables in the block are free.
- *Good Halfspace.* $H \restriction \rho$ is good.
- *Strategy Consistent.* The unlifted assignment $\mathsf{XOR}_4^n(\rho)$ does not falsify any clause in F.

Initially, we set $\rho = *^{4n}$ and the initial halfspace is $1 \geq 0$, so the pair (H, ρ) trivially satisfy the invariants. Suppose we have reached the halfspace H in our walk and ρ is a restriction satisfying the invariants. We claim that H cannot be a leaf. To see this, suppose that H is a leaf, then by definition $H \cap \{0,1\}^{4n} \subseteq C^{-1}(0)$ for some clause $C \in F \circ \mathsf{XOR}_4^n$. By the definition of the lifted formula, this implies that $\mathsf{XOR}_4^n(H) \subseteq D^{-1}(0)$ for some clause $D \in F$. Since (H, ρ) satisfy the invariants, the lifted assignment $\mathsf{XOR}_4^n(\rho)$ does not falsify D, and so by the block-closed property it follows that there must be a variable $z_i \in D$ such that all lifted variables in the block i are free under ρ. But then applying Lemma 22 to the block of variables $\{x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}\}$, we can extend ρ to a partial assignment ρ' such that $z_i = \mathsf{XOR}_4(\rho(x_{i,1}), \rho(x_{i,2}), \rho(x_{i,3}), \rho(x_{i,4}))$ satisfies D. But $H \restriction \rho'$ is a projection of $H \restriction \rho$ and so this contradicts that $\mathsf{XOR}_4^n(H)$ violates D.

It remains to show how to take a step down the proof. Suppose that we have taken $t < d/2$ steps down the Semantic CP proof, the current node is labelled with a halfspace H, and the partial assignment ρ satisfies the invariants. If H has only a single child H_1, then $H \cap \{0,1\}^{4n} \subseteq H_1 \cap \{0,1\}^{4n}$ and ρ will still satisfy the invariants for H_1. Otherwise, if H has two children H_1 and H_2 then applying Lemma 21 to the halfspaces $H \restriction \rho, H_1 \restriction \rho, H_2 \restriction \rho$ we can find an $i \in \{1,2\}$ and a restriction τ such that $H_i \restriction (\rho\tau)$ is good and τ restricts at most 2 extra coordinates. Let $i_1, i_2 \in [n]$ be the two blocks of variables in which τ restricts variables, and note that it could be that $i_1 = i_2$.

Finally, we must restore our invariants. We do this in the following three step process.

- Query the Adversary strategy at the state $\mathsf{XOR}_4^n(\rho)$ on variables z_{i_1}, z_{i_2} and let $b_1, b_2 \in \{0,1\}$ be the responses.
- For $i = i_1, i_2$ let I_i be the set of variables free in the block i, and note that $|I_i| \geq 2$. Apply Lemma 22 to $H \restriction (\rho\tau)$ and I_i to get new restrictions ρ_{i_1}, ρ_{i_2} so that blocks i_1 and i_2 both take values consistent with the Adversary responses b_1, b_2.
- Update $\rho \leftarrow \rho\tau\rho_{i_1}\rho_{i_2}$.

By Lemma 22 the new restriction ρ satisfies the block-closed and the good halfspace invariants. At each step we fix at most two blocks of variables, and thus the final invariant is satisfied as long as $t < d/2$. This completes the proof.

5.2 Semantic CP Depth Lower Bounds for Unlifted Formulas

Next we show how to prove depth lower bounds directly on *unlifted* families of $\mathbb{F}_2$-linear equations. The strength of these lower bounds will depend directly on the expansion of the underlying constraint-variable graph of F.

Throughout this section, let F denote a set of $\mathbb{F}_2$-linear equations. In a Semantic CP proof, we must encode F as a CNF formula, but while proving the lower bound we will instead work with the underlying system of equations. For a set F of $\mathbb{F}_2$-linear equations let $G_F := (F \cup V, E)$ be the bipartite *constraint-variable* graph defined as follows. Each vertex in F corresponds to an equation in F and each vertex in V correspond to variables x_i. There is an edge $(C_i, x_j) \in E$ if x_j occurs in the equation C_i. For a subset of vertices $X \subseteq F \cup V$ define the *neighbourhood* of X in G_F as $\Gamma(X) := \{v \in F \cup V : \exists u \in X, (u,v) \in E\}$.

▶ **Definition 25.** *For a bipartite graph $G = (U \cup V, E)$ the* boundary *of a set $W \subseteq U$ is*

$$\delta(W) := \{v \in V : |\Gamma(v) \cap W| = 1\}.$$

The boundary expansion *of a set $W \subseteq U$ is $|\delta(W)|/|W|$. The graph G is a (r,s)-*boundary expander *if the boundary expansion of every set $W \subseteq U$ with $|W| \leq r$ has boundary expansion at least s.*

If F is a system of linear equations then we say that F is an (r,s)-boundary expander if its constraint graph G_F is. The main result of this section is the following theorem, analogous to Theorem 23.

▶ **Theorem 26.** *For any system of $\mathbb{F}_2$-linear equations F that is an $(r, s+3)$-boundary expander,*

$$\mathsf{depth}_{\mathsf{sCP}}(F) \geq rs/2.$$

The proof of this theorem follows the proof of Theorem 23 with some small changes. As before, we will maintain a partial assignment $\rho \in \{0,1,*\}^n$ that will guide us on a root-to-leaf walk through a given Semantic CP proof; we also require that each halfspace H that we visit is *good* relative to our restriction ρ. Now our invariants are (somewhat) simpler: we will only require that $F \restriction \rho$ is a sufficiently good boundary expander.

We first prove an auxiliary lemma that will play the role of Lemma 22 in the proof of Theorem 26. We note that it follows immediately from Lemma 22 and boundary expansion.

▶ **Lemma 27.** *Suppose F is a system of $\mathbb{F}_2$-linear equations that is an (r,s)-boundary expander for $s > 1$, and suppose $F' \subseteq F$ with $|F'| \leq r$. Let H be a good halfspace. Then there exists a $\rho \in \{0,1,*\}^n$ with $\mathsf{fix}(\rho) = \Gamma(F')$ such that*

- *F' is satisfied by ρ, and*
- *$H \restriction \rho$ is good.*

Proof. We first use expansion to find, for each constraint $C_i \in F'$, a pair of variables $y_{i,1}, y_{i,2}$ that are in C_i's boundary. To do this, first observe that $|\delta(F')| \geq s|F'| > |F'|$ by the definition of boundary expansion. The pigeonhole principle then immediately implies that there are variables $y_{i,1}, y_{i,2} \in \delta(F')$ and a constraint $C_i \in F'$ such that $y_{i,1}, y_{i,2} \in C_i$. Since $y_{i,1}, y_{i,2}$ do not occur in $F' \setminus \{C_i\}$, it follows that $F' \setminus \{C_i\}$ is still an (r, s)-boundary expander. So, we update $F' = F' \setminus \{C_i\}$ and repeat the above process.

When the process terminates, we have for each constraint $C_i \in F'$ a pair of variables $y_{i,1}, y_{i,2}$ that occur *only* in C_i. Write the halfspace $H = \sum_i w_i x_i \geq c$, and let $I = \Gamma(F') \setminus \bigcup_{i \in I} \{y_{i,1}, y_{i,2}\}$ be the set of variables occurring in F' that were not collected by the above process. We define a partial restriction ρ with $\mathsf{fix}(\rho) = I$ that depends on $|I|$ as follows.

- If $|I| = 0$ then $\rho = *^n$.
- If $I = \{x_i\}$ then define $\rho(x_i) = 1$ if $w_i \geq 0$ and $\rho(x_i) = 0$ otherwise, and for all other variables set $\rho(x) = *$.
- If $|I| > 2$ then apply Lemma 22 to generate a partial restriction ρ with $\mathsf{fix}(\rho) = I$ that sets the XOR of I arbitrarily.

Observe that $H \restriction \rho$ is good. The only non-trivial case is when $|I| = 1$, but, in this case we observe

$$(H \restriction \rho)((1/2)^{n-1}) = w_i \rho(x_i) + \sum_{j \neq i} w_i/2 \geq \sum_i w_i/2 \geq c,$$

where we have used that H is good and the definition of ρ.

Next we extend ρ as follows: for each $i = 1, 2, \ldots, |F'|$ apply Lemma 22 to $I_i = \{y_{i,1}, y_{i,2}\}$ to generate a partial restriction ρ_i with $\mathsf{fix}(\rho_i) = I_i$ so that the constraint $C_i \restriction \rho\rho_1 \cdots \rho_{i-1}$ is satisfied by ρ_i. Observe that this is always possible since I_i is in the boundary of C_i. Finally, we update $\rho \leftarrow \rho\rho_1 \cdots \rho_{|F'|}$. It follows by Lemma 22 that F' is satisfied by ρ and $H \restriction \rho$ is good. ◀

We are now ready to prove Theorem 26. Fix any Semantic CP refutation of F and let n be the number of variables. We take a root-to-leaf walk through the refutation while maintaining a partial assignment $\rho \in \{0, 1, *\}^n$ and an integer valued parameter $k \geq 0$. Throughout the walk we maintain the following invariants with respect to the current halfspace H:

- *Good Expansion.* $F \restriction \rho$ is a (k, t)-boundary expander with $t > 3$.
- *Good Halfspace.* $H \restriction \rho$ is good.
- *Consistency.* The partial assignment ρ does not falsify any clause of F.

Initially, we set $k = r$, $\rho = *^n$, and $t = s + 3$, so the invariants are clearly satisfied since F is an $(r, s + 3)$-expander. So, suppose that we have reached a halfspace H in our walk, and let k, ρ be parameters satisfying the invariants. We first observe that if $k > 0$ then H cannot be a sink node of the proof. To see this, it is enough to show that H contains a satisfying assignment for each equation $C \in F$. Because $H \restriction \rho$ is non-empty (since it is good) there exists a satisfying assignment in H for every equation satisfied by ρ, so, assume that C is not satisfied by ρ. In this case, since $F \restriction \rho$ is a (k, t)-expander for $k > 0$ we can apply Lemma 27 to $\{C\}$ and $H \restriction \rho$ and obtain a partial restriction τ with $\mathsf{fix}(\tau) = \Gamma(C)$ such that τ satisfies C. It follows that H is not a leaf.

Next, we show how to take a step down the proof while maintaining the invariants. If H has only a single child H_1, then $H \subseteq H_1$ and we can move to H_1 without changing ρ or k. Otherwise, let the children of H be H_1 and H_2. Applying Lemma 21 to $H \restriction \rho, H_1 \restriction \rho, H_2 \restriction \rho$ we get a partial restriction τ and an $i \in \{1, 2\}$ such that $H_i \restriction \rho\tau$ is good and $|\mathsf{fix}(\tau)| \leq 2$. Due to this latter fact, since $F \restriction \rho$ is a (k, t)-expander it follows that $F \restriction \rho\tau$ is a $(k, t - 2)$-expander in the worst case. Observe that since $t > 3$ it follows that $F \restriction \rho\tau$ still satisfies the consistency invariant. It remains to restore the expansion invariant.

To restore the expansion invariant, let W be the largest subset of equations such that $|W| \leq k$ and W has boundary expansion at most 3 in $F \restriction \rho\tau$, and note that W has boundary expansion at least $t - 2 > 1$. Applying Lemma 27, we can find a restriction ρ' such that $W \restriction \rho\tau\rho'$ is satisfied, and $H \restriction \rho\tau\rho'$ is a good halfspace. Since W is the largest subset with expansion at most 3, it follows that $F \restriction \rho\tau\rho'$ is now a $(k - |W|, t')$-boundary expander with $t' > 3$. Suppose otherwise, then there exists a subset of equations W' which has boundary expansion at most 3 in $F \restriction \rho\tau\rho'$. Then $W \cup W'$ would have had boundary expansion at most 3 in $F \restriction \rho\tau$, contradicting the maximality of W. Now update $\rho \leftarrow \rho\tau\rho'$ and $k \leftarrow k - |W|$. Finally, we halt the walk if $k = 0$.

We now argue that this path must have had depth at least $rs/2$ upon halting. Assume that we have taken t steps down the proof. For each step $i \leq t$ let W_i be the set of equations which lost boundary expansion during the ith cleanup step. Note that $W_i \cap W_j = \emptyset$ for every $i \neq j$. Let $W^* = \cup_{i=1}^t W_i$, note that $|W^*| = r$ because at the ith step we decrease k by $|W_i|$. Furthermore, at the end of the walk, W^* has no neighbours and therefore no boundary in $F \restriction \rho$. Before the start of the ith cleanup step, W_i has at most $3|W_i|$ boundary variables. Therefore, at most $3|W^*| = 3r$ boundary variables were removed during the cleanup step. Since F started as an $(r, s+3)$-boundary expander, it follows that W^* had at least $r(s+3)$ boundary variables at the start of the walk. But, since *all* variables have been removed from the boundary by the end, this means that rs variables must have been removed from the boundary during the move step. Thus, as each move step sets at most 2 variables, it follows that $t \geq rs/2$ before the process halted.

5.3 Proof of Lemma 21 and Lemma 22

In this section we prove our two key technical lemmas: Lemma 21 and Lemma 22. We begin by proving Lemma 22 as it is simpler.

Proof of Lemma 22. Let H be represented by $\sum_{i\in[n]} w_i x_i \geq c$ and suppose without loss of generality that $c \geq 0$ and that $I = \{1, \ldots, k\}$. Let the weights of I in H be ordered $|w_1| \geq |w_2| \geq \ldots |w_k|$. Define ρ by setting $\rho(x_i) = *$ for $i \notin I$, for $i \leq k-1$ set $\rho(x_i) = 1$ if $w_i \geq 0$ and $\rho(x_i) = 0$ otherwise, and set $\rho(x_k)$ so that $\bigoplus_{i\in I} \rho(x_i) = b$. Clearly the parity constraint is satisfied, we show that $H \restriction \rho$ is good. This follows by an easy calculation:

$$\begin{aligned}(H \restriction \rho)((1/2)^{[n]\setminus I}) &= w_{k-1}\rho(x_{k-1}) + w_k\rho(x_k) + \sum_{i\leq k-2} w_i\rho(x_i) + \sum_{i\geq k+1} w_i/2 \\ &\geq w_{k-1}/2 + w_k/2 + \sum_{i\leq k-2} w_i\rho(x_i) + \sum_{i\geq k+1} w_i/2 \\ &\geq \sum_{i\in[n]} w_i/2 \geq c\end{aligned}$$

where the first inequality follows by averaging since $|w_{k-1}| \geq |w_k|$, and the final inequality follows since H is good. ◀

In the remainder of the section we prove Lemma 21. It will be convenient to work over $\{-1,1\}^n$ rather than $\{0,1\}^n$, so, we restate it over this set and note that we can move between these basis by using the bijection $v \mapsto (1-v)/2$.

▶ **Lemma 28.** *Let $H \in \mathbb{R}^n$ be a halfspace such that $0^n \in H$ and suppose that $H \cap \{-1,1\}^n \subseteq H_1 \cup H_2$. Then one of H_1 or H_2 contains a point $y \in \{-1,0,1\}^n$ such that y has at most two coordinates in $\{-1,1\}$.*

The key ingredient in our proof of Lemma 28 is the following simple topological lemma, which will allow us to find a well-behaved point lying on a 2-face of the $\{-1,1\}^n$ cube

▶ **Definition 29** (2-face). *A 2-face of the n-cube with vertices $\{-1,1\}^n$ are the 2-dimensional 2-by-2 squares spanned by four vertices of the cube that agree on all but two coordinates. That is, a two face is a set $A \subseteq [-1,1]^n$ such that there exists $\rho \in \{-1,1,*\}^n$ with $|\mathsf{free}(\rho)| = 2$ and $A = [-1,1]^n \restriction \rho$.*

▶ **Lemma 30.** *Let $w_1, w_2 \in \mathbb{R}^n$ be any pair of non-zero vectors, then we can find a vector $v \in \mathbb{R}^n$ orthogonal to w_1, w_2, such that v lies on a 2-face.*

Proof. We will construct the vector v iteratively by rounding one coordinate at a time to a $\{-1,1\}$-value until v contains exactly $n-2$ coordinates fixed to $\{-1,1\}$. At each step, we will maintain that $v \in [-1,1]^n$ and that v is orthogonal to w_1 and w_2. Therefore when the process halts v will lie on a 2-face.

Initially, set $v = 0^n$ and observe that the invariants are satisfied. Suppose that we have constructed a vector v that is orthogonal to w_1 and w_2, all of its coordinates belong to $[-1,1]$, and exactly $i < n-2$ of its coordinates belong to $\{-1,1\}$; suppose w.l.o.g. that they are the first i coordinates. We will show how to "booleanize" an additional coordinate of v. Let u be any non-zero vector that is orthogonal to $\{w_1, w_2, e_1, \dots, e_i\}$, where e_j is the jth standard basis vector. Begin moving from v in the direction of u and let $\alpha > 0$ be the smallest value such that one of the coordinates $j > i$ of $v + \alpha u$ is in $\{-1,1\}$. We verify that the following properties hold:

1. The first i coordinates of $v + \alpha u$ are in $\{-1,1\}$. This follows because we moved in a direction that is orthogonal to $e_1, \dots, e_i$.
2. $v + \alpha u$ is orthogonal to w_1 and w_2. Let w be either of the vectors w_1 or w_2 and observe that $v_{i+1}w = v_i w + \alpha(uw) = 0$, where the final equality follows because w is orthogonal to v_i by induction and to u by assumption.

Finally, set v to be $v + \alpha u$. ◀

Proof of Lemma 28. Let the children H_1 and H_2 of H be given by the halfspaces $w_1 x \geq b_1$ and $w_2 x \geq b_2$ respectively. By Lemma 30 we can find a vector v which is orthogonal to w_1 and w_2, and which lies on some 2-face F of the $[-1,1]^n$ cube corresponding to some restriction $\rho \in \{0,1,*\}^n$. Then, v lies in (at least) one of the four 1-by-1 quadrants of the 2-face, $[0,1]^2$, $[0,1] \times [-1,0]$, $[-1,0] \times [0,1]$, or $[-1,0]^2$; suppose that v lies in the $[-1,0] \times [0,1]$ quadrant of F, the other cases will follow by symmetry (see Figure 3).

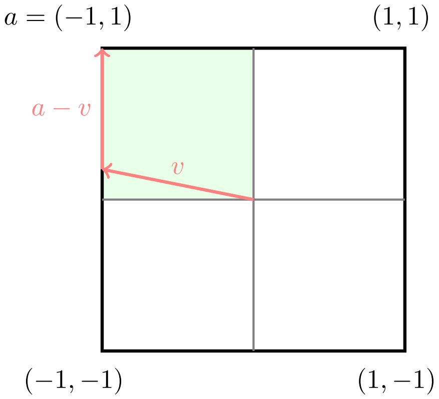


Figure 3 A 2-face of the n-cube together with a depiction of the booleanizing process.

Let $a \in \mathbb{R}^n$ be the vector corresponding to the $(-1, 1)$ corner of F, i.e., a is ρ extended by setting the two free bits to -1 and 1. By symmetry and the fact that H is good (and therefore $0^n \in H$), we can assume that a is contained in H – otherwise, simply exchange a and v for $-a$ and $-v$. Since $H \cap \{-1,1\}^n \subseteq H_1 \cup H_2$ and $a \in \{-1,1\}^n$, it follows that a is in one of H_1 or H_2. Assume that $a \in H_1$; that is, $w_1 a \geq b_1$. Our goal is to construct a vector $y \in H_1$ that satisfies the statement of the lemma. Consider the following two cases:

(i) If $w_1(a - v) \leq 0$, then it follows that $y := 0^n \in H_1$. Indeed, $w_1 y = w_1 v \geq w_1 a \geq b_1$, where first equality follows because w_1 and p are orthogonal by assumption, and the final inequality follows because $a \in H_1$.

(ii) Otherwise, we have that $w_1(a-v) > 0$. We construct a point that satisfies the statement of the lemma as follows. First, note that since $a, v \in F$, it follows that the vector $a - v$ has at most two non-zero coordinates. Beginning at the origin 0^n, move in the direction $a - v$ until a free coordinate coordinate becomes fixed to -1 or 1; that is, let $\alpha > 0$ be the minimum value such that $\alpha(a - v)$ has at most one coordinate which is not $\{-1, 1\}$-valued. Since both a and v belong to the same 1×1 quadrant of the 2-face, $\|a - v\|_\infty \leq 1$ and so $\alpha \geq 1$. We can then verify that $\alpha(a - v) \in H_1$, since

$$w_1 \alpha(a - v) = \alpha(w_1 a) - 0 \geq w_1 a \geq b_1,$$

where we have used the fact that v is orthogonal to w_1 and $\alpha \geq 1$. Finally, since $\alpha(a - v) \in H_1$ we can round the final non-zero coordinate to -1 or 1; since H_1 is a halfspace one of the two vectors will remain in H_1. ◀

5.4 Applications

We now use the theorems from the previous sections to obtain several concrete lower bounds. First, we give strong depth lower bounds for sCP proofs of Tseitin formulas on expander graphs.

▶ **Theorem 31.** *There exists a graph G and labelling $\ell : V \to \{0, 1\}$ such that any* sCP *refutation of* Tseitin(G, ℓ) *requires depth $\Omega(n)$.*

Proof. A graph $G = (V, E)$ is a γ-*vertex expander* if

$$\min \{|\Gamma(W)| : W \subseteq V, |W| \leq |V|/2\} \geq \gamma |W|,$$

where $\Gamma(W)$ is the neighbourhood of W. We claim that if G is a γ-vertex expander then any Tseitin formula over G is a $(n/2, \gamma)$-boundary expander. Fix any subset W of the equations with $|W| \leq n/2$. By the definition of vertex expansion we have that $|\Gamma(W)| \geq \gamma|W|$, and since each variable is contained in exactly two constraints, it follows that the boundary of W in Tseitin(G, ℓ) has size at least $|\delta(W)| \geq \gamma|W|$. The result then follows from Theorem 26 and the existence of strong vertex expanders G (e.g. d-regular Ramanujan graphs are at least $d/4$-vertex expanders, and exist for all d and n [48]). ◀

Next, we give lower bounds on the depth of Semantic CP refutations of random k-XOR and random k-CNF formulas for constant k.

▶ **Definition 32.** *Let* XOR(m, n, k) *be the distribution on random k-*XOR *formulas obtained by sampling m equations from the set of all* mod 2 *linear equations with exactly k variables.*

▶ **Theorem 33.** *The following holds for Semantic* CP :

1. *For any $k \geq 6$ there exists $m = O(n)$ such that $F \sim \mathsf{XOR}(m, n, k)$ requires refutations of depth at least $\Omega(n)$ with high probability.*
2. *For any $k \geq 6$ there exists $m = O(n)$ such that $F \sim \mathcal{F}(m, n, k)$ requires refutations of depth at least $\Omega(n)$ with high probability.*

Proof. We first prove (1) and obtain (2) via a reduction. Fix $m = O(n)$ so that F is unsatisfiable with high probability. For any constant k, δ and $m = O(n)$, $F \sim \mathsf{XOR}(m, n, k)$ is an $(\alpha n, k - 2 - 2\delta)$-boundary expander for some $\alpha > 0$ (see e.g. [16, 21]). Thus, setting $k \geq 6$ and ε to be some small constant, the boundary expansion of G_F is at least 3. By Theorem 26, F requires depth $\Omega(n)$ to refute in Semantic CP with high probability.

The proof of (2) is via a reduction from $\mathcal{F}(m, n, k)$ to $\mathsf{XOR}(m, n, k)$. Every k-clause occurs in the clausal encoding of exactly one k-XOR constraint. It follows that from any k-CNF formula F we can generate a k-XOR formula whose clausal expansion F' contains F as follows: for each clause $C \in F$, if C contains an even (odd) number of positive literals then add to F' every clause on the variables of C which contains an even (odd) number of positive literals. The resulting F' is the clausal encoding of a set of $|F|$ k-XOR constraints. As there is a unique k-XOR consistent with the clauses of F, we can define the distribution $\mathsf{XOR}(m, n, k)$ equivalently as follows:

1. Sample $F \sim \mathcal{F}(m, n, k)$,
2. Return the k-XOR F' generated from F according to the aforementioned process.

It follows that the complexity of refuting $F \sim \mathcal{F}(m, n, k)$ is at least that of refuting $F' \sim \mathsf{XOR}(m, n, k)$ and (2) follows from (1) with the same parameters. ◀

Finally, we use Theorem 26 to extend the integrality gaps from [16] to sCP by essentially the same argument. For a linear program with constraints given by a system of linear inequalities $Ax \leq b$, the *r-round* sCP *relaxation* adds all inequalities that can be derived from $Ax \leq b$ by a depth-r sCP proof. We show that the r-round Semantic sCP linear program relaxation cannot well-approximate the number of satisfying assignments to a random k-SAT or k-XOR instance.

First we define our LP relaxations. Suppose that F is a k-CNF formula with m clauses $C_1, C_2, \ldots, C_m$ and n variables $x_1, x_2, \ldots, x_n$. If $C_i = \bigvee_{i \in P} x_i \vee \bigvee_{i \in N} \overline{x}_i$ then let $E(C_i) = \sum_{i \in P} x_i + \sum_{i \in N} 1 - x_i$. We consider the following LP relaxation of F:

$$\begin{aligned} & \max \sum_{i=1}^{m} y_i \\ \text{subject to} \quad & E(C_i) \geq y_i \quad \forall i \in [m] \\ & 0 \leq x_j \leq 1 \quad \forall j \in [n] \\ & 0 \leq y_i \leq 1 \quad \forall i \in [m] \end{aligned}$$

If F is a k-XOR formula with m constraints and n variables then we consider the above LP relaxation obtained by writing F as a k-CNF. Finally, recall that the *integrality gap* is the ratio between the optimal integral solution to a linear program and the optimal solution produced by the LP.

▶ **Theorem 34.** *For any $\varepsilon > 0$ and $k \geq 6$,*

1. *There is $\kappa > 0$ and $m = O(n)$ such that for $F \sim \mathsf{XOR}(m, n, k)$ the integrality gap of the κn-round* sCP *relaxation of F is at least $(2 - \varepsilon)$ with high probability.*
2. *There is $\kappa > 0$ and $m = O(n)$ such that for $F \sim \mathcal{F}(m, n, k)$ the integrality gap of the κn-round* sCP *relaxation of F is at least $2^k/(2^k - 1) - \varepsilon$ with high probability.*

Proof. Let $F \sim \mathsf{XOR}(m, n, k)$ and let Y_i be the event that the ith constraint is falsified by a uniformly random assignment. Let $\delta := \varepsilon/(2-\varepsilon)$, then by a multiplicative Chernoff Bound, the probability that a uniformly random assignment satisfies at least a $1/(2-\varepsilon)$-fraction of F is $\Pr[\sum_{i \in [m]} Y_i \geq (1+\delta)\frac{m}{2}] \leq 2^{-\delta m/6}$. By a union bound, the probability that there exists an assignment satisfying at least a $1/(2-\varepsilon)$ fraction of F is $2^{n-\delta m/6}$ which is exponentially small when $m \geq 7n(2-\varepsilon)/\varepsilon$.

On the other hand, consider the partial restriction to the LP relaxation of F that sets $y_i = 1$ for all $i \in [m]$. Setting $m \geq 7n(2-\varepsilon)/\varepsilon$ large enough, by Theorem 33 there some $\kappa > 0$ such that with high probability F requires depth κn. Hence, the κn round Semantic CP LP relaxation is non-empty, and there is a satisfying assignment $\alpha \in \mathbb{R}^n$. Thus $\alpha \cup \{y_i = 1\}$ satisfies all constraints of $\max(F)$.

The second result follows by an analogous argument. ◀

6 Conclusion

We end by discussing some problems left open by this paper. The most obvious of which is a resolution to Conjecture 6. A related question is whether supercritical size-depth tradeoffs can be established for monotone circuits? Indeed, current size lower bound techniques [32,33,39,54] are via reduction to monotone circuit lower bounds. As a first step towards both of these, can one prove a supercritical size-depth tradeoff for a weaker proof system such as resolution?

The simulation results presented in Section 3 leave open several questions regarding the relationship between SP and CP. First, the simulation of SP* by CP incurs a significant blowup in the coefficient size due to Shrijver's lemma. It would be interesting to understand whether SP* can be quasi-polynomially simulated by CP*; that is, whether this blowup in the size of the coefficients is necessary.

The most obvious question left open by these simulations is whether CP can polynomially simulate SP, or even *polynomially* simulate SP*. Similarly, what are the relationships of both SP and CP, to (bounded-coefficient) R(CP), the system which corresponds to dag-like SP. R(CP) can polynomially simulate DNF resolution, and therefore has polynomial size proofs of the Clique-Colouring formulas, for cliques of size $\Omega(\sqrt{n})$ and colourings of size $o(\log^2 n)$ [4]. Quasi-polynomial lower bounds on the size of CP refutations are known for this range of parameters and this rules out a polynomial simulation by Cutting Planes; however, a quasi-polynomial simulation may be possible. A potential approach to resolving this question is to use the added expressibility of R(CP) over DNF resolution to extend the upper bound on Clique-Colouring to the range of parameters for which superpolynomial CP lower bounds are known.

References

1 Karen Aardal, Robert E. Bixby, Cor A. J. Hurkens, Arjen K. Lenstra, and Job W. Smeltink. Market split and basis reduction: Towards a solution of the cornuéjols-dawande instances. *INFORMS J. Comput.*, 12(3):192–202, 2000. doi:10.1287/ijoc.12.3.192.12635.

2 Karen Aardal and Arjen K. Lenstra. Hard equality constrained integer knapsacks. *Math. Oper. Res.*, 29(3):724–738, 2004. doi:10.1287/moor.1040.0099.

3 Michael Alekhnovich and Alexander A. Razborov. Satisfiability, branch-width and tseitin tautologies. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 593–603. IEEE Computer Society, 2002. doi:10.1109/SFCS.2002.1181983.

4 Albert Atserias, Maria Luisa Bonet, and Juan Luis Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Inf. Comput.*, 176(2):136–152, 2002. doi:10.1006/inco.2002.3114.

5 Albert Atserias, Massimo Lauria, and Jakob Nordström. Narrow proofs may be maximally long. *ACM Trans. Comput. Log.*, 17(3):19:1–19:30, 2016. doi:10.1145/2898435.

6 Boaz Barak, Fernando G. S. L. Brandão, Aram Wettroth Harrow, Jonathan A. Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 307–326, 2012. doi:10.1145/2213977.2214006.

7 Paul Beame, Christopher Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: superpolynomial lower bounds for superlinear space. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 213–232. ACM, 2012. doi:10.1145/2213977.2213999.

8 Paul Beame, Noah Fleming, Russell Impagliazzo, Antonina Kolokolova, Denis Pankratov, Toniann Pitassi, and Robert Robere. Stabbing planes. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 10:1–10:20, 2018. doi:10.4230/LIPIcs.ITCS.2018.10.

9 Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus: extended abstract. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 813–822. ACM, 2013. doi:10.1145/2488608.2488711.

10 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001. doi:10.1145/375827.375835.

11 Christoph Berkholz and Jakob Nordström. Supercritical space-width trade-offs for resolution. *SIAM J. Comput.*, 49(1):98–118, 2020. doi:10.1137/16M1109072.

12 Alexander Bockmayr, Friedrich Eisenbrand, Mark E. Hartmann, and Andreas S. Schulz. On the chvátal rank of polytopes in the 0/1 cube. *Discret. Appl. Math.*, 98(1-2):21–27, 1999. doi:10.1016/S0166-218X(99)00156-0.

13 Merve Bodur, Alberto Del Pia, Santanu S. Dey, Marco Molinaro, and Sebastian Pokutta. Aggregation-based cutting-planes for packing and covering integer programs. *Math. Program.*, 171(1-2):331–359, 2018. doi:10.1007/s10107-017-1192-x.

14 Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Comput. Complex.*, 10(4):261–276, 2001. doi:10.1007/s000370100000.

15 Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. *J. Symb. Log.*, 62(3):708–728, 1997. doi:10.2307/2275569.

16 Joshua Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi. Rank bounds and integrality gaps for cutting planes procedures. *Theory of Computing*, 2(4):65–90, 2006. doi:10.4086/toc.2006.v002a004.

17 Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. Syst. Sci.*, 62(2):267–289, 2001. doi:10.1006/jcss.2000.1726.

18 Vasek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973. doi:10.1016/0012-365X(73)90167-2.

19 Vašek Chvátal. *Cutting-plane proofs and the stability number of a graph.* Inst. für Ökonometrie und Operations Research, Rhein. Friedrich-Wilhelms-Univ., 1984.

20 Vašek Chvátal, William Cook, and Mark Hartmann. On cutting-plane proofs in combinatorial optimization. *Linear algebra and its applications*, 114:455–499, 1989.

21 Vasek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988. doi:10.1145/48014.48016.

22 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. doi:10.2307/2273702.

23 William J. Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987. doi:10.1016/0166-218X(87)90039-4.

24 Gérard Cornuéjols and Yanjun Li. On the rank of mixed 0, 1 polyhedra. *Math. Program.*, 91(2):391–397, 2002. doi:10.1007/s101070100250.

25 Daniel Dadush and Samarth Tiwari. On the complexity of branching proofs. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 34:1–34:35. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.34.

26 Adnan Darwiche. Recursive conditioning. *Artif. Intell.*, 126(1-2):5–41, 2001. doi:10.1016/S0004-3702(00)00069-2.

27 Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960. doi:10.1145/321033.321034.

28 Rina Dechter. Bucket elimination: A unifying framework for processing hard and soft constraints. *ACM Comput. Surv.*, 28(4es):61, 1996. doi:10.1145/242224.242302.

29 Friedrich Eisenbrand and Andreas S. Schulz. Bounds on the chvátal rank of polytopes in the 0/1-cube. In Gérard Cornuéjols, Rainer E. Burkard, and Gerhard J. Woeginger, editors, *Integer Programming and Combinatorial Optimization, 7th International IPCO Conference, Graz, Austria, June 9-11, 1999, Proceedings*, volume 1610 of *Lecture Notes in Computer Science*, pages 137–150. Springer, 1999. doi:10.1007/3-540-48777-8_11.

30 Yuval Filmus, Pavel Hrubes, and Massimo Lauria. Semantic versus syntactic cutting planes. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 35:1–35:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.STACS.2016.35.

31 Matteo Fischetti and Andrea Lodi. Local branching. *Math. Program.*, 98(1-3):23–47, 2003. doi:10.1007/s10107-003-0395-5.

32 Noah Fleming, Denis Pankratov, Toniann Pitassi, and Robert Robere. Random Θ(log n)-CNFs are hard for cutting planes. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 109–120, 2017. doi:10.1109/FOCS.2017.19.

33 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 902–911. ACM, 2018. doi:10.1145/3188745.3188838.

34 Michel X. Goemans and David P. Williamson. .879approximationn algorithms for max cut and max 2sat. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 422–431, New York, NY, USA, 1994. ACM. doi:10.1145/195058.195216.

35 Ralph E Gomory. An algorithm for integer solutions to linear programs. *Recent advances in mathematical programming*, 64(260-302):14, 1963.

36 Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is np-hard. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 68–77. ACM, 2020. doi:10.1145/3357713.3384248.

37 Dima Grigoriev. Tseitin's tautologies and lower bounds for nullstellensatz proofs. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 648–652. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743515.

38 Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theor. Comput. Sci.*, 259(1-2):613–622, 2001. doi:10.1016/S0304-3975(00)00157-2.

39 Pavel Hrubes and Pavel Pudlák. Random formulas, monotone circuits, and interpolation. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 121–131. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.20.

40 Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*, pages 220–228. IEEE Computer Society, 1994. doi:10.1109/LICS.1994.316069.

41 Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In Benjamin Kuipers and Bonnie L. Webber, editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA*, pages 203–208. AAAI Press / The MIT Press, 1997. URL: http://www.aaai.org/Library/AAAI/1997/aaai97-032.php.

42 Miroslav Karamanov and Gérard Cornuéjols. Branching on general disjunctions. *Math. Program.*, 128(1-2):403–436, 2011. doi:10.1007/s10107-009-0332-3.

43 Arist Kojevnikov. Improved lower bounds for tree-like resolution over linear inequalities. In João Marques-Silva and Karem A. Sakallah, editors, *Theory and Applications of Satisfiability Testing - SAT 2007, 10th International Conference, Lisbon, Portugal, May 28-31, 2007, Proceedings*, volume 4501 of *Lecture Notes in Computer Science*, pages 70–79. Springer, 2007. doi:10.1007/978-3-540-72788-0_10.

44 Jan Krajícek. Discretely Ordered Modules as a First-Order Extension of the Cutting Planes Proof System. *The Journal of Symbolic Logic*, 63(4):1582–1596, 1998.

45 Bala Krishnamoorthy and Gábor Pataki. Column basis reduction and decomposable knapsack problems. *Discret. Optim.*, 6(3):242–270, 2009. doi:10.1016/j.disopt.2009.01.003.

46 Neha Lodha, Sebastian Ordyniak, and Stefan Szeider. A SAT approach to branchwidth. *ACM Trans. Comput. Log.*, 20(3):15:1–15:24, 2019. doi:10.1145/3326159.

47 Ashutosh Mahajan and Theodore K Ralphs. Experiments with branching using general disjunctions. In *Operations Research and Cyber-Infrastructure*, pages 101–118. Springer, 2009.

48 Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families IV: bipartite ramanujan graphs of all sizes. *SIAM J. Comput.*, 47(6):2488–2509, 2018. doi:10.1137/16M106176X.

49 Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001*, pages 530–535. ACM, 2001. doi:10.1145/378239.379017.

50 Jonathan H. Owen and Sanjay Mehrotra. Experimental results on using general disjunctions in branch-and-bound for general-integer linear programs. *Comput. Optim. Appl.*, 20(2):159–170, 2001. doi:10.1023/A:1011207119557.

51 Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, California Institute of Technology, 2000.

52 Sebastian Pokutta and Andreas S. Schulz. On the rank of cutting-plane proof systems. In Friedrich Eisenbrand and F. Bruce Shepherd, editors, *Integer Programming and Combinatorial Optimization, 14th International Conference, IPCO 2010, Lausanne, Switzerland, June 9-11, 2010. Proceedings*, volume 6080 of *Lecture Notes in Computer Science*, pages 450–463. Springer, 2010. doi:10.1007/978-3-642-13036-6_34.

53 Sebastian Pokutta and Andreas S. Schulz. Integer-empty polytopes in the 0/1-cube with maximal gomory-chvátal rank. *Oper. Res. Lett.*, 39(6):457–460, 2011. doi:10.1016/j.orl.2011.09.004.

54 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

55 Pavel Pudlák. Proofs as games. *The American Mathematical Monthly*, 107(6):541–550, 2000. doi:10.1080/00029890.2000.12005233.

56 Alexander A. Razborov. A new kind of tradeoffs in propositional proof complexity. *J. ACM*, 63(2):16:1–16:14, 2016. doi:10.1145/2858790.

57 Alexander A. Razborov. On the width of semialgebraic proofs and algorithms. *Math. Oper. Res.*, 42(4):1106–1134, 2017. doi:10.1287/moor.2016.0840.

58 Thomas Rothvoß and Laura Sanita. 0/1 polytopes with quadratic chvátal rank. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 349–361. Springer, 2013.

59 Grant Schoenebeck. Linear level lasserre lower bounds for certain k-csps. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 593–602. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.74.

60 A. Schrijver. On cutting planes. In Peter L. Hammer, editor, *Combinatorics 79*, volume 9 of *Annals of Discrete Mathematics*, pages 291–296. Elsevier, 1980. doi:10.1016/S0167-5060(08)70085-2.

61 João P. Marques Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Trans. Computers*, 48(5):506–521, 1999. doi:10.1109/12.769433.

Separating ABPs and Some Structured Formulas in the Non-Commutative Setting

Prerona Chatterjee
Tata Institute of Fundamental Research, Mumbai, India

Abstract

The motivating question for this work is a long standing open problem, posed by Nisan [20], regarding the relative powers of algebraic branching programs (ABPs) and formulas in the non-commutative setting. Even though the general question remains open, we make some progress towards its resolution. To that effect, we generalise the notion of ordered polynomials in the non-commutative setting (defined by Hrubeš, Wigderson and Yehudayoff [11]) to define abecedarian polynomials and models that naturally compute them.

Our main contribution is a possible new approach towards resolving the $\mathsf{VF}_{\mathrm{nc}}$ vs $\mathsf{VBP}_{\mathrm{nc}}$ question, via lower bounds against abecedarian formulas. In particular, we show the following.

There is an explicit n^2-variate degree d abecedarian polynomial $f_{n,d}(\mathbf{x})$ such that

- $f_{n,d}(\mathbf{x})$ can be computed by an abecedarian ABP of size $O(nd)$;
- any abecedarian formula computing $f_{n,\log n}(\mathbf{x})$ must have size at least $n^{\Omega(\log \log n)}$.

We also show that a super-polynomial lower bound against abecedarian formulas for $f_{\log n,n}(\mathbf{x})$ would separate the powers of formulas and ABPs in the non-commutative setting.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Non-Commutative Formulas, Lower Bound, Separating ABPs and Formulas

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.7

Funding Research supported by the Department of Atomic Energy, Government of India, under project number RTI4001.

Acknowledgements We are thankful to Ramprasad Saptharishi, Mrinal Kumar, C. Ramya and especially Anamay Tengse for the discussions at various stages of this work. We would also like to thank Ramprasad Saptharishi, Anamay Tengse and Kshitij Gajjar for helping with the presentation of the paper. Finally, we would like to thank the anonymous reviewers for their valuable comments that have helped in improving the paper.

1 Introduction

Algebraic Circuit Complexity is the study of multivariate polynomials and their classification based on how hard it is to compute them, using various computational models. The most well studied model is that of algebraic circuits. These are directed acyclic graphs that use algebraic operations like addition and multiplication over some field or ring, to compute polynomials. When the underlying graph is only allowed to be a tree, the model is that of algebraic formulas. The central question in this area is whether the class VNP (algebraic analogue of the class NP) is contained in the class VP (algebraic analogue of the class P). Valiant [23] has shown that the permanent polynomial is complete for VNP, and therefore the VP vs VNP question essentially boils down to asking whether the $n \times n$ permanent can be computed by a $\mathsf{poly}(n)$-sized algebraic circuit.

In this paper, we are interested in polynomials that come from the non-commutative polynomial ring $\mathbb{F}\langle x_1, \ldots, x_n \rangle$, where the indeterminates do not commute with each other (that is, $xy \neq yx$ for indeterminates x, y). As a consequence, any monomial in a non-commutative polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is essentially a string over the alphabet $\{x_1, \ldots, x_n\}$. This is a natural restriction and there has been a long line of work that studies non-commutative computation beginning with the seminal work of Nisan [20][1].

It was shown by Hrubeš, Wigderson and Yehudayoff [10] that the non-commutative permanent polynomial is complete for the class $\mathsf{VNP}_{\mathrm{nc}}$ (the non-commutative version of VNP). Later Arvind, Joglekar and Raja [1] gave a natural polynomial that is complete for the class of n-variate non-commutative polynomials computable by $\mathsf{poly}(n)$-sized circuits (denoted by $\mathsf{VP}_{\mathrm{nc}}$). The question of whether the classes $\mathsf{VP}_{\mathrm{nc}}$ and $\mathsf{VNP}_{\mathrm{nc}}$ are different is the central open problem in the non-commuatative setting. Although the general question of showing lower bounds against non-commutative circuits remains open, there has been significant progress in restricted settings [17, 16, 15, 22, 8].

With respect to the general question, Hrubeš, Wigderson and Yehudayoff [11] showed that a sufficiently strong super-linear lower bound for the classical sum-of-squares problem implies a separation between $\mathsf{VP}_{\mathrm{nc}}$ and $\mathsf{VNP}_{\mathrm{nc}}$. In another related work, Carmosino, Impagliazzo, Lovett and Mihajlin [5] showed that proving mildly super-linear lower bounds against non-commutative circuits would imply exponential lower bounds against the same model.

One motivation for studying non-commutative computation is that it is possibly easier to prove strong lower bounds in this setting as compared to the usual commutative setting. At least intuitively, it seems harder to *cancel* monomials once they have been calculated when commutativity is not allowed amongst the variables.

For example, the $n \times n$ determinant can be computed by an $O(n^3)$ algebraic circuit, but to the best of our knowledge there is no circuit for the non-commutative determinant of size $2^{o(n)}$. In fact, it was shown by Arvind and Srinivasan [2] that if the non-commutative determinant had a poly-sized circuit, then $\mathsf{VP}_{\mathrm{nc}} = \mathsf{VNP}_{\mathrm{nc}}$.

Even though a super-polynomial lower bound is not known for the non-commutative determinant against circuits, Nisan [20] gave an exponential lower bound on the number of gates in any formula computing it. In contrast, the best lower bound known against formulas in the commutative setting is quadratic[2] [19, 14, 6].

A point to note about the lower bound given by Nisan however, is that the proof actually works for a computational model, called Algebraic Branching Programs (or ABPs), that is believed to be more general than algebraic formulas. In fact, Nisan [20] gave an exact characterisation for the size of any ABP computing a non-commutative polynomial. As far as we are aware, any lower bound known against general non-commutative formulas uses this characterisation and hence is essentially a lower bound against non-commutative ABPs itself.

The motivating question for this work is whether there is a separation between the powers of ABPs and formulas in the non-commutative setting. Let us denote the class of non-commutative polynomials over n variables that can be computed by $\mathsf{poly}(n)$-sized ABPs by $\mathsf{VBP}_{\mathrm{nc}}$. Similarly, let $\mathsf{VF}_{\mathrm{nc}}$ denote the class of non-commutative polynomials over n variables that can be computed by $\mathsf{poly}(n)$-sized formulas. The question is essentially whether $\mathsf{VBP}_{\mathrm{nc}}$ is contained in $\mathsf{VF}_{\mathrm{nc}}$ or not.

This question had been posed by Nisan [20], and the only work we are aware of that has made some progress with respect to this question is the one by Lagarde, Limaye and Srinivasan [15]. They show that certain syntactically restricted non-commutative formulas (called Unique Parse Tree formulas) cannot compute $\mathrm{IMM}_{n,n}$ unless they have size $n^{\Omega(\log n)}$.

In this paper, we study restrictions of a different kind. From here on, we will only be talking about non-commutative computation unless specifically mentioned otherwise.

[1] Hyafil [13] had considered non-commutative computation before this, but the main result in that paper is unfortuantely false as shown in [20].

[2] For the elementary symmetric polynomial.

1.1 Abecedarian Polynomials and Models That Compute Them

In [11], Hrubeš et al. have defined the notion of *ordered* polynomials. A homogeneous polynomial of degree d is said to be ordered if the set of variables it depends on can be partitioned into d buckets such that variables occuring in position k only come from the k-th bucket. We generalise this notion by making the bucket indices *position independent*. That is, a variables in position k need not necessarily come from the k-th bucket as long as the variables appear in non-decreasing order of their bucket indices. We call such polynomials abecedarian since, in English, an abecedarian word is one in which all of the letters are arranged in alphabetical order [18].

The difference between ordered polynomials and abecedarian ones can be explained succintly using the notion of *regular expressions* from Automata Theory. For a non-commutative polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n\rangle$, suppose the variables can been partitioned into buckets $\{X_1, \ldots, X_m\}$. f is said to be *ordered* with respect to $\{X_1, \ldots, X_m\}$ if every monomial in it is a word that can be generated using the *regular expression* $X_1 \cdots X_m$. Note that this is equivalent to set-multilinear polynomials in the commutative setting. On the other hand, f is abecedarian if the monomials in it are words that can be generated using the regular expression $X_1^* \cdots X_m^*$. Subsection 2.1 has a formal definition.

"Getting our Hands Dirty" with Abecedarian Polynomials

Before moving ahead, let us take a look at an example of an abecedarian polynomial. Given a commutative polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, define its non-commutative analogue, $f^{(\text{nc})}$ as follows.

> f and $f^{(\text{nc})}$ look essentially the same, except that variables in every monomial in $f^{(\text{nc})}$ are arranged in non-decreasing order of their indices.

Then, $f^{(\text{nc})}$ is abecedarian with respect to the partition $\{X_i \ : \ X_i = \{x_i\}\}$.

Let us also look at a possibly important polynomial that is *not* abecedarian with respect to the partition $\{X_i \ : \ X_i = \{x_i\}\}$. Consider the *arc-full rank polynomial*, f, which was constructed by Dvir, Malod, Perifel and Yehudayoff [7] to give a super-polynomial separation between the powers of formulas and ABPs in the multilinear setting.

We look at f as a non-commutative polynomial, f', in the following sense.

> Let $\mathcal{A}$ be the ABP that computes f and think of $\mathcal{A}$ as a non-commutative ABP $\mathcal{A}'$. Then, f' is the polynomial computed by $\mathcal{A}'$.

It is not hard to see that across different monomials in f', the order in which variables are arranged is not consistent. Thus, f' is not abecedarian with respect to the given partition.

A final point to note before we move ahead is that a polynomial might be abecedarian with respect to different partitions[3]. In fact, even the sizes of the different partitions might be different. For example, the polynomial

$$\text{ESYM}_{n,d}^{(\mathsf{ord})} = \sum_{1 \le i_1 < \ldots < i_d \le n} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}$$

is abecedarian with respect to the partition $\left\{X_k = \left\{x_i^{(k)} \ : \ i \in [n]\right\}\right\}$ which has size d, as well as $\left\{X_i = \left\{x_i^{(k)} \ : \ k \in [d]\right\}\right\}$ which has size n.

[3] Every polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n\rangle$ is abecedarian with respect to the partition $\{X\}$ for $X = \{x_1, \ldots, x_n\}$.

Abecedarian Models of Computation

Hrubeš et al. [11] have defined *ordered circuits*, a model naturally suited to compute ordered polynomials. We generalise this notion to define circuits that naturally compute abecedarian polynomials. We also define abecedarian ABPs and abecedarian formulas similarly.

Suppose f is an abecedarian polynomial with respect to the partition $\{X_1, \ldots, X_m\}$. For any $1 \leq a \leq b \leq m+1$, $f[a,b)$ is a sub-polynomial of f defined as follows.

- For any $1 \leq a \leq [m+1]$, $f[a,a)$ is the constant term in f.
- For $1 \leq a < b \leq m+1$, $f[a,b)$ contains only those monomials of f in which the first variable is from bucket X_a and the last variable is from any of the buckets in the set $\{X_a, \ldots, X_{b-1}\}$.

A circuit is said to be abecedarian if every gate v in it can be labelled by a tuple (a,b) such that if f_v is the polynomial computed at that gate, then $f_v = f_v[a,b)$. We call a formula abecedarian if it has a similar syntactic property at every gate. For formal definitions, see Definition 15 and Definition 17 respectively. On the other hand, an ABP is said to be abecedarian when every vertex in it can be labelled by a bucket index such that if f is the polynomial computed between vertices labelled with indices a and b respectively, then $f = f[a, b+1)$. Definition 16 is a formal definition.

1.2 Our Main Results

Our main result is a super-polynomial separation between abecedarian formulas and ABPs.

▶ **Theorem 1** (Separating Abecedarian Formulas and Abecedarian ABPs). *Define*

$$\text{linked\_CHSYM}_{n,d}(\mathbf{x}) = \sum_{i_0=1}^{n} \left(\sum_{i_0 \leq i_1 \leq \ldots \leq i_d \leq n} x_{i_0,i_1} \cdot x_{i_1,i_2} \cdots x_{i_{d-1},i_d} \right)$$

to be the linked *complete homogeneous polynomial over n-variables of degree d. This polynomial is* abecedarian *with respect to the partition $\{X_i \ : \ i \in [n]\}$ if $X_i = \{x_{i,j} \ : \ i \leq j \leq n\}$.*

With respect to this partition,

1. $\text{linked\_CHSYM}_{n,d}(\mathbf{x})$ *has an* abecedarian *ABP of size $O(nd)$;*
2. *any* abecedarian *formula computing* $\text{linked\_CHSYM}_{n/2,\log n}(\mathbf{x})$ *has size $n^{\Omega(\log \log n)}$.*

That is, there is a super-polynomial separation between abecedarian *formulas and ABPs.*

Our second main result shows that in certain settings, formulas computing abecedarian polynomials can be assumed to be abecedarian without loss of generality.

▶ **Theorem 2** (Converting Formulas into Abecedarian Formulas). *Let f be an* abecedarian *polynomial with respect to a partition of size m, and $\mathcal{F}$ be a formula of size s computing f. If $m = O(\log s)$, then there is an* abecedarian *formula $\mathcal{F}'$ computing f of size* $\mathsf{poly}(s)$.

In other words, an $n^{\omega(1)}$ lower bound against abecedarian formulas computing any polynomial that is abecedarian with respect to a partition of size $O(\log n)$, would result in a super-polynomial lower bound against general non-commutative formulas. These statements suggest a new approach towards resolving the general $\mathsf{VF}_{\mathrm{nc}}$ vs $\mathsf{VBP}_{\mathrm{nc}}$ question.

Connections to the General $\mathsf{VF}_{\mathrm{nc}}$ vs $\mathsf{VBP}_{\mathrm{nc}}$ Question

Theorem 1 gives a separation between abecedarian formulas and ABPs. On the other hand, Theorem 2 shows that if we are given a formula that computes a polynomial that is abecedarian with respect to a partition of *small* size, then we can assume that the formula is abecedarian without loss of generality. Unfortunately, the partition with respect to which our *hard polynomial* from Theorem 1 is abecedarian, is *not small* in size. Thus, the general question of whether $\mathsf{VBP}_{\mathrm{nc}}$ is contained in $\mathsf{VF}_{\mathrm{nc}}$ or not still remains open. However, there are two natural questions that arise at this point.

1. Can any formula computing an abecedarian polynomial be converted to an abecedarian formula without much blow-up in size, irrespective of the size of the partition?
2. Is there a polynomial f which is abecedarian with respect to a partition that has *small* size such that f witnesses a separation between abecedarian formulas and ABPs?

Clearly, a positive answer to either of these questions would imply that $\mathsf{VBP}_{\mathrm{nc}} \neq \mathsf{VF}_{\mathrm{nc}}$. In particular, a super-polynomial lower bound against abecedarian formulas for a polynomial very similar to the one we used to show our separation would separate $\mathsf{VBP}_{\mathrm{nc}}$ and $\mathsf{VF}_{\mathrm{nc}}$.

▶ **Corollary 3.** *Let the polynomial* linked_$\mathrm{CHSYM}_{n,d}(\mathbf{x})$ *be as defined in Theorem 1. An* $n^{\omega(1)}$ *lower bound against* abecedarian *formulas for* linked_$\mathrm{CHSYM}_{\log n,n}(\mathbf{x})$ *would imply a super-polynomial separation between non-commutative ABPs and formulas.*

In fact our proof technique also shows that a super-polynomial lower bound against *homogeneous* formulas for our hard polynomial would separate $\mathsf{VBP}_{\mathrm{nc}}$ and $\mathsf{VF}_{\mathrm{nc}}$.

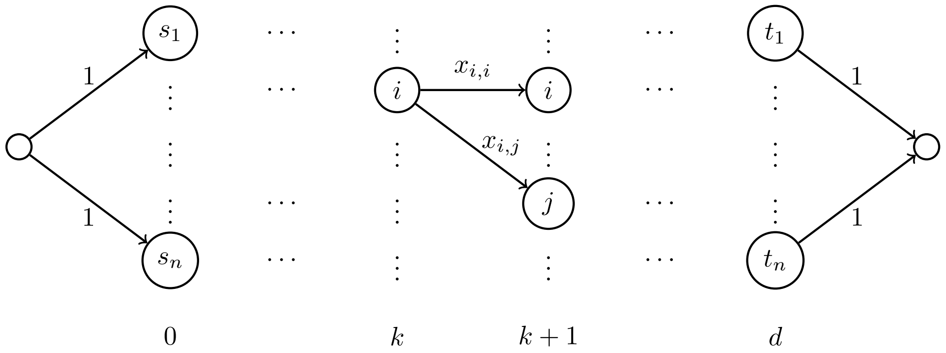
▶ **Corollary 4.** *Let* linked_$\mathrm{CHSYM}_{n,d}(\mathbf{x})$ *be as defined in Theorem 1. An* $n^{\omega(1)}$ *lower bound against homogeneous formulas for* linked_$\mathrm{CHSYM}_{n,\log n}(\mathbf{x})$ *would result in a super-polynomial separation between ABPs and formulas in the non-commutative setting.*

1.3 Proof Overview

We now give a proof overview of our main theorems.

Separating Abecedarian Formulas and ABPs

Let us first consider Theorem 1.
A *small* abecedarian ABP computing linked_$\mathrm{CHSYM}_{n,d}(\mathbf{x})$ is essentially the following.



For the lower bound, assume that we have been given a *small* abecedarian formula computing the polynomial. We then keep modifying this formula till we get a *small* homogeneous multilinear formula computing the *elementary symmetric polynomial* of degree $n/2$. We then use the known lower bound against homogeneous multilinear formulas for this polynomial (shown by Hrubeš and Yehudayoff [12]), to get a contradiction.

Let us spell out the proof in some more detail.

Step 1: Suppose we are given an abecedarian formula computing linked_$\mathrm{CHSYM}_{n/2,\log n}(\mathbf{x})$ of size $O(n^{\epsilon \log\log n})$. Since the degree of the polynomial being computed is *small*, we can assume that there is in fact a *homogeneous* abecedarian formula computing linked_$\mathrm{CHSYM}_{n/2,\log n}(\mathbf{x})$ of size $O(n^{c\cdot\epsilon\log\log n})$ for some constant c independent of ϵ.

Step 2: Using the homogeneous abecedarian formula from Step 1, we obtain a more *structured* homogeneous abecedarian formula, of size $O(n^{c\cdot\epsilon\log\log n})$, that computes the same polynomial.

Step 3: We consider the complete homogeneous polynomial over n variables of degree d

$$\mathrm{CHSYM}_{n,d}(\mathbf{x}) = \sum_{1\le i_1\le\ldots\le i_d\le n} x_{i_1}\cdots x_{i_d},$$

and show that there is a homogeneous abecedarian formula of size $\mathsf{poly}(n)$ that computes $\mathrm{CHSYM}_{n/2,\log n}(\mathbf{x})$.

Step 4: If the formula in Step 2 has size s and that in Step 3 has size s', then we show that there is a homogeneous abecedarian formula of size $(s\cdot s')$ computing $\mathrm{CHSYM}_{n/2,\log^2 n}(\mathbf{x})$.

Step 5: Next, we show that Step 4 can be used repeatedly at most $O({}^{\log n}/_{\log\log n})$ times, to obtain a homogeneous abecedarian formula computing $\mathrm{CHSYM}_{n/2,n/2}(\mathbf{x})$ of size $O(n^{c\cdot\epsilon\log n})$.

Step 6: Using the formula obtained in Step 5, we get a homogeneous multilinear formula computing the elementary symmetric polynomial of degree $n/2$, of size $O(n^{c\cdot\epsilon\log n})$.

Step 7: Finally, we choose ϵ in such a way that Step 6 contradicts the theorem in [12].

The crucial observation that makes this proof work, is that the polynomial we are working with is structured enough for us to be able to amplify its degree in a systematic way (without blowing up the size by much). This is the 4th step in the description above.

Apart from that, the entire proof essentially boils down to the fact that when formulas are computing low degree polynomials, there are some additional tricks available to make them more structured. A complete proof of Theorem 1 can be found in Section 5.

We now elaborate a little on the first step, since the observations made to prove this step are quite general and possibly useful in various settings. These statements are known to be true in the commutative setting and their proofs in the non-commutative setting are fairly similar to the ones for their commutative counterparts. We state them here nevertheless, since to the best of our knowledge, they have not been stated formally before for the non-commutative setting.

Homogenising Abecedarian Formulas computing Low Degree Polynomials

Raz [21] had shown that if there is a formula computing a homogeneous polynomial of *low* degree in the commutative world, then it can be assumed without loss of generality that the formula is homogeneous. We show that this statement is true even in the non-commutative setting.

▶ **Lemma 5** (Homogenising Abecedarian Formulas computing Low Degree Polynomials). *Suppose f is a non-commutative homogeneous polynomial that can be computed by a fan-in 2 formula, $\mathcal{F}$, of size s, and has degree $d = O(\log s)$. Then there is a homogeneous formula $\mathcal{F}'$ computing f, that has size* $\mathsf{poly}(s)$ *and whose multiplication gates have fan-in 2. Further, if $\mathcal{F}$ was* abecedarian *with respect to some partition, then $\mathcal{F}'$ is also* abecedarian *with respect to the same partition.*

The only thing that needs to be checked for Raz's proof to work in this setting is whether non-commutative formulas, and in particular abecedarian formulas can be depth-reduced to log-depth. We show that infact they can be.

Depth Reduction for Abecedarian Formulas

Brent [4] had shown that if there is a formula of size s computing a commutative polynomial f, then there is a formula of depth $O(\log s)$ and size $\mathsf{poly}(s)$ that computes the same polynomial. A similar statemnt was shown by Hrubeš and Wigderson [9] in the non-commutative setting[4]. We show that the statement continues to be true for abecedarian formulas. The proof is exactly along the same lines as the one by Brent [4].

▶ **Lemma 6** (Depth Reduction of Abecedarian Formulas). *If there is a fan-in 2 formula $\mathcal{F}$ of size s computing a non-commutative polynomial f, then there is a fan-in 2 formula $\mathcal{F}'$ of size* $\mathsf{poly}(s)$ *and depth $O(\log(s))$ computing f. Further if $\mathcal{F}$ is homogeneous, $\mathcal{F}'$ is also homogeneous. Similarly, if $\mathcal{F}$ is* abecedarian *with respect to some partition, then $\mathcal{F}'$ is also* abecedarian *with respect to the same partition.*

Converting Formulas into Abecedarian Formulas

Next we go over the proof idea of Theorem 2. In order to prove the statement, we first convert the given formula $\mathcal{F}$ into an abecedarian circuit $\mathcal{C}$, and then unravel $\mathcal{C}$ in order to get an abecedarian formula $\mathcal{F}'$ computing the same polynomial.

The first step is fairly straightforward. The proof is along the same lines as that for homogenising circuits, the only difference being that we keep track of bucket indices of the variables on either ends of the monomials being computed, instead of their degrees.

In the second step, we convert $\mathcal{C}$ into a formula $\mathcal{F}'$. In order to do that, we need to recompute vertices every time it is reused. Thus, to give an upper bound on the size of $\mathcal{F}'$, we need to find an upper bound on the number of distinct paths from any vertex in $\mathcal{C}$ to the root. This analysis is done similarly to the one by Raz [21] in his proof of the fact that formulas computing low degree polynomials can be homogenised efficiently. The requirement of the size of the partition being small also arises because of this analysis.

The only additional point that needs to be checked for the proof to go through is that similar to the commutative setting, non-commutative formulas can be depth reduced as well (Lemma 6). A complete proof of Theorem 2 can be found in Subsection 4.3.

1.4 Other Results: A Complete View of the Abecedarian World

We now go over some other results that helps in completing the view of the abecedarian world. As mentioned earlier, Hrubeš et al. [11] had defined ordered circuits, a model naturally suited to compute ordered polynomials. They had then gone on to show that without loss of generality, any circuit computing an ordered polynomial can be assumed to be ordered[5]. We show that even in the abecedarian setting, such a statement is true.

▶ **Observation 7** (Converting Circuits into Abecedarian Circuits). *Let f be an* abecedarian *polynomial with respect to a Partition of size m, and $\mathcal{C}$ be a circuit of size s computing f. Then there is an* abecedarian *circuit $\mathcal{C}'$ computing f of size $O(m^3 s)$.*

What this implies is that an $n^{\omega(1)}$ lower bound against abecedarian circuits for any explicit polynomial that is abecedarian would result in a super-polynomial lower bound against general non-commutative circuits. We also show that an analogous statement is true even for abecedarian ABPs.

[4] They in fact showed it for rational functions

[5] Theorem 7.1 in [11].

▶ **Observation 8** (Converting ABPs into Abecedarian ABPs). *Suppose f is an* abecedarian *polynomial with respect to a partition of size m. If there is an ABP $\mathcal{A}$ of size s computing it, then there is an* abecedarian *ABP $\mathcal{A}'$ computing it of size $O(ms)$.*

Next, we define the natural classes of abecedarian polynomials. Let abc - $\mathsf{VP}_{\mathrm{nc}}$ denote the class of abecedarian polynomials that can be computed by poly-sized abecedarian circuits. Similarly let abc - $\mathsf{VBP}_{\mathrm{nc}}$ and abc - $\mathsf{VF}_{\mathrm{nc}}$ denote the classes of abecedarian polynomials that can be computed by poly-sized abecedarian ABPs and abecedarian formulas respectively. We first note that the logical inclusions that should hold, do hold.

▶ **Observation 9** (The Usual Inclusions). *Let* abc - VP_{nc}*,* abc - VBP_{nc} *and* abc - VF_{nc} *denote the classes of* abecedarian *polynomials over n variables that can be computed by* $\mathsf{poly}(n)$ *sized* abecedarian *circuits,* abecedarian *ABPs and* abecedarian *formulas respectively. Then,*

$$\mathsf{abc}\text{ - }\mathsf{VF}_{nc} \subseteq \mathsf{abc}\text{ - }\mathsf{VBP}_{nc} \subseteq \mathsf{abc}\text{ - }\mathsf{VP}_{nc}.$$

We also observe that if a degree d polynomial has an abecedarian ABP of size s, then it has an abecedarian formula of size $O(s^{\log d})$ via the usual divide-and-conquer algorithm.

▶ **Observation 10** (Converting Abecedarian ABPs into Abecedarian Formulas). *Suppose f is an* abecedarian *polynomial of degree d. If there is an* abecedarian *ABP $\mathcal{A}$ of size s computing it, then there is an* abecedarian *formula $\mathcal{F}$ computing f of size $O(s^{\log d})$.*

What Theorem 1 essentially shows is that the blow-up observed in Observation 10 is tight.

Finally, it is not hard to see that Nisan's proof can be modified to give an exponential separation between abecedarian ABPs and abecedarian circuits.

General Formula Lower Bound from Homogeneous Formula Lower Bound

We end by showing that homogeneous formula lower bounds for the iterated matrix multiplication polynomial would lead to separating $\mathsf{VF}_{\mathrm{nc}}$ and $\mathsf{VBP}_{\mathrm{nc}}$. This is a corollary of Lemma 5.

▶ **Corollary 11.** *An $n^{\omega(1)}$ lower bound against homogeneous formulas computing the n-variate iterated matrix multiplication polynomial of degree $\log n$, $\mathrm{IMM}_{n,\log n}(\mathbf{x})$, implies a super-polynomial separation between ABPs and formulas in the non-commutative setting.*

To put the requirement of degree being $O(\log n)$ in perspective, note the following.

▶ Remark 12 (Analogous to Remark 5.12 in [15]). The standard divide and conquer approach for computing the iterated matrix multiplication polynomial $\mathrm{IMM}_{n,d}$ yields a (homogeneous) formula of size $n^{O(\log d)}$. It would be quite surprising if this standard algorithm were not optimal in terms of formula size.

Intuitively, improving on the standard divide and conquer algorithm gets harder as d gets smaller. This is because any (homogeneous) formula of size $n^{o(\log d)}$ for computing $\mathrm{IMM}_{n,d}$ can be used in a straightforward manner to recursively obtain (homogeneous) formulas for $\mathrm{IMM}_{,n,D}$ of size $n^{o(logD)}$ for any $D > d$. The case of smaller d, which seems harder algorithmically, is thus a natural first candidate for lower bounds.

1.5 Structure of the Paper

We begin, in Section 2, with formal definitions for abecedarian polynomials and naturally restricted version of circuits, ABPs and formulas that compute them. Then, in Section 3, we prove some structural statements, namely Lemma 6 and Lemma 5. In Section 4, we prove

Theorem 2 along with Observation 7 and Observation 8. We then prove our main result (Theorem 1), that gives a super-polynomial separation between abecedarian formulas and ABPs, in Section 5. Finally, in Section 6, we prove the remaining statements mentioned.

2 Preliminaries

Let us begin by formally defining abecedarian polynomials and the naturally restricted versions of circuits, ABPs and formulas that compute them. Throughout the write-up, we will use $[n]$ to denote the set $\{1, \ldots, n\}$.

2.1 Abecedarian Polynomials

First, we formally define abecedarian polynomials.

▶ **Definition 13** (Abecedarian Polynomials). *A polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ of degree d is said to be* abecedarian *with respect to a partition $\{X_1, \ldots, X_m\}$ for $\{x_1, \ldots x_n\}$, if*

$$f = f[\emptyset] + \sum_{k=1}^{d} \left(\sum_{1 \le i_1 \le \cdots \le i_k \le m} f[X_{i_1}, \ldots, X_{i_k}] \right)$$

where $f[\emptyset]$ is the constant term in f, and for any $k \in [d]$, $f[X_{i_1}, \ldots, X_{i_k}]$ is defined as follows. For a polynomial f, $f[X_{i_1}, \ldots, X_{i_k}]$ is the homogeneous polynomial of degree k such that for every monomial α,

$$\mathsf{coeff}_\alpha(f[X_{i_1}, \ldots, X_{i_k}]) = \begin{cases} \mathsf{coeff}_\alpha(f) & \text{if } \alpha = x_{\ell_1} \cdots x_{\ell_k} \text{ with } x_{\ell_j} \in X_{i_j} \text{ for every } j \in [k] \\ 0 & \text{otherwise.} \end{cases}$$

In this case, we say that f is abecedarian *with respect to $\{X_1, \ldots, X_m\}$, a partition of size m.*

Abecedarian polynomials are essentially generalisations of ordered polynomials (defined by Hrubeš, Wigderson and Yehudayoff [11]). A homogeneous polynomial, of degree d, is said to be ordered if the set of variables it depends on can be partitioned into d buckets such that variables occuring in position k only come from the k-th bucket.

It is easy to see that any ordered polynomial is also abecedarian with respect to the same partition. This is because position indices are always increasing. For example, consider the following version of the *complete homogeneous symmetric polynomial.*

$$\mathrm{CHSYM}_{n,d}^{(\mathsf{ord})}(\mathbf{x}) = \sum_{1 \le i_1 \le \ldots \le i_d \le n} x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}.$$

It is both ordered and abecedarian with respect to the partition $\left\{X_k = \left\{x_i^{(k)} : i \in [n]\right\}\right\}$.

However, note that there are homogeneous polynomials that are abecedarian but not ordered. The following version of the same polynomial is an example.

$$\mathrm{CHSYM}_{n,d}(\mathbf{x}) = \sum_{1 \le i_1 \le \ldots \le i_d \le n} x_{i_1} \cdots x_{i_d}$$

is abecedarian with respect to $\{X_i : X_i = \{x_i\}\}$, but is not ordered.

The reason is that for a polynomial to be ordered, the bucket labels have to essentially be position labels. On the other hand, for a polynomial to be abecedarian with respect to a partition, the bucket labels can be independent of position. For example, note that $\mathrm{CHSYM}_{n,d}^{(\mathsf{ord})}(\mathbf{x})$ is abecedarian with respect to the partition $\left\{X_i = \left\{x_i^{(k)} : k \in [d]\right\}\right\}$ along with the one mentioned earlier.

We now move on to defining algebraic models that naturally compute such polynomials.

2.2 Abecedarian Models of Computation

Homogeneous formulas have the property that any vertex can be labelled by a tuple of position indices (a, b) such that all the monomials being computed at that vertex occur exactly from position a to position b in the final polynomial that is being computed by it. Hrubeš et al. [11] defined *ordered* circuits to be those circuits that have this property.

A circuit computing a degree d polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n\rangle$ is said to be ordered, if $\{X_1, \ldots, X_d\}$ forms a partition of $\{x_1, \ldots, x_n\}$ such that

- every gate v in the circuit is labelled by a tuple of position indices (a, b);
- if f_v is the polynomial computed at v, then
  - f_v is homogeneous and has degree $(b - a + 1)$;
  - every monomial in f_v is a product of exactly one variable from each of the buckets $X_a, \ldots, X_b$, multiplied in increasing order of their bucket indices.

We generalise this notion to define circuits that naturally compute abecedarian polynomials. Before we can do that, we need the notion of sub-polynomials of any abecedarian polynomial.

▶ **Definition 14** (Sub-Polynomials of an Abecedarian Polynomial). *Suppose f is an* abecedarian *polynomial with respect to the partition $\{X_1, \ldots, X_m\}$, and has degree d. For any $1 \le a \le b \le m+1$, $f[a, b)$ is the sub-polynomial of f defined as follows.*

- *For any $a \in [m+1]$, $f[a, a) = f[\emptyset)$ is the constant term in f.*
- *For any $1 \le a < b \le m+1$,*

$$f[a, b) = \sum_{k=1}^{d} \left(\sum_{\substack{i_1, \ldots, i_k \in [m] \\ a = i_1 \le \cdots \le i_k < b}} f[X_{i_1}, \ldots, X_{i_k}] \right)$$

  where $f[X_{i_1}, \ldots, X_{i_k}]$ is as defined in Definition 13.

Further, we say that a polynomial f is of type *$[a, b)$ if $f = f[a, b)$.*

Let us now formally define abecedarian circuits.

▶ **Definition 15** (Abecedarian Circuits). *For any $a, b \in \mathbb{N}$, let $[a, b)$ denote a set of the form $I = \{i \ : \ a \le i < b\}$. As a convention, $[a, a)$ denotes the empty set for every $a \in \mathbb{N}$.*

A multi-output circuit $\mathcal{C}$ is said to be abecedarian *when*

- *every gate v in $\mathcal{C}$ is associated with a set $I_v = [a, b)$;*
- *if f_v is the polynomial computed at v, then $f_v = f[a, b)$;*
- *if $v = v_1 + v_2$, then $I_v = I_{v_1} = I_{v_2}$;*
- *if $v = v_1 \times v_2$ with $I_v = [a, a)$, then $I_{v_1} = I_{v_2} = [a, a)$*
- *if $v = v_1 \times v_2$ with $I_v = [a, b)$ and $a < b$, then one of the following is true*
  - *$I_{v_1} = [a, b)$ and $I_{v_2} = [b, b)$;*
  - *$I_{v_1} = [a, a)$ and $I_{v_2} = [a, b)$;*
  - *there exists $a \le c < b$ such that $I_{v_1} = [a, c+1)$ and $I_{v_2} = [c, b)$.*

The polynomial computed by $\mathcal{C}$ is the sum of the polynomials computed at the output gates.

Next, we define abecedarian ABPs and abecedarian formulas as the restricted versions of ABPs and formulas respectively, that naturally compute abecedarian polynomials.

2.3 Abecedarian ABPs and Formulas

Homogeneous ABPs have the property that every vertex in it is labelled by a position index such that, polynomials computed between vertices labelled with indices a and b only contain monomials between positions a and $(b-1)$. We define abecedarian ABPs analogously except that the labels on the vertices are bucket labels instead of position labels. These restricted ABPs naturally compute abecedarian polynomials.

▶ **Definition 16** (Abecedarian ABPs). *A multi-input, multi-output ABP $\mathcal{A}$ is said to be* abecedarian *when*

- *every vertex in it is labelled by a bucket index;*
- *if f is the polynomial computed between vertices labelled with indices a and b respectively, then $f = f[a, b+1)$.*

The polynomial computed by $\mathcal{A}$ is the sum of all the polynomials computed between the various (input, output) gate pairs.

Similarly, we define abecedarian formulas as analogues of homogeneous formulas, with the labels again referring to bucket indices instead of position indices.

▶ **Definition 17** (Abecedarian Formulas). *Let sets of the form $[a, b)$, with $a, b \in \mathbb{N}$, be as defined in Definition 15. Suppose $\mathcal{F}$ is a formula computing a polynomial f that is* abecedarian *with respect to a partition of size m. Then $\mathcal{F}$ is said to be* abecedarian *if $\mathcal{F} = \mathcal{F}_1 + \cdots + \mathcal{F}_m$ for sub-formulas $\mathcal{F}_1, \ldots, \mathcal{F}_{m+1}$, where for every $i \in [m+1]$:*

- *$\mathcal{F}_i$ computes the polynomial $f[i, m+1)$;*
- *every gate v in $\mathcal{F}_i$ is associated with a set $I_v = [a, b)$, and in particular, the root node must be associated with the set $[i, m+1)$*
- *if f_v is the polynomial computed at v, then $f_v = f_v[a, b)$;*
- *if $v = v_1 + v_2$, then $I_v = I_{v_1} = I_{v_2}$;*
- *if $v = v_1 \times v_2$ with $I_v = [a, a)$, then $I_{v_1} = I_{v_2} = [a, a)$*
- *if $v = v_1 \times v_2$ with $I_v = [a, b)$ and $a < b$, then one of the following is true*
  - *$I_{v_1} = [a, b)$ and $I_{v_2} = [b, b)$;*
  - *$I_{v_1} = [a, a)$ and $I_{v_2} = [a, b)$;*
  - *there exists $a \leq c < b$ such that $I_{v_1} = [a, c+1)$ and $I_{v_2} = [c, b)$.*

Further, $\mathcal{F}$ is said to be homogeneous if each $\mathcal{F}_i$ is homogeneous.

With these definitions in mind, we now move to proving some structural statements.

3 Structural Statements

In this section, we prove two structural statements in the non-commutative setting that are known to be true in the commutative setting. Apart from being crucial to our proofs, they are possibly interesting observations in their own right.

3.1 Depth Reduction for Non-Commutative Formulas

Brent [4] had shown that if there is a formula of size s computing a commutative polynomial f, then there is a formula of depth $O(\log s)$ and size $\mathsf{poly}(s)$ that computes the same polynomial. We show that this is also true in the non-commutative setting.

The proof is essentially the same as the one by Brent [4], just analysed carefully. We give the complete proof for the sake of completeness.

▶ **Lemma 6** (Depth Reduction of Abecedarian Formulas). *If there is a fan-in* 2 *formula* $\mathcal{F}$ *of size* s *computing a non-commutative polynomial* f*, then there is a fan-in* 2 *formula* $\mathcal{F}'$ *of size* $\mathsf{poly}(s)$ *and depth* $O(\log(s))$ *computing* f*. Further if* $\mathcal{F}$ *is homogeneous,* $\mathcal{F}'$ *is also homogeneous. Similarly, if* $\mathcal{F}$ *is* abecedarian *with respect to some partition, then* $\mathcal{F}'$ *is also* abecedarian *with respect to the same partition.*

Proof. Suppose $\mathcal{F}$ is a fan-in 2 formula of size s that computes f. Then, we claim the following.

▷ Claim 18. Suppose $\mathcal{F}_0$ is a formula computing a polynomial f_0 and has fan-in 2. Then the there exist sub-formulas, $L, \mathcal{F}_1, R, \mathcal{F}_2$, of $\mathcal{F}_0$ such that

- $\mathcal{F}_0' = L \cdot \mathcal{F}_1 \cdot R + \mathcal{F}_2$ also computes f_0;
- each of $L, \mathcal{F}_1, R, \mathcal{F}_2$ have size at least $(s/3)$ and at most $(2s/3)$;
- if $\mathcal{F}_0$ is homogeneous, then so are $L, \mathcal{F}_1, R, \mathcal{F}_2$;
- if $\mathcal{F}_0$ is abecedarian with respect to some partition, f_{left}, f_1, f_{right}, f_2 are polynomials computed by $L, \mathcal{F}_1, R, \mathcal{F}_2$ respectively and $f_0 = f_0[a, b)$, then $f_2 = f_2[a, b)$ and
  - each of $L, \mathcal{F}_1, R, \mathcal{F}_2$ are abecedarian with respect to the same partition as $\mathcal{F}_0$
  - when $a = b$, $\qquad f_{\mathsf{left}} = f_{\mathsf{left}}[a, a) \qquad f_1 = f_1[a, a) \qquad f_{\mathsf{right}} = f_{\mathsf{right}}[a, a)$;
  - when $a < b$, there exist $a \le i \le j \le b$ such that

$$\begin{aligned}
a = i < j = b &\implies & f_{\mathsf{left}} &= f_{\mathsf{left}}[a, i) & f_1 &= f_1[i, j) & f_{\mathsf{right}} &= f_{\mathsf{right}}[j, b).\\
a = i = j < b &\implies & f_{\mathsf{left}} &= f_{\mathsf{left}}[a, i) & f_1 &= f_1[i, j) & f_{\mathsf{right}} &= f_{\mathsf{right}}[j, b).\\
a = i < j < b &\implies & f_{\mathsf{left}} &= f_{\mathsf{left}}[a, i) & f_1 &= f_1[i, j+1) & f_{\mathsf{right}} &= f_{\mathsf{right}}[j, b).\\
a < i = j = b &\implies & f_{\mathsf{left}} &= f_{\mathsf{left}}[a, i+1) & f_1 &= f_1[i, j) & f_{\mathsf{right}} &= f_{\mathsf{right}}[j, b).\\
a < i = j < b &\implies & f_{\mathsf{left}} &= f_{\mathsf{left}}[a, i+1) & f_1 &= f_1[i+1, j+1) & f_{\mathsf{right}} &= f_{\mathsf{right}}[j, b).\\
a < i < j = b &\implies & f_{\mathsf{left}} &= f_{\mathsf{left}}[a, i+1) & f_1 &= f_1[i, j) & f_{\mathsf{right}} &= f_{\mathsf{right}}[j, b).\\
a < i < j < b &\implies & f_{\mathsf{left}} &= f_{\mathsf{left}}[a, i+1) & f_1 &= f_1[i, j+1) & f_{\mathsf{right}} &= f_{\mathsf{right}}[j, b).
\end{aligned}$$

Before proving Claim 18, let us complete the proof of Lemma 6 using it.

By the above claim, we have a formula $\mathcal{F}_0'$ computing f_0 that looks like $L \cdot \mathcal{F}_1 \cdot R + \mathcal{F}_2$ where each of $L, \mathcal{F}_1, R, \mathcal{F}_2$ have size at most $(2s/3)$. Further if $\mathcal{F}$ is homogeneous, then so are each of $L, \mathcal{F}_1, R, \mathcal{F}_2$. Hence, $\mathcal{F}_0'$ is homogeneous. On the other hand, when $\mathcal{F}_0$ is abecedarian, so are $L, \mathcal{F}_1, R, \mathcal{F}_2$. Further, note that $\mathcal{F}_0'$ is also abecedarian in this case since $f_{\mathsf{left}}, f_1, f_{\mathsf{right}}, f_2$ are of the *correct type* due to Claim 18.

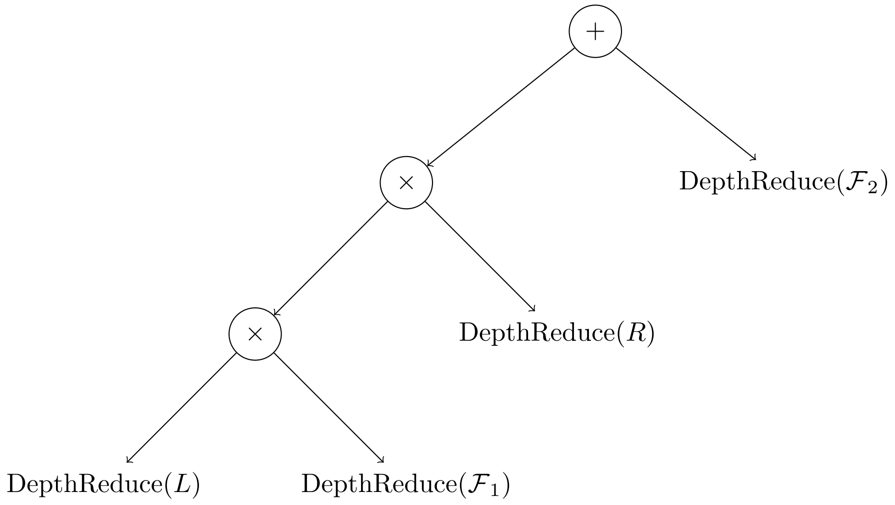
In all the cases, recursively applying this technique, on each of $L, \mathcal{F}_1, R, \mathcal{F}_2$, we get

$$\mathsf{depth}(s) \le \mathsf{depth}(2s/3) + 3 \qquad \text{and} \qquad \mathsf{size}(s) \le 4 \cdot \mathsf{size}(2s/3) + 3.$$

Note that in the base case, when s is constant, both $\mathsf{size}(s)$ and $\mathsf{depth}(s)$ are constants. Thus,

$$\mathsf{depth}(s) = O(\log s) \qquad \text{and} \qquad \mathsf{size}(s) = \mathsf{poly}(s).$$ ◀

Pictorially, once we have Claim 18, we essentially do the following recursively.



We now complete the proof of Claim 18.

Proof of Claim 18. From the root let us traverse $\mathcal{F}_0$ towards the leaves, always choosing the child that has a larger sub-tree under it, till we find a vertex v such that the associated sub-tree has size at most $(2s/3)$. Since $\mathcal{F}_0$ tree has fan-in 2, we also know that the size of this sub-tree must be at least $(s/3)$. Let this sub-tree be $\mathcal{F}_1$. Additionally, in the case when $\mathcal{F}_0$ is abecedarian, let us assume that v is labelled with $[i_v, j_v)$.

Let $\mathcal{P}$ be the path from v to the root and v_{add} the addition gate on $\mathcal{P}$ which is closest to v. Also let the set of multiplication gates on $\mathcal{P}$ be $\{v_1, \ldots, v_\ell\}$ for some $\ell \in \mathbb{N}$. Assume, without loss of generality, that v_1 is closest to v and v_ℓ to the root. Further, for every $i \in [\ell]$, let L_i be sub-formula corresponding to the left child of v_i and R_i the one to its right child. Note that for every $i \in [\ell]$, exactly one of children of v_i is a vertex in $\mathcal{P}$. We can then define L and R as follows.

Step 1: Set $L = R = 1$.

Step 2: For i from 1 to ℓ,

$$L = \begin{cases} L_i \times L & \text{if the right child of } v_i \text{ is a vertex in } \mathcal{P}, \\ L & \text{otherwise.} \end{cases}$$

and

$$R = \begin{cases} R & \text{if the right child of } v_i \text{ is a vertex in } \mathcal{P}, \\ R \times R_i & \text{otherwise.} \end{cases}$$

Also define $\mathcal{F}_2$ to be the formula we get by replacing the vertex v and the sub-tree under it with 0, and then removing the redundant gates.

Clearly, by construction, $\mathcal{F}_1$, L, R and $\mathcal{F}_2$ are sub-formulas of $\mathcal{F}_0$. Further, $\mathcal{F}_1$ is disjoint from L, R and $\mathcal{F}_2$. As a result, since $\mathcal{F}_1$ has size at least $(s/3)$ and at most $(2s/3)$, it must be the case that each of L, R and $\mathcal{F}_2$ have size at least $(s/3)$ and at most $(2s/3)$.

Also, it is not hard to see that $\mathcal{F}_0' = L \cdot \mathcal{F}_1 \cdot R + \mathcal{F}_2$ computes f_0. What is left to check is that when $\mathcal{F}_0$ is homogeneous or abecedarian, then $L, \mathcal{F}_1, R, \mathcal{F}_2$ have the additional properties claimed. The one line proof of this is that each *parse-tree*[6] of $\mathcal{F}_0$ is merely restructured in the above process, without changing its value. We however go over the proof explicitly for the sake of completeness.

[6] For a definition, see for example [15].

When $\mathcal{F}_0$ is homogeneous, since $L, \mathcal{F}_1, R, \mathcal{F}_2$ are sub-formulas, they are also homogeneous. On the other hand, suppose $\mathcal{F}_0$ is abecedarian and $f_0 = f_0[a, b)$. Recall that the vertex v was labelled by $[i_v, j_v)$. Let us set $i = i_v$ and $j = j_v$. Then, by definition, $\mathcal{F}_1$ is labelled by $[i, j)$. Hence, if f_1 is the polynomial computed at v, then $f_1 = f_1[i, j)$. Further, $\mathcal{F}_1$ is abecedarian since it is a sub-formula of $\mathcal{F}_0$ and computes an abecedarian polynomial.

Now let us focus on $\mathcal{F}_2$. Essentially $\mathcal{F}_2$ is got by removing from $\mathcal{F}_0$, v and all the multiplication gates on P between v and v_{add} along with the sub-trees under them. Thus $\mathcal{F}_2$ is also abecedarian in this case, and if f_2 is the polynomial by it, then $f_2 = f_2[a, b)$.

Finally, note that the left indices of labels on the various vertices of $\mathcal{P}$ change only at the gates at which multiplications to L occur. Further, note that they occur in the *correct order* and are of the *correct type*. Thus, by induction, it is easy to see that the labels on L are consistent with those on the L_is when the respective multiplications happen. Therefore L is abecedarian, and $f_{\mathsf{left}} = f_{\mathsf{left}}[a, i)$.

For similar reasons, R is also abecedarian and $f_{\mathsf{right}} = f_{\mathsf{right}}[j, b)$. This completes the proof. ◁

3.2 Homogenisation

Raz [21] had shown that if there is a formula computing a homogeneous polynomial of *low* degree in the commutative world, then it can be assumed without loss of generality that the formula is homogeneous. We show that his proof also works in the non-commutative setting because of Lemma 6. A complete proof is given here for the sake of completeness.

▶ **Lemma 5** (Homogenising Abecedarian Formulas computing Low Degree Polynomials). *Suppose f is a non-commutative homogeneous polynomial that can be computed by a fan-in 2 formula, $\mathcal{F}$, of size s, and has degree $d = O(\log s)$. Then there is a homogeneous formula $\mathcal{F}'$ computing f, that has size* $\mathsf{poly}(s)$ *and whose multiplication gates have fan-in 2. Further, if $\mathcal{F}$ was* abecedarian *with respect to some partition, then $\mathcal{F}'$ is also* abecedarian *with respect to the same partition.*

Proof. We first note that since s is the ABP complexity of f, $s' \geq s$. Further if $\mathcal{F}$ has depth r, then by Lemma 6, we can assume without loss of generality, that $r = O(\log s')$.

In order to construct a homogeneous formula computing f, we first homogenise $\mathcal{F}$ to obtain a circuit $\mathcal{C}$, and then *unravel* $\mathcal{C}$ to make it into a formula $\mathcal{F}'$.

The first step is done in the usual manner. For every gate v in $\mathcal{F}$, we have $d+1$ gates $(v, 0)$, …, (v, d) in $\mathcal{C}$. Intuitively if f_v is the polynomial computed at v, then the polynomial computed at (v, i) is the degree i homogeneous component of f_v. These vertices are then connected as follows.

- If $v = u_1 + u_2$, then for every $i \in \{0, \ldots, d\}$, $(v, i) = (u_1, i) + (u_2, i)$.
- If $v = u_1 \times u_2$, then for every $i \in \{0, \ldots, d\}$, $(v, i) = \sum_{j=0}^{i} (u_1, j) \times (u_2, i - j)$.

So, we now have a homogeneous circuit $\mathcal{C}$ that computes f and has size at most $O(d^2 \cdot s')$. Also, the depth of this circuit is at most twice that of $\mathcal{F}$, and the multiplication gates have fan-in 2. To convert $\mathcal{C}$ into a formula $\mathcal{F}'$, we have to recompute nodes whenever they have to be reused. That is, a particular vertex in $\mathcal{C}$ has to be duplicated as many times as there are paths from the vertex to the root. Thus, to upper bound the size of $\mathcal{F}'$, we need to give an upper bound on the number of distinct paths from every vertex of $\mathcal{C}$ to its root.

Let us arbitrarily choose a vertex (v, i) in $\mathcal{C}$, and consider the path from it to the root. Suppose the path is $(v, i) = (v_1, i_1) \to \cdots \to (v_\ell, i_\ell) = (\mathsf{root}, d)$ where ℓ is at most the depth of $\mathcal{C}$. Now since $\mathcal{C}$ comes from a formula, the only reason multiple paths can exist is because of the second index, and therefore it is enough to focus on that. Note that it must be the case that $i = i_1 \leq \cdots \leq i_\ell = d$. Hence, if we define $\delta_j = i_{j+1} - i_j$ for $j \in [\ell - 1]$, then the δ_js are non-negative integers such that $\delta_1 + \cdots + \delta_{\ell-1} = (d - i)$. Thus, the number of choices we have for $(i_2, \ldots, i_\ell)$ such that $i = i_1 \leq \cdots \leq i_\ell = d$, is the same as the number of choices we have for $(\delta_1, \ldots, \delta_{\ell-1})$ such that $\delta_1 + \cdots + \delta_{\ell-1} = (d - i) \leq d$. This is at most $\binom{\ell+d}{\ell}$. Note that in this process the fan-in of the gates have not changed, and hence the multiplication gates in $\mathcal{F}'$ continue to have fan-in 2. Further, we know that the $\mathcal{C}$ has depth $2r$ and hence $\ell \leq 2r$. Therefore, the number of paths from (v, i) to the root is at most $\binom{2r+d}{2r}$. Hence, if $\mathcal{F}'$ is the formula obtained by unravelling $\mathcal{C}$, then $\mathsf{size}(\mathcal{F}') \leq s' \cdot d^2 \cdot \binom{2r+d}{d}$. Here $r = O(\log(s'))$, and $s \leq s'$ implying that $d = O(\log(s)) = O(\log(s'))$. Thus, $\mathsf{size}(\mathcal{F}') \leq \mathsf{poly}(s')$.

Finally, assume that $\mathcal{F}$ is abecedarian. Then every vertex v is labelled with a tuple of bucket indices, say (a_v, b_v). In that case, we add the label (a_v, b_v) to the gates $\{(v, i)\}_{i=0}^{d}$ in $\mathcal{C}$ and continue with the proof as is. Note that the final formula that we get, $\mathcal{F}'$, is abecedarian and all the other properties that were true in the general case, continue to be true. ◀

4 Converting Computational Models into Abecedarian Ones

In this section we show that, without loss of generality, circuits and ABPs computing abecedarian polynomials can be assumed to be abecedarian. For formulas however, we can prove such a statement only in certain cases.

4.1 Circuits

Hrubeš et al. [11] had shown that any circuit computing an ordered polynomial can be assumed to be ordered without loss of generality.

▶ **Theorem 19** (Theorem 7.1 in [11]). *Let $\mathcal{C}$ be a circuit of size s computing an ordered polynomial f of degree d. Then, there is an ordered circuit $\mathcal{C}'$ of size $O(d^3 s)$ that computes f.*

We show that the proof of this statement can be generalised to show Observation 7. A complete proof is given for the sake of completeness.

▶ **Observation 7** (Converting Circuits into Abecedarian Circuits). *Let f be an* abecedarian *polynomial with respect to a Partition of size m, and $\mathcal{C}$ be a circuit of size s computing f. Then there is an* abecedarian *circuit $\mathcal{C}'$ computing f of size $O(m^3 s)$.*

Proof. Without loss of generality, let us assume that $\mathcal{C}$ has fan-in 2.

We prove the given statement by describing how to construct $\mathcal{C}'$ from $\mathcal{C}$. For each gate v in $\mathcal{C}$, we make $O(m^2)$ copies in $\mathcal{C}'$, $\{(v, [a, b)) \ : \ 1 \leq a \leq b \leq m + 1\}$; and if root is the output gate in $\mathcal{C}$, then we define the set of output gates in $\mathcal{C}'$ to be $\{(\mathsf{root}, [i, m + 1))\}_{i \in [m+1]}$.

Intuitively, if f_v is the polynomial computed at v in $\mathcal{C}$, then the polynomial computed at $(v, [a, b))$ is $f_v[a, b)$. Thus if f was the polynomial computed at root, then the polynomial computed by $\mathcal{C}'$ is $\sum_{i=1}^{m+1} f[i, m + 1)$ which is indeed f.

We ensure this property at every gate by adding edges as follows.

- If v is an input gate labelled by a field element γ,
  - we set $(v, [a, a)) = \gamma$ for every $a \in [m + 1]$;
  - we set $(v, [a, b)) = 0$ for every $1 \leq a < b \leq m + 1$.
- If v is an input gate labelled by a variable x_i and $x_i \in X_k$,
  - we set $(v, [k, k + 1)) = x_i$;
  - we set $(v, [a, b)) = 0$ for every $a \neq k$, $b \neq k + 1$.

- If $v = v_1 + v_2$, we set $(v, [a, b)) = (v_1, [a, b)) + (v_2, [a, b))$ for every $a \leq b \in [m + 1]$.
- If $v = v_1 \times v_2$, we set $(v, [a, a)) = (v_1, [a, a)) \cdot (v_2, [a, a))$ for every $a \in [m + 1]$; and

$$(v, [a, b)) = (v_1, [a, a)) \cdot (v_2, [a, b)) + (v_1, [a, b)) \cdot (v_2, [b, b)) + \sum_{c=a}^{b-1} (v_1, [a, c + 1)) \times (v_2, [c, b))$$

  for every $1 \leq a < b \leq m + 1$.

Finally, for every $1 \leq a \leq b \leq m + 1$, we associate the gate $(v, [a, b))$ in $\mathcal{C}'$ with the set $[a, b)$.

Using induction, one can easily show that the gates in $\mathcal{C}'$ have the claimed properties. Hence $\mathcal{C}'$ is indeed an abecedarian circuit computing f. Further for every gate v in $\mathcal{C}$, there are at most $O(m^3)$ vertices in $\mathcal{C}'$. Thus the size of $\mathcal{C}'$ is $O(m^3 s)$. ◀

4.2 Algebraic Branching Programs

Next, we show that a similar statement is true for ABPs as well.

▶ **Observation 8** (Converting ABPs into Abecedarian ABPs). *Suppose f is an* abecedarian *polynomial with respect to a partition of size m. If there is an ABP $\mathcal{A}$ of size s computing it, then there is an* abecedarian *ABP $\mathcal{A}'$ computing it of size $O(ms)$.*

Proof. Let f have degree d and be abecedarian with respect to the buckets $\{X_i\}_{i=1}^m$, where $X_i = \{x_{i,j} \ : \ j \in [n_i]\}$. Without loss of generality, we can assume that $\mathcal{A}$ is homogeneous[7]. If f is not homogeneous, $\mathcal{A}$ can be thought of as a collection of homogeneous ABPs $\{\mathcal{A}_1, \ldots, \mathcal{A}_d\}$ where $\mathcal{A}_k$ computes the k-th homogeneous component of f.

We prove the theorem by describing how to construct $\mathcal{A}'$. For each vertex v in $\mathcal{A}$, make $O(m)$ copies in $\mathcal{A}'$, namely $\{(v, a) \ : \ 0 \leq a \leq m\}$. Intuitively, if $g_{(u,v)}$ is the polynomial computed between u and v in $\mathcal{A}$, then the polynomial computed between (u, a) and (v, b) in $\mathcal{A}'$ is $g_{(u,v)}[a, b + 1)$. The way we ensure this property at every vertex is by adding edges in $\mathcal{A}'$ as follows.

> For any two vertices u, v in $\mathcal{A}$, suppose there is an edge between them that is labelled with $\sum_{i\in[m]} \sum_{j\in[n_i]} \gamma_{i,j} x_{i,j}$. Then, for every $a, b \in [m]$ with $a \leq b$, add an edge from (u, a) to (v, b) with label $\sum_{i=a}^{b} \left(\sum_{j\in[n_i]} \gamma_{i,j} x_{i,j}\right)$.

Also, associate the bucket index a with the gate (v, a) in $\mathcal{A}'$.

By induction, one can easily show that the gates in $\mathcal{A}'$ have the claimed property. Hence $\mathcal{A}'$ is indeed an abecedarian ABP computing f. Further, every vertex v in $\mathcal{A}$, there are at most $O(m)$ vertices in $\mathcal{A}'$. Therefore, the size of $\mathcal{A}'$ is $O(ms)$. ◀

4.3 Formulas

Finally we show that in the case of formulas, we can prove a similar statement only when the polynomial is abecedarian with respect to a partition of *small size.* The proof is very similar to that of Lemma 5.

▶ **Theorem 2** (Converting Formulas into Abecedarian Formulas). *Let f be an* abecedarian *polynomial with respect to a partition of size m, and $\mathcal{F}$ be a formula of size s computing f. If $m = O(\log s)$, then there is an* abecedarian *formula $\mathcal{F}'$ computing f of size* poly(s)*.*

[7] Every edge is labelled by a homogeneous form.

Proof. Let us assume additionally that $\mathcal{F}$ has depth r. Now Lemma 6 implies that $r = \log(s)$ without loss of generality. By Observation 7, there is an abecedarian circuit $\mathcal{C}$ that computes f and has size at most $s' = O(s \cdot m^3)$. Further its proof implies that the depth of $\mathcal{C}$ is at most $2r$.

To convert $\mathcal{C}$ into an abecedarian formula $\mathcal{F}'$, we have to recompute a node each time it has to be reused. That is, a particular vertex in $\mathcal{C}$ has to be duplicated as many times as there are paths from the vertex to the root. Thus to upper bound the size of $\mathcal{F}'$, we need to give an upper bound on the number of distinct paths from every vertex in $\mathcal{C}$ to its root.

Let us arbitrarily choose a vertex $(v, [a, b))$ in $\mathcal{C}$, and consider the path from it to the root. Suppose the path is $(v, [a, b)) = (v_1, [a_1, b_1)) \to \cdots \to (v_\ell, [a_\ell, b_\ell)) = (\mathsf{root}, [i, m + 1))$ for some ℓ that is at most the depth of $\mathcal{C}$. Note that it must be the case that

$$i \leq a_\ell \leq \cdots \leq a_1 \leq a \leq b \leq b_1 \leq b_\ell \leq m + 1.$$

Let us define $\delta_j = a_j - a_{j+1}$ and $\delta'_j = b_{j+1} - b_j$ for $j \in [\ell - 1]$. Then, the number of choices we have for $(a_1, \ldots, a_\ell)$ and $(b_1, \ldots, b_\ell)$ such that

$$i = a_\ell \leq \cdots a_1 = a \leq b = b_1 \leq \cdots \leq b_\ell = m + 1$$

is the same as the number of choices we have for $(\delta_1, \ldots, \delta_{\ell-1}, \delta'_1, \ldots, \delta'_{\ell-1})$ such that

$$\delta_1 + \cdots + \delta_{\ell-1} + \delta'_1 + \cdots + \delta'_{\ell-1} = (m + 1 - (b - a) - i) \leq m.$$

This is clearly at most $\binom{2\ell+m}{m}$.

Further, we know that the $\mathcal{C}$ has depth $2r$ and hence $\ell \leq 2r$. Therefore, the number of paths from (v, i) to the root is at most $\binom{4r+m}{m}$. Hence if $\mathcal{F}'$ is the formula obtained by unravelling $\mathcal{C}$, then $\mathsf{size}(\mathcal{F}') \leq s' \cdot m^2 \cdot \binom{4r+m}{m}$. Here $s' = O(m^3 \cdot s)$, $r = O(\log(s))$ and $m = O(\log(s))$. Thus, $\mathsf{size}(\mathcal{F}') \leq \mathsf{poly}(s)$. ◀

5 Separating Abecedarian ABPs and Abecedarian Formulas

In this section, we prove our main theorem: a super-polynomial separataion between the powers of abecedarian formulas and ABPs. Before proceeding to the proof however, we first go over some observations that will help us with the proof.

5.1 Some Simple Observations

The two main polynomials we will be working with are linked_$\mathrm{CHSYM}_{n,d}$ and $\mathrm{CHSYM}_{n,d}$. Let us recall their definitions.

$$\mathsf{linked\_}\mathrm{CHSYM}_{n,d}(\mathbf{x}) = \sum_{i_0=1}^{n} \left(\sum_{i_0 \leq i_1 \leq \ldots \leq i_d \leq n} x_{i_0,i_1} \cdot x_{i_1,i_2} \cdots x_{i_{d-1},i_d} \right),$$

is abecedarian with respect to the partition $\{X_1, \ldots, X_n\}$ where $X_i = \{x_{i,j} \ : \ j \in [n]\}$, and

$$\mathrm{CHSYM}_{n,d}(\mathbf{x}) = \sum_{1 \leq i_1 \leq \ldots \leq i_d \leq n} x_{i_1} \cdots x_{i_d}.$$

is abecedarian with respect to the partition $\{X_i \ : \ X_i = \{x_i\}\}$.

We begin with the notion of a *linked* abecedarian formula computing linked_$\mathrm{CHSYM}_{n,d}(\mathbf{x})$.

▶ **Definition 20.** *An* abecedarian *formula computing* linked_CHSYM$_{n,d}$ *is said to be* linked *if at every gate, all the monomials occuring in the polynomial computed at that gate have the following property.*

$$x_{ij} \text{ appears right before } x_{i'j'} \text{ in the monomial} \implies j = i'.$$

The first observation shows that any abecedarian formula computing linked_CHSYM$_{n,d}(\mathbf{x})$ can be assumed to be *linked* without loss of generality.

▶ **Observation 21.** *Let $\mathcal{F}$ be a homogeneous* abecedarian *formula of size s that computes* linked_CHSYM$_{n,d}(\mathbf{x})$*, and let the multiplication gates of $\mathcal{F}$ have fan-in 2. Then there is a homogeneous* linked abecedarian *formula $\mathcal{F}'$ computing the same polynomial of size $O(s)$.*

Proof. For any leaf ℓ in $\mathcal{F}$ labelled by a variable, say $x_{i,j}$, suppose $\mathcal{P}$ is the path from ℓ to the root. Consider the set of multiplication gates on $\mathcal{P}$ whose left child is part of $\mathcal{P}$, and let v be the one that is closest to ℓ. Since $\mathcal{F}$ is abecedarian, the right child of v must be associated with a set, say $[a, b)$. If $j \neq a$, we set the label of ℓ to zero; otherwise we let it be $x_{i,j}$.

Note that this operation does not kill any *valid* monomial. Let $\mathcal{F}'$ be the formula we get by performing the above operation on every leaf of $\mathcal{F}$ that is labelled by a variable. $\mathcal{F}'$ is clearly homogeneous and abecedarian. We show that $\mathcal{F}'$ is also *linked.*

Suppose that is not the case. Then there is must be a *problematic* vertex in $\mathcal{F}'$. Let v be such a vertex of minimal height. That is, there is a monomial in the polynomial computed at v in which, say, $x_{i,j}$ appears right before $x_{i',j'}$ but $j \neq i'$. Further, the sub-formulas corresponding to the children of v are linked. Note that v must be a multiplication gate; not a leaf or an addition gate.

Let f_{left} and f_{right} be the polynomials computed at the left and right children of v respectively. Also, let $[a, b)$ be the set associated with the right child of v. Then, it must be the case that the first variable in any monomial in f_{right} looks like $x_{a,j'}$ for some j'. Further, there must be a monomial in f_{left} in which the last variable looks like $x_{i,j}$ for $j \neq a$.

Look at the leaf corresponding to this variable. Let this leaf be ℓ and let $\mathcal{P}$ be the path from ℓ to the root. Since $x_{i,j}$ is the right most variable in f_{left}, it must be the case that v is the multiplication gate that is closest to ℓ, whose left child is on $\mathcal{P}$. But then, we should have set $x_{i,j}$ to zero since $j \neq a$. Hence, such a monomial can not appear in f_{left}.

This shows that $\mathcal{F}'$ is indeed a homogeneous *linked* abecedarian formula of size at most that of $\mathcal{F}$ that computes linked_CHSYM$_{n,d}(\mathbf{x})$. ◀

The next observation shows that there is a poly-sized homogeneous abecedarian formula that computes CHSYM$_{n,\log n}(\mathbf{x})$.

▶ **Observation 22.** CHSYM$_{n/2,\log n}(\mathbf{x})$ *can be computed by a homogeneous* abecedarian *formula of size* poly(n)*.*

Proof. Consider the following polynomial over variables $\{t, x_1, \ldots, x_n\}$, where we think of t as a commuting variable and $x_1, \ldots, x_n$ as non-commuting variables.

$$f_{n,d}(\mathbf{x}) = \prod_{i=1}^{n} \left(1 + \sum_{j=1}^{d} t^j \cdot x_i^j\right)$$

Note that the coefficient of t^d in $f_{n,d}(\mathbf{x})$ is exactly CHSYM$_{n,d}(\mathbf{x})$. Further, it is not hard to see that $f_{n/2,\log n}(\mathbf{x})$ is abecedarian in terms of $\mathbf{x}$ with respect to the partition $\{X_i \,:\, X_i = \{x_i\}\}$, and that the given expression results in an abecedarian formula of size $O(n(\log n)^2)$.

Since t is a commuting variable, we can use the usual interpolation techniques [3], to get an abecedarian formula computing $\mathrm{CHSYM}_{n/2,\log n}(\mathbf{x})$ of size $O(n\log n\cdot n(\log n)^2) = O(n^2(\log n)^3) = \mathsf{poly}(n)$. Since the degree of $\mathrm{CHSYM}_{n/2,\log n}(\mathbf{x})$ is $O(\log n)$, by Lemma 5, there is a homogeneous abecedarian formula computing $\mathrm{CHSYM}_{n/2,\log n}(\mathbf{x})$ of size $\mathsf{poly}(n)$. ◀

Another simple observation is that if we are given a homogeneous abecedarian formula for an abecedarian polynomial, then we almost immediately have one for its various sub-polynomials.

▶ **Observation 23.** *Suppose there is a homogeneous* abecedarian *formula $\mathcal{F}$ computing a polynomial f that is* abecedarian *with respect to a partition of size m. Then, for any $a, b \in [m+1]$, there is a homogeneous* abecedarian *formula $\mathcal{F}_{a,b}$ of size s that computes $f[a,b)$.*

Proof. Recall that if $\mathcal{F}$ is a homogeneous abecedarian formula computing f, then $\mathcal{F}$ is in fact a set of formulas $\{\mathcal{F}_i : \mathcal{F}_i \text{ computes } f[i, m+1)\}$. Consider the formula $\mathcal{F}_a$ and set all variables that belong to buckets $\{X_b, \ldots, X_m\}$ to zero in $\mathcal{F}_a$. This operation clearly kills exactly the monomials in $f[a, m+1)$ that are not in $f[a,b)$. Thus if we call this new formula $\mathcal{F}_{a,b}$, then $\mathcal{F}_{a,b}$ is homogeneous, abecedarian and computes $f[a,b)$. ◀

The next observation is extremely crucial, since it allows us to *amplify the degree* of $\mathrm{CHSYM}_{n,d}$.

▶ **Lemma 24.** *Suppose there is a homogeneous* abecedarian *formula computing* $\mathrm{CHSYM}_{n,d}(\mathbf{x})$ *of size s, and a homogeneous* linked abecedarian *formula computing* $\mathsf{linked\_}\mathrm{CHSYM}_{n,d'}(\mathbf{x})$ *of size s'. Then, there is a homogeneous* abecedarian *formula computing* $\mathrm{CHSYM}_{n,(d\cdot d')}(\mathbf{x})$ *of size $(s\cdot s')$.*

Proof. Let $\mathcal{F}$ be the homogeneous abecedarian formula computing $\mathrm{CHSYM}_{n,d}(\mathbf{x})$ of size s, and $\mathcal{F}'$ be the homogeneous *linked* abecedarian formula computing $\mathsf{linked\_}\mathrm{CHSYM}_{n,d'}(\mathbf{x})$ of size s'. We think of the variable $x_{a,b}$ in $\mathsf{linked\_}\mathrm{CHSYM}_{n,d'}(\mathbf{x})$ as a placeholder for the sub-polynomial $\mathrm{CHSYM}_{n,d}[a,b+1)(\mathbf{x})$[8] of $\mathrm{CHSYM}_{n,d}(\mathbf{x})$. Note that there is a bijection between monomials in $\mathrm{CHSYM}_{n,(d\cdot d')}(\mathbf{x})$ and those in the polynomial we get by substituting $x_{a,b}$ in $\mathsf{linked\_}\mathrm{CHSYM}_{n,d'}(\mathbf{x})$ with $\mathrm{CHSYM}_{n,d}[a,b+1)(\mathbf{x})$.

By Observation 23, there is homogeneous abecedarian formula $\mathcal{F}_{a,b}$, of size $O(s)$ computing $\mathrm{CHSYM}_{n,d}[a,b+1)(\mathbf{x})$ for every $a,b \in [n+1]$. Thus, if we replace every leaf of $\mathcal{F}'$ labelled by $x_{a,b}$ with $\mathcal{F}_{a,b}$, then the resulting formula is a homogeneous abecedarian formula computing $\mathrm{CHSYM}_{n,(d\cdot d')}(\mathbf{x})$ of size $(s\cdot s')$. ◀

Finally, we observe that if we are given a homogeneous abecedarian formula computing the polynomial $\mathrm{CHSYM}_{(n-d+1),d}(\mathbf{x})$, then we get a homogeneous multilinear formula computing the non-commutative version of $\mathrm{ESYM}_{n,d}(\mathbf{x})$.

▶ **Observation 25.** *Consider the* elementary symmetric polynomial

$$\mathrm{ESYM}_{n,d}(\mathbf{x}) = \sum_{1\le i_1<\ldots<i_d\le n} x_{i_1}\cdots x_{i_d}.$$

If there is a homogeneous abecedarian *formula computing* $\mathrm{CHSYM}_{(n-d+1),d}(\mathbf{x})$ *of size s, then there is a homogeneous multilinear formula computing* $\mathrm{ESYM}_{n,d}(\mathbf{x})$ *of size s.*

[8] Sum of monomials in $\mathrm{CHSYM}_{n,d}(\mathbf{x})$ whose first variable is a and last variable is one of $\{x_a,\ldots,x_b\}$.

Proof. Suppose $\mathcal{F}$ is a homogeneous abecedarian formula computing $\mathrm{CHSYM}_{(n-d+1),d}(\mathbf{x})$ of size s. Since $\mathcal{F}$ is homogeneous, every leaf labelled by a variable can be associated with a position index. If a leaf labelled x_i has position k associated with it, then replace the label of that leaf with x_{i+k-1}. Call this formula $\mathcal{F}'$. Then clearly $\mathcal{F}'$ is a homogeneous formula of size s computing $\mathrm{ESYM}_{n,d}(\mathbf{x})$. Further note that since $\mathcal{F}$ was abecedarian, $\mathcal{F}'$ is multilinear. ◀

5.2 Proof of the Separation

We now prove Theorem 1. Let us first recall the statement.

▶ **Theorem 1** (Separating Abecedarian Formulas and Abecedarian ABPs). *Define*

$$\mathsf{linked\_CHSYM}_{n,d}(\mathbf{x}) = \sum_{i_0=1}^{n} \left(\sum_{i_0 \leq i_1 \leq \ldots \leq i_d \leq n} x_{i_0,i_1} \cdot x_{i_1,i_2} \cdots x_{i_{d-1},i_d} \right)$$

to be the linked *complete homogeneous polynomial over n-variables of degree d. This polynomial is* abecedarian *with respect to the partition $\{X_i \ : \ i \in [n]\}$ if $X_i = \{x_{i,j} \ : \ i \leq j \leq n\}$.*

With respect to this partition,

1. $\mathsf{linked\_CHSYM}_{n,d}(\mathbf{x})$ *has an* abecedarian *ABP of size $O(nd)$;*
2. *any* abecedarian *formula computing* $\mathsf{linked\_CHSYM}_{n/2,\log n}(\mathbf{x})$ *has size $n^{\Omega(\log\log n)}$.*

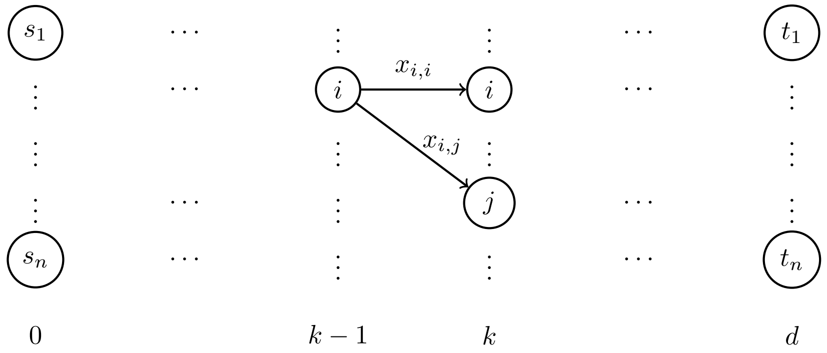
That is, there is a super-polynomial separation between abecedarian *formulas and ABPs.*

That $\mathsf{linked\_CHSYM}_{n,d}(\mathbf{x})$ has a *small* abecedarian ABP is not very hard to see. For the lower bound, we assume that we have been given an abecedarian formula $\mathcal{F}$, computing the polynomial $\mathsf{linked\_CHSYM}_{n,\log n}(\mathbf{x})$, of size $\mathsf{poly}(n)$. We then keep making changes to this formula till we get a homogeneous multilinear formula computing $\mathrm{ESYM}_{n,n/2}(\mathbf{x})$ of size $\mathsf{poly}(n)$. Finally, we use the following theorem of Hrubeš and Yehudayoff [12] to get a contradiction.

▶ **Theorem 26** (Theorem 1, [12]). *Any homogeneous multilinear formula that computes $\mathrm{ESYM}_{n,d}(\mathbf{x})$, for $d \leq n/2$, must have size $n \times d^{\Omega(\log d)}$.*

Let us now complete the proof of our main theorem.

Proof of Theorem 1. An abecedarian ABP of size $O(nd)$ computing $\mathsf{linked\_CHSYM}_{n,d}(\mathbf{x})$ is the following.



The ABP has $d+1$ layers, labelled 0 through d, each with n nodes. Between any consecutive layers $k-1$ and k, where $1 \leq k \leq d$, there is an edge from the i-th node in layer $k-1$ to the j-th node in layer k layer if $i \leq j$. The label on this edge is $x_{i,j}$. All the nodes in the first layer are start nodes, and all the ones in the last layer are terminal nodes.

It is easy to check, by induction, that the polynomial computed between s_a and the b-th vertex in layer k computes $\mathrm{CHSYM}_{n,k}[a,b+1)(\mathbf{x})$. Thus the polynomial computed by the abecedarian ABP constructed above is indeed $\mathrm{CHSYM}_{n,d}(\mathbf{x})$, and its size is clearly $O(nd)$.

Let us now move on to proving the lower bound against abecedarian formulas. We show that there is a fixed constant ϵ_0 such that any abecedarian formula that computes the polynomial linked_CHSYM$_{n/2,\log n}(\mathbf{x})$ must have size atleast $\Omega(n^{\epsilon_0 \log\log n})$. Suppose this is not the case. Then for every $\epsilon > 0$, there is an abecedarian formula $\mathcal{F}'(\epsilon)$ of size $O(n^{\epsilon\log\log n})$ that computes linked_CHSYM$_{n/2,\log n}(\mathbf{x})$.

Without loss of generality, we can assume that $\mathcal{F}'(\epsilon)$ has fan-in 2. Further, by Lemma 6, we can reduce the depth of $\mathcal{F}'(\epsilon)$ to log-depth. That is, we get an abecedarian formula $\mathcal{F}'_1(\epsilon)$ computing linked_CHSYM$_{n/2,\log n}(\mathbf{x})$ of depth $O(\epsilon \log n \log\log n)$ and size $O(n^{c_1\epsilon\log\log n})$. Here c_1 is a fixed constant independent of ϵ.

Next, since the degree of the polynomial being computed is *small*, Lemma 5 implies that $\mathcal{F}'_1(\epsilon)$ can in fact be homogenised without much blow-up in size. In other words, there is a homogeneous abecedarian formula computing linked_CHSYM$_{n/2,\log n}(\mathbf{x})$ of size $O(n^{c_1c_2\epsilon\log\log n})$, where c_2 is again a fixed constant independent of ϵ. Let this formula be $\mathcal{F}'_2(\epsilon)$.

By Observation 21, we can then use $\mathcal{F}'_2(\epsilon)$ to get a homogeneous linked abecedarian formula $\mathcal{F}'_3(\epsilon)$ of size $O(n^{c_1c_2\epsilon\log\log n})$ that computes the same polynomial. Further, because of Observation 22, we know that there is a homogeneous abecedarian formula, say $\mathcal{F}$, of size $\mathsf{poly}(n) = O(n^{c_1c_2\epsilon\log\log n})$ that computes $\mathrm{CHSYM}_{n/2,\log n}(\mathbf{x})$.

With $\mathcal{F}$ and $\mathcal{F}'_3(\epsilon)$ in hand, we get a homogeneous abecedarian formula $\mathrm{CHSYM}_{n/2,\log^2 n}(\mathbf{x})$ because of Lemma 24. To get such a formula for $\mathrm{CHSYM}_{n/2,n/2}(\mathbf{x})$, we need to use Lemma 24 at most k times where

$$(\log n)^k = \frac{n}{2} \implies k = O\left(\frac{\log n}{\log\log n}\right).$$

Thus, using Lemma 24 repeatedly at most $O({}^{\log n}/_{\log\log n})$ times, we get that there is a homogeneous abecedarian formula, $\mathcal{F}(\epsilon)$, computing $\mathrm{CHSYM}_{n/2,n/2}(\mathbf{x})$ of size

$$O(n^{(c_1c_2\epsilon\log\log n)\cdot({}^{\log n}/_{\log\log n})}) = O(n^{(c_1c_2\epsilon\log n)}).$$

By Observation 25, we know that $\mathcal{F}(\epsilon)$ can be used to get a homogeneous multilinear formula, $\mathcal{F}_1(\epsilon)$, computing $\mathrm{ESYM}_{n-1,n/2}(\mathbf{x})$ of size $O(n^{(c_1c_2\epsilon\log n)})$. Finally, Theorem 26 tells us that there is a constant δ such that any homogeneous multilinear formula computing $\mathrm{ESYM}_{n-1,n/2}(\mathbf{x})$ must have size at least $n^{\delta\cdot\log n}$. For $\epsilon = {}^{\delta}/_{2c_1c_2}$, this contradicts the existence of $\mathcal{F}_1(\epsilon)$ and hence $\mathcal{F}'(\epsilon)$. Thus, it must be the case that any abecedarian formula computing linked_CHSYM$_{n/2,\log n}(\mathbf{x})$ has size at least $n^{\Omega(\log\log n)}$. This completes the proof. ◀

6 Proofs of the Remaining Statements

In this section we give proof ideas of the remaining statements mentioned in the introduction.

6.1 Formula Lower Bounds from Structured Formula Lower Bounds

▶ **Corollary 3.** *Let the polynomial* linked_CHSYM$_{n,d}(\mathbf{x})$ *be as defined in Theorem 1. An* $n^{\omega(1)}$ *lower bound against* abecedarian *formulas for* linked_CHSYM$_{\log n,n}(\mathbf{x})$ *would imply a super-polynomial separation between non-commutative ABPs and formulas.*

Proof. By Theorem 1, we know that the ABP complexity of $\mathsf{linked\_}\mathrm{CHSYM}_{\log n,n}(\mathbf{x})$ is $\mathsf{poly}(n)$. Therefore any formula computing the polynomial must have size at least $n^{\Omega(1)}$. Further, note that the polynomial is abecedarian with respect to a partition of size $O(\log n)$. Therefore, by Theorem 2, if there is a formula $\mathcal{F}$ computing $\mathsf{linked\_}\mathrm{CHSYM}_{\log n,n}(\mathbf{x})$ of size s, then there is an abecedarian formula computing it of size $\mathsf{poly}(s)$. This immediately implies the given statement. ◀

▶ **Corollary 4.** *Let $\mathsf{linked\_}\mathrm{CHSYM}_{n,d}(\mathbf{x})$ be as defined in Theorem 1. An $n^{\omega(1)}$ lower bound against homogeneous formulas for $\mathsf{linked\_}\mathrm{CHSYM}_{n,\log n}(\mathbf{x})$ would result in a super-polynomial separation between ABPs and formulas in the non-commutative setting.*

Proof. By Theorem 1, we know that the ABP complexity of $\mathsf{linked\_}\mathrm{CHSYM}_{n,\log n}$ is $\mathsf{poly}(n)$. Further, the degree of the polynomial is $O(\log n)$. Thus, by Lemma 5, if there is a formula computing $\mathsf{linked\_}\mathrm{CHSYM}_{n,\log n}(\mathbf{x})$ of size s, then there is a homogeneous formula computing it of size $\mathsf{poly}(s)$. This immediately implies the given statement. ◀

▶ **Corollary 11.** *An $n^{\omega(1)}$ lower bound against homogeneous formulas computing the n-variate iterated matrix multiplication polynomial of degree $\log n$, $\mathrm{IMM}_{n,\log n}(\mathbf{x})$, implies a super-polynomial separation between ABPs and formulas in the non-commutative setting.*

Proof. Clearly, the ABP complexity of $\mathrm{IMM}_{n,\log n}(\mathbf{x})$ is $\mathsf{poly}(n)$. Thus, by Lemma 5, if there is a formula computing $\mathrm{IMM}_{n,\log n}(\mathbf{x})$ of size s, then there is a homogeneous formula computing it of size $\mathsf{poly}(s)$. This immediately implies the given statement. ◀

6.2 Known Relations in the Non-Commutative Setting that Continue to Hold with the Abecedarian Restriction

▶ **Observation 9** (The Usual Inclusions). *Let abc - VP_{nc}, abc - VBP_{nc} and abc - VF_{nc} denote the classes of abecedarian polynomials over n variables that can be computed by $\mathsf{poly}(n)$ sized abecedarian circuits, abecedarian ABPs and abecedarian formulas respectively. Then,*

$$\mathsf{abc}\text{ - }\mathsf{VF}_{nc} \subseteq \mathsf{abc}\text{ - }\mathsf{VBP}_{nc} \subseteq \mathsf{abc}\text{ - }\mathsf{VP}_{nc}.$$

Proof. Suppose $f \in \mathsf{abc}$ - $\mathsf{VF}_{\mathrm{nc}}$. Then f is abecedarian, and in particular $f \in \mathsf{VF}_{\mathrm{nc}}$. But we know that $\mathsf{VF}_{\mathrm{nc}} \subseteq \mathsf{VBP}_{\mathrm{nc}}$, and so $f \in \mathsf{VBP}_{\mathrm{nc}}$. By Observation 8, this implies that $f \in \mathsf{abc}$ - $\mathsf{VBP}_{\mathrm{nc}}$.

Similarly, suppose $f \in \mathsf{abc}$ - $\mathsf{VBP}_{\mathrm{nc}}$. Then f is abecedarian, and $f \in \mathsf{VBP}_{\mathrm{nc}}$. But $\mathsf{VBP}_{\mathrm{nc}} \subseteq \mathsf{VP}_{\mathrm{nc}}$, and so $f \in \mathsf{VP}_{\mathrm{nc}}$. By Observation 7, this implies that $f \in \mathsf{abc}$ - $\mathsf{VP}_{\mathrm{nc}}$. ◀

▶ **Observation 10** (Converting Abecedarian ABPs into Abecedarian Formulas). *Suppose f is an abecedarian polynomial of degree d. If there is an abecedarian ABP $\mathcal{A}$ of size s computing it, then there is an abecedarian formula $\mathcal{F}$ computing f of size $O(s^{\log d})$.*

Proof. The formula we get using the usual divide-and-conquer algorithm has the property that polynomials computed at any of its gate is a polynomial computed between two vertices in the ABP. Thus by definition of abecedarian ABPs, the statement follows via the usual algorithm. ◀

References

1 V. Arvind, P. S. Joglekar, and S. Raja. Noncommutative valiant's classes: Structure and complete problems. *ACM Trans. Comput. Theory*, 9(1), 2016. doi:10.1145/2956230.

2 Vikraman Arvind and Srikanth Srinivasan. On the hardness of the noncommutative determinant. *Comput. Complex.*, 27(1):1–29, 2018. doi:10.1007/s00037-016-0148-5.

3 Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992. doi:10.1137/0221006.

4 Richard P. Brent. The parallel evaluation of general arithmetic expressions. *Journal of the ACM*, 21(2):201–206, 1974. doi:10.1145/321812.321815.

5 Marco L. Carmosino, Russell Impagliazzo, Shachar Lovett, and Ivan Mihajlin. Hardness amplification for non-commutative arithmetic circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC*, volume 102 of *LIPIcs*, pages 12:1–12:16, 2018. doi:10.4230/LIPIcs.CCC.2018.12.

6 Prerona Chatterjee, Mrinal Kumar, Adrian She, and Ben Lee Volk. A quadratic lower bound for algebraic branching programs and formulas. *CoRR*, 1911.11793v2, 2019. arXiv:1911.11793v2.

7 Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 615–624. ACM, 2012. doi:10.1145/2213977.2214034.

8 Nathanaël Fijalkow, Guillaume Lagarde, Pierre Ohlmann, and Olivier Serre. Lower bounds for arithmetic circuits via the hankel matrix. In *37th International Symposium on Theoretical Aspects of Computer Science, STACS*, volume 154 of *LIPIcs*, pages 24:1–24:17, 2020. doi:10.4230/LIPIcs.STACS.2020.24.

9 Pavel Hrubes and Avi Wigderson. Non-commutative arithmetic circuits with division. *Theory Comput.*, 11:357–393, 2015. doi:10.4086/toc.2015.v011a014.

10 Pavel Hrubes, Avi Wigderson, and Amir Yehudayoff. Relationless completeness and separations. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC*, pages 280–290. IEEE Computer Society, 2010. doi:10.1109/CCC.2010.34.

11 Pavel Hrubeš, Avi Wigderson, and Amir Yehudayoff. Non-commutative circuits and the sum-of-squares problem. *Journal of the American Mathematical Society*, 24(3):871–898, 2011. URL: https://www.ams.org/journals/jams/2011-24-03/S0894-0347-2011-00694-2/S0894-0347-2011-00694-2.pdf.

12 Pavel Hrubes and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Comput. Complex.*, 20(3):559–578, 2011. doi:10.1007/s00037-011-0007-3.

13 L. Hyafil. The power of commutativity. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 171–174, 1977. doi:10.1109/SFCS.1977.31.

14 K. Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985. doi:10.1137/0214050.

15 Guillaume Lagarde, Nutan Limaye, and Srikanth Srinivasan. Lower bounds and PIT for non-commutative arithmetic circuits with restricted parse trees. *Comput. Complex.*, 28(3):471–542, 2019. doi:10.1007/s00037-018-0171-9.

16 Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. Non-commutative computations: lower bounds and polynomial identity testing. *Chic. J. Theor. Comput. Sci.*, 2019, 2019. URL: http://cjtcs.cs.uchicago.edu/articles/2019/2/contents.html.

17 Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for non-commutative skew circuits. *Theory of Computing*, 12(1):1–38, 2016. doi:10.4086/toc.2016.v012a012.

18 Merriam and Webster. Definition of abecedarian. Word of the Day at www.merriam-webster.com, 2019. URL: https://www.merriam-webster.com/word-of-the-day/abecedarian-2019-03-06.

19 Eduard Ivanovich Nechiporuk. On a boolean function. *Dokl. Akad. Nauk SSSR*, 169:765–766, 1966. URL: http://mi.mathnet.ru/dan32449.

20 Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 410–418. ACM, 1991. doi:10.1145/103418.103462.

21 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013. doi:10.1145/2535928.

22 Ramprasad Saptharishi and Anamay Tengse. Quasipolynomial hitting sets for circuits with restricted parse trees. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, volume 122 of *LIPIcs*, pages 6:1–6:19, 2018. doi:10.4230/LIPIcs.FSTTCS.2018.6.

23 Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

The (Generalized) Orthogonality Dimension of (Generalized) Kneser Graphs: Bounds and Applications

Alexander Golovnev
Georgetown University, Washington, DC, USA

Ishay Haviv
School of Computer Science, The Academic College of Tel Aviv-Yaffo, Israel

Abstract

The *orthogonality dimension* of a graph $G = (V, E)$ over a field $\mathbb{F}$ is the smallest integer t for which there exists an assignment of a vector $u_v \in \mathbb{F}^t$ with $\langle u_v, u_v \rangle \neq 0$ to every vertex $v \in V$, such that $\langle u_v, u_{v'} \rangle = 0$ whenever v and v' are adjacent vertices in G. The study of the orthogonality dimension of graphs is motivated by various applications in information theory and in theoretical computer science. The contribution of the present work is two-fold.

First, we prove that there exists a constant c such that for every sufficiently large integer t, it is NP-hard to decide whether the orthogonality dimension of an input graph over $\mathbb{R}$ is at most t or at least $3t/2 - c$. At the heart of the proof lies a geometric result, which might be of independent interest, on a *generalization* of the orthogonality dimension parameter for the family of *Kneser graphs*, analogously to a long-standing conjecture of Stahl (J. Comb. Theo. Ser. B, 1976).

Second, we study the smallest possible orthogonality dimension over finite fields of the complement of graphs that do not contain certain fixed subgraphs. In particular, we provide an explicit construction of triangle-free n-vertex graphs whose complement has orthogonality dimension over the binary field at most $n^{1-\delta}$ for some constant $\delta > 0$. Our results involve constructions from the family of *generalized Kneser graphs* and they are motivated by the rigidity approach to circuit lower bounds. We use them to answer a couple of questions raised by Codenotti, Pudlák, and Resta (Theor. Comput. Sci., 2000), and in particular, to disprove their Odd Alternating Cycle Conjecture over every finite field.

2012 ACM Subject Classification Mathematics of computing → Graph coloring; Mathematics of computing → Approximation algorithms; Theory of computation → Circuit complexity

Keywords and phrases Orthogonality dimension, minrank, rigidity, hardness of approximation, circuit complexity, chromatic number, Kneser graphs

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.8

Related Version *Full Version*: `https://arxiv.org/abs/2002.08580` [19]

Funding *Alexander Golovnev*: Partially supported by a Rabin Postdoctoral Fellowship.
Ishay Haviv: Research supported in part by the Israel Science Foundation (grant No. 1218/20).

1 Introduction

A t-dimensional *orthogonal representation* of a graph $G = (V, E)$ over a field $\mathbb{F}$ is an assignment of a vector $u_v \in \mathbb{F}^t$ with $\langle u_v, u_v \rangle \neq 0$ to every vertex $v \in V$, such that $\langle u_v, u_{v'} \rangle = 0$ whenever v and v' are adjacent vertices in G. The *orthogonality dimension* of a graph G over $\mathbb{F}$, denoted by $\overline{\xi}(G, \mathbb{F})$, is the smallest integer t for which there exists a t-dimensional orthogonal representation of G over $\mathbb{F}$. The orthogonality dimension parameter is closely related to

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 8; pp. 8:1–8:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

several other well-studied graph parameters. In particular, for every graph G and every field $\mathbb{F}$, $\overline{\xi}(G,\mathbb{F})$ is sandwiched between the clique number and the chromatic number of G, that is, $\omega(G) \leq \overline{\xi}(G,\mathbb{F}) \leq \chi(G)$.[1]

Orthogonal representations of graphs have been found useful over the years for various applications in information theory and in theoretical computer science. They were originally introduced over the real field in a seminal work of Lovász [32], where they were used to define the influential Lovász ϑ-function. The latter was used in [32] to determine the Shannon capacity, a notoriously difficult information-theoretic graph parameter, of the cycle on five vertices, and in the last decades it was successfully applied in algorithmic and combinatorial results (see, e.g., [28, 17, 3]). The orthogonality dimension of graphs plays an important role in several areas of computational complexity. Over finite fields, the orthogonality dimension and its extension due to Haemers [21] to a graph parameter called *minrank* have attracted a significant attention in circuit complexity, and more specifically, in the study of Valiant's rigidity approach to circuit lower bounds [43] (see, e.g., [11, 37, 20]). Over the complex field, the orthogonality dimension was used in a characterization of the quantum communication complexity of promise equality problems [12, 4, 5] and in the study of the quantum chromatic number [8, 39]. The orthogonality dimension parameter was also investigated in the contexts of hardness of approximation [36, 29], integrality gaps for linear programming [26, 25], and algorithms based on semi-definite programming [9, 23].

The present work studies two aspects of the orthogonality dimension of graphs. First, we prove an NP-hardness result for approximating the orthogonality dimension of graphs over the real field $\mathbb{R}$. At the heart of the proof lies a geometric result, which might be of independent interest, on a *generalization* of the orthogonality dimension parameter for the family of *Kneser graphs*, analogously to a long-standing graph-theoretic conjecture due to Stahl [40]. The second aspect of the orthogonality dimension parameter considered in this work, motivated by the area of circuit complexity, is that of determining the smallest possible orthogonality dimension over finite fields of the complement of graphs that do not contain certain fixed subgraphs. In this context, we prove a new bound on the minrank parameter over finite fields for the family of *generalized Kneser graphs*. The bound is used to settle a couple of questions raised by Codenotti, Pudlák, and Resta in [11] and to disprove their Odd Alternating Cycle Conjecture over every finite field.

1.1 Our Contribution

1.1.1 The Generalized Orthogonality Dimension of Kneser Graphs

We start by considering the computational hardness of determining the orthogonality dimension of graphs over the real field $\mathbb{R}$. The challenge of understanding the hardness of this parameter was posed already in the late eighties by Lovász, Saks, and Schrijver [34] (see also [33, Chapter 10]), and yet, the problem is far from being well-understood. It is easy to see that deciding whether an input graph G satisfies $\overline{\xi}(G,\mathbb{R}) \leq t$ can be solved in polynomial running-time for $t \in \{1,2\}$, and Peeters [36] has shown that it is NP-hard for $t \geq 3$. His result is known to imply that for every $t \geq 6$ it is NP-hard to decide whether an input graph G satisfies $\overline{\xi}(G,\mathbb{R}) \leq t$ or $\overline{\xi}(G,\mathbb{R}) \geq \lceil 4t/3 \rceil$ (see [23]). In the current work, we improve on the 4/3 multiplicative gap and prove the following.

[1] Orthogonal representations of graphs are sometimes defined in the literature as orthogonal representations of the complement, namely, the definition requires vectors associated with *non-adjacent* vertices to be orthogonal. We have decided to use here the other definition, but one may view the notation $\overline{\xi}(G,\mathbb{F})$ as standing for $\xi(\overline{G},\mathbb{F})$.

▶ **Theorem 1.** *There exists a constant c such that for every sufficiently large integer t, it is* NP*-hard to decide whether an input graph G satisfies $\overline{\xi}(G,\mathbb{R}) \leq t$ or $\overline{\xi}(G,\mathbb{R}) \geq 3t/2 - c$.*

It is worth noting that in order to obtain hardness results for the orthogonality dimension parameter, it is natural to employ known hardness results regarding the closely related chromatic number of graphs. Indeed, it is easy to verify (see, e.g., [23]) that every graph G satisfies

$$\log_3 \chi(G) \leq \overline{\xi}(G,\mathbb{R}) \leq \chi(G),$$

hence hardness of deciding whether an input graph G satisfies $\chi(G) \leq t_1$ or $\chi(G) \geq t_2$ immediately implies the hardness of deciding whether it satisfies $\overline{\xi}(G,\mathbb{R}) \leq t_1$ or $\overline{\xi}(G,\mathbb{R}) \geq \log_3 t_2$. In particular, a result of Dinur, Mossel, and Regev [14] on the hardness of the chromatic number implies that assuming a certain variant of the unique games conjecture, deciding whether a given graph G satisfies $\overline{\xi}(G,\mathbb{R}) \leq 3$ or $\overline{\xi}(G,\mathbb{R}) \geq t$ is NP-hard for every $t \geq 4$. However, if one is interested in standard NP-hardness for the orthogonality dimension, the state of the art for the hardness of the chromatic number does not seem to imply any hardness results, despite some remarkable recent progress [7, 44]. Moreover, most hardness proofs for the chromatic number crucially use the fact that an upper bound on the independence number of a graph implies a strong lower bound on its chromatic number (namely, $\chi(G) \geq \frac{|V(G)|}{\alpha(G)}$), whereas an analogue of such a statement for the orthogonality dimension does not hold in general (see, e.g., [23, Proposition 2.2]).

One technique for proving hardness results for the chromatic number that can be applied for the orthogonality dimension is that of Garey and Johnson [18], who have related hardness of graph coloring to the *multichromatic numbers of Kneser graphs*. The kth multichromatic number of a graph G, denoted by $\chi_k(G)$, is the smallest number of colors needed in order to assign to every vertex of G a set of k colors so that adjacent vertices are assigned to disjoint sets. Notice that $\chi_1(G)$ is simply the standard chromatic number $\chi(G)$. The family of Kneser graphs is defined as follows.

▶ **Definition 2** (Kneser Graphs)**.** *For integers $d \geq 2s$, the* Kneser graph *$K(d,s)$ is the graph whose vertices are all the s-subsets of $[d] = \{1,\ldots,d\}$, where two sets are adjacent if they are disjoint.*

Note that the multichromatic numbers can be defined in terms of Kneser graphs, namely, $\chi_k(G)$ is the smallest integer d for which there exists a homomorphism from G to $K(d,k)$.

In the seventies, Stahl [40] has made the following conjecture.

▶ **Conjecture 3** (Stahl's Conjecture [40])**.** *For all integers k and $d \geq 2s$,*

$$\chi_k(K(d,s)) = \left\lceil \frac{k}{s} \right\rceil \cdot (d-2s) + 2k.$$

Stahl's conjecture has received a significant attention in the literature over the years. Even very recently, it was related to the well-known recently disproved Hedetniemi's conjecture [42]. Nevertheless, more than forty years since it was proposed, Stahl's conjecture is still open. It is known that the right-hand side in Conjecture 3 forms an upper bound on $\chi_k(K(d,s))$, and that this bound is tight up to an additive constant that depends solely on s [10, 41]. The precise statement of the conjecture was confirmed only for a few special cases. This includes the case of $k = 1$ proved by Lovász [31], the cases of $s \leq 2$, $k \leq s$, $d = 2s+1$, and k divisible by s proved by Stahl [40, 41], and the case of $s = 3$ and $k = 4$ proved by Garey and Johnson [18] (extended to $s = 3$ with any k in [41]). The result of [18] was combined there with a simple reduction to show that for every $t \geq 6$, it is NP-hard to decide whether a given graph G satisfies $\chi(G) \leq t$ or $\chi(G) \geq 2t - 4$.

The recent work [23] has suggested to borrow the reduction of [18] to prove hardness results for the orthogonality dimension parameter. This approach requires the following generalization of orthogonal representations of graphs over the reals.

▶ **Definition 4** (Orthogonal Subspace Representation). *A t-dimensional* orthogonal k-subspace representation *of a graph $G = (V, E)$ is an assignment of a subspace $U_v \subseteq \mathbb{R}^t$ with $\dim(U_v) = k$ to every vertex $v \in V$, such that the subspaces U_v and $U_{v'}$ are orthogonal whenever v and v' are adjacent in G. For a graph G, let $\overline{\xi}_k(G, \mathbb{R})$ denote the smallest integer t for which there exists a t-dimensional orthogonal k-subspace representation of G.*[2]

Note that for $k = 1$, Definition 4 coincides with the orthogonality dimension over the reals, and that for every graph G and every k it holds that $\overline{\xi}_k(G, \mathbb{R}) \leq \chi_k(G)$.

A combination of the hardness result of Peeters [36] and the reduction of [18] implies the following.

▶ **Proposition 5** ([23, Theorem 1.3]). *For every graph F, it is* NP*-hard to decide whether an input graph G satisfies $\overline{\xi}(G, \mathbb{R}) \leq \overline{\xi}_3(F, \mathbb{R})$ or $\overline{\xi}(G, \mathbb{R}) \geq \overline{\xi}_4(F, \mathbb{R})$.*

With Proposition 5 in hand, it is of interest to find graphs F with a large gap between $\overline{\xi}_3(F, \mathbb{R})$ and $\overline{\xi}_4(F, \mathbb{R})$. In light of Conjecture 3, it is natural to consider the generalized orthogonality dimension parameters for the family of Kneser graphs. For $k = 1$, it was shown in [25] that the standard chromatic number and the standard orthogonality dimension over $\mathbb{R}$ coincide on all Kneser graphs. In addition, a result of Bukh and Cox [6, Proposition 23] implies that for every $d \geq 2s$ and every k, $\overline{\xi}_k(K(d, s), \mathbb{R}) \geq kd/s$. This implies that the kth chromatic number and the kth orthogonality dimension over $\mathbb{R}$ coincide on $K(d, s)$ whenever k is divisible by s.

In this work we initiate a systematic study of the generalized orthogonality dimension parameters of Kneser graphs, analogously to Conjecture 3. Let us already mention that the arguments applied in the study of Stahl's conjecture do not seem to extend to our question. The main reason is that the proofs in [40, 18, 10, 41] use Hilton-Milner-type theorems to characterize the possible structures of the independent sets induced by generalized colorings of Kneser graphs, whereas in our setting, orthogonal subspace representations do not naturally induce large independent sets and the problem seems to require a more geometric approach.

The first non-trivial case is that of Kneser graphs $K(d, s)$ with $s = 2$, for which we show that the generalized orthogonality dimension parameters are equal to the multichromatic numbers.

▶ **Theorem 6.** *For all integers $k \geq 1$ and $d \geq 4$, $\overline{\xi}_k(K(d, 2), \mathbb{R}) = \left\lceil \frac{k}{2} \right\rceil \cdot (d - 4) + 2k$.*

We proceed by considering a general $s \geq 3$ and prove the following lower bound.

▶ **Theorem 7.** *For every integers $k \geq s \geq 3$ there exists $c = c(s, k)$ such that for all integers $d \geq 2s$,*

$$\overline{\xi}_k(K(d, s), \mathbb{R}) \geq \frac{k - \lceil \frac{k+1}{s} \rceil + 1}{s - 1} \cdot d - c.$$

Note that for $k = \ell \cdot s - 1$ where ℓ is an integer, the bound provided by Theorem 7 is tight up to the additive constant c. Indeed, in this case we get that there exists a constant c such that for all integers $d \geq 2s$ it holds that

$$\ell \cdot d - c \leq \overline{\xi}_{\ell \cdot s-1}(K(d, s), \mathbb{R}) \leq \chi_{\ell \cdot s-1}(K(d, s)) \leq \ell \cdot d - 2.$$

[2] Over the complex field, the definition is equivalent to the notion of a projective representation from [35, Definition 6.1].

Note further that for the special case of $k = 4$ and $s = 3$, Theorem 7 implies that there exists a constant c such that $\overline{\xi}_4(K(d, 3), \mathbb{R}) \geq 3d/2 - c$ for every sufficiently large integer d. This, combined with Proposition 5 and the fact that $\overline{\xi}_3(K(d, 3), \mathbb{R}) = d$, yields our hardness result Theorem 1.

It will be interesting to figure out if the bounds given in Theorem 7 can be tightened to the quantity given in the right-hand side of Conjecture 3, at least up to an additive term independent of d. In particular, it will be nice to decide whether for all integers $d \geq 6$ it holds that $\overline{\xi}_4(K(d, 3), \mathbb{R}) = 2d - 4$. A positive answer would imply that for every $t \geq 6$, it is NP-hard to decide whether an input graph G satisfies $\overline{\xi}(G) \leq t$ or $\overline{\xi}(G) \geq 2t - 4$. We remark, however, that the approach suggested by Proposition 5 for the hardness of the orthogonality dimension cannot yield a multiplicative hardness gap larger than 2, as it is easy to see that every graph F satisfies $\overline{\xi}_4(F, \mathbb{R}) \leq \overline{\xi}(F, \mathbb{R}) + \overline{\xi}_3(F, \mathbb{R}) \leq 2 \cdot \overline{\xi}_3(F, \mathbb{R})$.

1.1.2 The Orthogonality Dimension of Generalized Kneser Graphs

We next consider the orthogonality dimension over finite fields of the complement of graphs that do not contain some fixed subgraphs. In fact, in this context we consider an extension of the orthogonality dimension parameter, called minrank, that was introduced by Haemers in [21] and is defined as follows.

▶ **Definition 8** (Minrank). *Let $G = (V, E)$ be a directed graph on the vertex set $V = [n]$ and let $\mathbb{F}$ be a field. We say that an n by n matrix M over $\mathbb{F}$* represents *G if $M_{i,i} \neq 0$ for every $i \in V$, and $M_{i,j} = 0$ for every distinct $i, j \in V$ such that $(i, j) \notin E$. The* minrank *of G over $\mathbb{F}$ is defined as*

$$\mathrm{minrk}_{\mathbb{F}}(G) = \min\{\mathrm{rank}_{\mathbb{F}}(M) \mid M \textit{ represents } G \textit{ over } \mathbb{F}\}.$$

The definition is naturally extended to (undirected) graphs by replacing every undirected edge with two oppositely directed edges.

Note that for every graph G and every field $\mathbb{F}$, $\mathrm{minrk}_{\mathbb{F}}(\overline{G}) \leq \overline{\xi}(G, \mathbb{F})$.[3]

We consider here the question of whether there are graphs with no short odd cycles and yet low minrank over finite fields. This question is motivated by the area of circuit complexity, and more specifically, by Valiant's approach to circuit lower bounds [43], as described next. The *rigidity* of an n by n matrix M over a field $\mathbb{F}$ with respect to a given parameter r is the smallest number of entries that one has to change in M in order to reduce its rank over $\mathbb{F}$ to below r. Roughly speaking, it was shown in [43] that n by n matrices with large rigidity for $r = \varepsilon \cdot n$ where $\varepsilon > 0$ is a constant can be used to obtain superlinear lower bounds on the size of logarithmic depth arithmetic circuits computing linear transformations. In 2000, Codenotti, Pudlák, and Resta [11] have proposed the *Odd Alternating Cycle Conjecture*, stated below. By an *alternating odd cycle* we refer to a directed graph which forms an odd cycle when the orientation of the edges is ignored, and such that the orientation of the edges alternates with one exception.

▶ **Conjecture 9** (The Odd Alternating Cycle Conjecture [11]). *For every field $\mathbb{F}$ there exist $\varepsilon > 0$ and an odd integer ℓ such that every n-vertex directed graph G with $\mathrm{minrk}_{\mathbb{F}}(G) \leq \varepsilon \cdot n$ contains an alternating cycle of length ℓ.*

[3] Indeed, given a t-dimensional orthogonal representation of an n-vertex graph G over a field $\mathbb{F}$, consider the matrix $B \in \mathbb{F}^{n \times t}$ whose rows contain the vectors associated with the vertices of G. Then, the n by n matrix $B \cdot B^T$ represents $\overline{G}$ and has rank at most t over $\mathbb{F}$, hence $\mathrm{minrk}_{\mathbb{F}}(\overline{G}) \leq t$.

It was proved in [11] that Conjecture 9 implies, if true, that certain explicit circulant matrices have superlinear rigidity. In contrast, for $\ell = 3$ it was shown in [11] that there are n-vertex (undirected) triangle-free graphs G satisfying $\mathrm{minrk}_{\mathbb{F}}(G) \leq O(n^{3/4})$ for every field $\mathbb{F}$, and it was left open whether the statement of Conjecture 9 may hold for larger values of ℓ. In the recent work [24] the conjecture was disproved over the real field, but remained open for finite fields which are of special interest in circuit complexity. For the orthogonality dimension over the binary field $\mathbb{F}_2$, it was shown in [11] that there exist triangle-free n-vertex graphs G satisfying $\overline{\xi}(\overline{G}, \mathbb{F}_2) = n/4 + 2$. It was asked there whether every n-vertex graph G satisfying $\overline{\xi}(\overline{G}, \mathbb{F}_2) \leq n/4 + 1$ must contain a triangle.

In the current work we prove a new upper bound on the minrank parameter over finite fields of *generalized Kneser graphs*. In these graphs the vertices are all the s-subsets of a universe $[d]$, where two sets are adjacent if their intersection size is smaller than some integer m. Note that for $m = 1$ we get the standard family of Kneser graphs (see Definition 2). In the proof we modify and extend an argument of [24], which is based on linear spaces of multivariate polynomials, building on a previous work of Alon [2]. For the precise statement, see Theorem 18. We turn to describe several applications of our bound.

As a first application, we establish an explicit construction of graphs that do not contain short odd cycles and yet have low minrank over every finite field.

▶ **Theorem 10.** *For every odd integer $\ell \geq 3$ there exists $\delta = \delta(\ell) > 0$ such that for every sufficiently large integer n, there exists an n-vertex graph G with no odd cycle of length at most ℓ such that for every finite field $\mathbb{F}$, $\mathrm{minrk}_{\mathbb{F}}(G) \leq n^{1-\delta}$.*

Theorem 10 immediately implies that the Odd Alternating Cycle Conjecture is false over every finite field, even for undirected graphs. This rules out the approach suggested in [11] for lower bounds on the rigidity of certain circulant matrices and thus falls into the recent line of non-rigidity results based on the polynomial method (see, e.g., [1, 15, 16]). We note, however, that the general upper bound of [16] on the rigidity of $n \times n$ circulant matrices does not apply to the setting of parameters considered in [11] (because in [16] the upper bound is $n^{1+\varepsilon}$ for a constant $\varepsilon > 0$, whereas in [11] the rigidity is only claimed to be $\Omega(n \cdot \log^{\varepsilon} n)$ for a constant $\varepsilon > 0$).

We next consider the behavior of the orthogonality dimension over the binary field of the complement of triangle-free graphs. It is relevant to mention here that in the proof of Theorem 10, the matrices that imply the stated bound on the minrank are symmetric (see Remark 20). For the binary field, this can be combined with a matrix decomposition result due to Lempel [30] to obtain the following theorem, which answers a question of [11] negatively.

▶ **Theorem 11.** *There exists a constant $\delta > 0$ such that for every sufficiently large integer n there exists a triangle-free n-vertex graph G such that $\overline{\xi}(\overline{G}, \mathbb{F}_2) \leq n^{1-\delta}$.*

The above result can also be stated in terms of nearly orthogonal systems. For a field $\mathbb{F}$, a system of vectors in $\mathbb{F}^m$ is said to be *nearly orthogonal* if every vector of the system is not self-orthogonal and any set of three of them contains an orthogonal pair. For the real field, it was proved by Rosenfeld [38] that every nearly orthogonal system in $\mathbb{R}^m$ has size at most $2m$. Theorem 11 shows that the situation is quite different over the binary field. Namely, it implies that there exists a constant $\delta > 0$ such that for infinitely many integers m there exists a nearly orthogonal system in $\mathbb{F}_2^m$ of size at least $m^{1+\delta}$.

We finally mention that our bound on the minrank parameter of generalized Kneser graphs can be used to obtain graphs with a constant *vector chromatic number* χ_{v} (see Definition 24) whose complement has a polynomially large minrank over every finite field.

▶ **Theorem 12.** *There exists a constant $\delta > 0$ such that for infinitely many integers n there exists an n-vertex graph G such that $\chi_{\mathrm{v}}(G) \leq 3$ and yet $\mathrm{minrk}_{\mathbb{F}}(\overline{G}) \geq n^{\delta}$ for every finite field $\mathbb{F}$.*

The interest in such graphs comes from the semidefinite programming algorithmic approach applied in [9] for approximating the minrank parameter. As explained in [22], such graphs imply a limitation on this approach, which is based on the constant vector chromatic number of the complement of the instances. Theorem 12 improves on [22, Theorem 1.3] where the bound on the minrank is shown only for sufficiently large finite fields.

1.2 Outline

The rest of the paper is organized as follows. In Section 2, we prove our bounds on the generalized orthogonality dimension parameters of Kneser graphs (Theorems 6 and 7) and derive our hardness result (Theorem 1). In Section 3, we prove our bound on the minrank parameter over finite fields of generalized Kneser graphs and deduce Theorems 10, 11, and 12.

2 The Generalized Orthogonality Dimension of Kneser Graphs

In this section we study the generalized orthogonality dimension parameters of Kneser graphs, namely, the quantities $\overline{\xi}_k(K(d,s))$ (recall Definitions 2 and 4), and prove Theorems 6 and 7. We start with a linear algebra lemma that will be useful in our proofs.

2.1 Linear Algebra Lemma

▶ **Lemma 13.** *Let U be a subspace of $\mathbb{R}^t$ with $\dim(U) = \ell$, let $\mathcal{W}$ be a finite collection of subspaces of $\mathbb{R}^t$, and let $\ell' \leq \ell$ be an integer satisfying $\dim(U \cap W) \leq \ell'$ for every $W \in \mathcal{W}$. Then, there exists a subspace U' of U with $\dim(U') = \ell - \ell'$ such that $\dim(U' \cap W) = 0$ for every $W \in \mathcal{W}$.*

Intuitively, given a subspace U and a collection $\mathcal{W}$ as in the lemma, a "random-like" subspace U' of U with dimension $\ell - \ell'$ is expected to have a trivial intersection with each of the subspaces of $\mathcal{W}$, and thus to satisfy the assertion of the lemma. The formal proof can be found in [19].

2.2 The case $s = 2$

We turn to prove Theorem 6 which determines the generalized orthogonality dimension parameters of Kneser graphs $K(d,s)$ for $s = 2$.

Proof of Theorem 6. Fix an integer $k \geq 1$. For the upper bound, recall that for all integers $d \geq 4$ we have

$$\overline{\xi}_k(K(d,2),\mathbb{R}) \leq \chi_k(K(d,2)) = \left\lceil \frac{k}{2} \right\rceil \cdot (d-4) + 2k.$$

For the lower bound, we consider the induced subgraph of $K(d,2)$, denoted by $K^{-}(d,2)$, obtained from $K(d,2)$ by removing one of its vertices, say, the vertex $\{1,2\}$. We turn to prove that for all integers $d \geq 4$ it holds that

$$\overline{\xi}_k(K^{-}(d,2),\mathbb{R}) \geq \left\lceil \frac{k}{2} \right\rceil \cdot (d-4) + 2k, \qquad (1)$$

which immediately implies the required lower bound on $\overline{\xi}_k(K(d,2),\mathbb{R})$ as well. To this end, we apply an induction on d. For $d=4$, the graph $K(d,2)$ is a perfect matching on 6 vertices, hence its subgraph $K^-(d,2)$ clearly contains an edge. Since every orthogonal k-subspace representation of this graph assigns to the vertices of this edge orthogonal k-subspaces it follows that $\overline{\xi}_k(K^-(4,2),\mathbb{R}) \geq 2k$, as desired. Now, fix some $d>4$. Assuming that (1) holds for $d-1$, we turn to prove it for d.

Recall that the vertex set V of $K^-(d,2)$ consists of all the 2-subsets of $[d]$ except $\{1,2\}$. Let $(U_A)_{A\in V}$ be a t-dimensional orthogonal k-subspace representation of $K^-(d,2)$. We proceed by considering the following two cases.

Assume first that there exists some $i \geq 4$ for which

$$\dim(U_{\{1,3\}} \cap U_{\{1,i\}}) \geq \left\lceil \frac{k}{2} \right\rceil. \tag{2}$$

In this case, consider the induced subgraph of $K^-(d,2)$ on the vertex set V' obtained from V by removing the vertex $\{3,i\}$ and all the vertices that include the element 1. Notice that this subgraph is isomorphic to $K^-(d-1,2)$ and that every vertex of V' is disjoint from either $\{1,3\}$ or from $\{1,i\}$ (or both). This implies that the restriction $(U_A)_{A\in V'}$ of the given assignment to the vertices of V' forms an orthogonal k-subspace representation of $K^-(d-1,2)$, all of whose subspaces lie in the subspace of $\mathbb{R}^t$ that is orthogonal to $U = U_{\{1,3\}} \cap U_{\{1,i\}}$. By applying an orthogonal linear transformation from this subspace to $\mathbb{R}^{t-\dim(U)}$, we obtain that

$$\overline{\xi}_k(K^-(d-1,2),\mathbb{R}) \leq t - \dim(U) \leq t - \left\lceil \frac{k}{2} \right\rceil,$$

where in the second inequality we have used (2). Using the induction hypothesis, this implies that

$$t \geq \overline{\xi}_k(K^-(d-1,2),\mathbb{R}) + \left\lceil \frac{k}{2} \right\rceil \geq \left\lceil \frac{k}{2} \right\rceil \cdot (d-5) + 2k + \left\lceil \frac{k}{2} \right\rceil = \left\lceil \frac{k}{2} \right\rceil \cdot (d-4) + 2k,$$

and we are done.

We are left with the case where for every $i \geq 4$ it holds that $\dim(U_{\{1,3\}} \cap U_{\{1,i\}}) \leq \left\lceil \frac{k}{2} \right\rceil - 1$. Apply Lemma 13 to the k-subspace $U_{\{1,3\}}$ and the collection $\{U_{\{1,i\}} \mid 4 \leq i \leq d\}$. It follows that there exists a subspace U of $U_{\{1,3\}}$ with $\dim(U) = k - (\lceil \frac{k}{2} \rceil - 1) \geq \lceil \frac{k}{2} \rceil$ such that for every $i \geq 4$ it holds that $\dim(U \cap U_{\{1,i\}}) = 0$. Consider the induced subgraph of $K^-(d,2)$ on the vertex set V' obtained from V by removing the vertex $\{2,3\}$ and all the vertices that include the element 1. As before, this subgraph is isomorphic to the graph $K^-(d-1,2)$.

We define an orthogonal k-subspace representation of this graph as follows. Let B be a set in V'. If $3 \notin B$ we define $\widetilde{U}_B = U_B$. Otherwise we have $B = \{3,i\}$ for some $i \geq 4$, and we let $\widetilde{U}_{\{3,i\}}$ be the projection of $U_{\{1,i\}}$ to the subspace of $\mathbb{R}^t$ that is orthogonal to U. Note that the fact that $\dim(U \cap U_{\{1,i\}}) = 0$ guarantees that $\dim(\widetilde{U}_{\{3,i\}}) = \dim(U_{\{1,i\}}) = k$.

To prove that the assignment $(\widetilde{U}_B)_{B\in V'}$ forms an orthogonal k-subspace representation of the graph, let B_1 and B_2 be disjoint sets in V'. If $3 \notin B_1 \cup B_2$ then we have $\widetilde{U}_{B_1} = U_{B_1}$ and $\widetilde{U}_{B_2} = U_{B_2}$, so it is clear that $\widetilde{U}_{B_1}$ and $\widetilde{U}_{B_2}$ are orthogonal. Otherwise, assume without loss of generality that $B_1 = \{3,i\}$ for some $i \geq 4$ and that $3 \notin B_2$. In this case we have $\widetilde{U}_{B_2} = U_{B_2}$, and since B_2 is disjoint from B_1 it is also disjoint from $\{1,i\}$ and from $\{1,3\}$, hence $\widetilde{U}_{B_2}$ is orthogonal to both $U_{\{1,i\}}$ and $U_{\{1,3\}}$ as well as to the projection $\widetilde{U}_{B_1}$ of $U_{\{1,i\}}$ to the subspace orthogonal to $U \subseteq U_{\{1,3\}}$. We get that $\widetilde{U}_{B_1}$ and $\widetilde{U}_{B_2}$ are orthogonal, as required.

Finally, observe that all the subspaces $\widetilde{U}_B$ lie in the subspace of $\mathbb{R}^t$ that is orthogonal to U. Indeed, for sets B with $3 \in B$ this follows from the definition of $\widetilde{U}_B$, and for the other sets this holds because they are disjoint from $\{1,3\}$. By applying an orthogonal linear transformation from this subspace to $\mathbb{R}^{t-\dim(U)}$, we obtain that $\overline{\xi}_k(K^-(d-1,2),\mathbb{R}) \leq t - \dim(U) \leq t - \left\lceil \frac{k}{2} \right\rceil$, and as in the previous case, by the induction hypothesis it follows that $t \geq \lceil \frac{k}{2} \rceil \cdot (d-4) + 2k$, completing the proof. ◀

2.3 General s

We now prove Theorem 7 which provides a lower bound on the generalized orthogonality dimension parameters of Kneser graphs $K(d,s)$ for $s \geq 3$.

Proof of Theorem 7. Fix integers $k \geq s \geq 3$ and denote $m = \lceil \frac{k+1}{s} \rceil$. Let $d_0 = d_0(s,k)$ be a sufficiently large integer to be determined later. We apply an induction on d. To do so, we define $c = c(s,k)$ to be sufficiently large, say, $c = \frac{k-m+1}{s-1} \cdot (d_0 + s - 2)$, so that the statement of the theorem trivially holds for all integers $d \leq d_0 + s - 2$, and turn to prove the statement for $d \geq d_0$ assuming that it holds for $d - (s-1)$.

Let $(U_A)_{A \in V}$ be a t-dimensional orthogonal k-subspace representation of $K(d,s)$. We start with some notation. For an s-subset A of $[d]$, an element $i \in A$, and an s-subset B of $[d]$ satisfying $A \cap B = \{i\}$, we let $\mathcal{G}_{A,i}(B)$ denote the collection that consists of the set B and all the sets obtained from B by replacing i with some element from $A \setminus \{i\}$. Note that $|\mathcal{G}_{A,i}(B)| = s$. We say that a vertex A of $K(d,s)$ is *good* (with respect to the given orthogonal subspace representation) if there exists an $i \in A$ such that for every vertex B satisfying $A \cap B = \{i\}$ it holds that $\dim(U_A \cap U_C) \leq m-1$ for some $C \in \mathcal{G}_{A,i}(B)$.

Assume first that there exists a good vertex A in $K(d,s)$ associated with an element $i \in A$. Applying Lemma 13, we get that there exists a $(k-m+1)$-subspace U of U_A such that for every vertex B satisfying $A \cap B = \{i\}$ it holds that $\dim(U \cap U_C) = 0$ for some $C \in \mathcal{G}_{A,i}(B)$. We define an orthogonal k-subspace representation of the graph $K(d-(s-1),s)$ on the ground set $[d] \setminus (A \setminus \{i\})$ as follows. Let B be an s-subset of $[d] \setminus (A \setminus \{i\})$. If $i \notin B$ we define $\widetilde{U}_B = U_B$. Otherwise, we have $A \cap B = \{i\}$, and we let $\widetilde{U}_B$ be the projection of U_C to the subspace of $\mathbb{R}^t$ orthogonal to U, where $C \in \mathcal{G}_{A,i}(B)$ is a set satisfying $\dim(U \cap U_C) = 0$. Note that this condition guarantees that $\dim(\widetilde{U}_B) = \dim(U_C) = k$.

We claim that the subspaces $\widetilde{U}_B$ form an orthogonal k-subspace representation of the graph $K(d-(s-1),s)$. To see this, let B_1 and B_2 be disjoint s-subsets of $[d] \setminus (A \setminus \{i\})$. If $i \notin B_1 \cup B_2$ then we have $\widetilde{U}_{B_1} = U_{B_1}$ and $\widetilde{U}_{B_2} = U_{B_2}$, so it is clear that $\widetilde{U}_{B_1}$ and $\widetilde{U}_{B_2}$ are orthogonal. Otherwise, assume without loss of generality that $i \in B_1$ and $i \notin B_2$. In this case, $\widetilde{U}_{B_2} = U_{B_2}$, and $\widetilde{U}_{B_1}$ is the projection of U_C to the subspace of $\mathbb{R}^t$ orthogonal to U for some $C \in \mathcal{G}_{A,i}(B_1)$. Since B_2 is disjoint from A, it follows that the subspace $\widetilde{U}_{B_2}$ is orthogonal to U_A as well as to its subspace U. It also follows that B_2 is disjoint from every set in $\mathcal{G}_{A,i}(B_1)$, hence the subspace $\widetilde{U}_{B_2}$ is orthogonal to U_C. We get that $\widetilde{U}_{B_2}$ is orthogonal to $\widetilde{U}_{B_1}$, as required.

Now, observe that the above orthogonal k-subspace representation of $K(d-(s-1),s)$ lies in the subspace of $\mathbb{R}^t$ that is orthogonal to the $(k-m+1)$-subspace U. Indeed, for sets B with $i \in B$ this follows from the definition of $\widetilde{U}_B$, and for the other sets this holds because they are disjoint from A. By applying an orthogonal linear transformation from this subspace to $\mathbb{R}^{t-\dim(U)}$, it follows that

$$\overline{\xi}_k(K(d-(s-1),s),\mathbb{R}) \leq t - \dim(U) = t - (k-m+1).$$

Using the induction hypothesis, this implies that

$$t \geq \frac{k-m+1}{s-1} \cdot (d-(s-1)) - c + (k-m+1) = \frac{k-m+1}{s-1} \cdot d - c,$$

and we are done.

We are left with the case where no vertex of $K(d,s)$ is good, for which we need the following lemma. Its proof can be found in [19].

▶ **Lemma 14.** *If a vertex A of $K(d,s)$ is not good then there exists a nonzero vector $u_A \in U_A$ such that the number of vertices D of $K(d,s)$ for which U_D is not orthogonal to u_A is at most $\binom{2s-1}{2} \cdot \binom{d-2}{s-2}$.*

We finally show how Lemma 14 completes the proof of the theorem. Assume that no vertex of $K(d,s)$ is good, and consider the following process: We start with the entire vertex set of $K(d,s)$, and in every iteration we choose an arbitrary vertex A associated with its nonzero vector $u_A \in U_A$ from Lemma 14 and eliminate all vertices whose subspaces are not orthogonal to u_A. The nonzero vectors associated with the chosen vertices are clearly pairwise orthogonal, and their number, just like the number of iterations in the process, is at least

$$\frac{\binom{d}{s}}{\binom{2s-1}{2} \cdot \binom{d-2}{s-2}} \geq \frac{k-m+1}{s-1} \cdot d - c,$$

where the inequality holds for every $d \geq d_0$ assuming that $d_0 = d_0(s,k)$ is sufficiently large (because the left-hand side of the inequality is quadratic in d whereas the right-hand side is linear in d). However, the size of the obtained orthogonal set cannot exceed the dimension t, hence

$$t \geq \frac{k-m+1}{s-1} \cdot d - c,$$

and we are done. ◀

As immediate corollaries of Theorem 7, we obtain the following.

▶ **Corollary 15.** *For every integers $s \geq 3$ and $\ell \geq 2$ there exists $c = c(s,\ell)$ such that for all integers $d \geq 2s$,*

$$\overline{\xi}_{\ell \cdot s-1}(K(d,s),\mathbb{R}) \geq \ell \cdot d - c.$$

As mentioned before, the bound given in Corollary 15 is tight up to the additive constant c.

▶ **Corollary 16.** *There exists a constant c such that for all integers $d \geq 6$, $\overline{\xi}_4(K(d,3),\mathbb{R}) \geq 3d/2 - c$.*

Equipped with Corollary 16, we are ready to deduce Theorem 1.

Proof of Theorem 1. Let t be a sufficiently large integer. Recall that a result of [6] implies that $\overline{\xi}_3(K(t,3),\mathbb{R}) = t$, whereas Corollary 16 implies that $\overline{\xi}_4(K(t,3),\mathbb{R}) \geq 3t/2 - c$ for an absolute constant c. Applying Proposition 5 with $F = K(t,3)$, it follows that it is NP-hard to decide whether an input graph G satisfies $\overline{\xi}(G,\mathbb{R}) \leq t$ or $\overline{\xi}(G,\mathbb{R}) \geq 3t/2 - c$, as desired. ◀

3 The Minrank of Generalized Kneser Graphs

In this section we consider a generalization of the family of Kneser graphs, defined as follows.

▶ **Definition 17** (Generalized Kneser Graphs). *For integers $m \leq s \leq d$, the* generalized Kneser graph *$K^{<}(d,s,m)$ is the graph whose vertices are all the s-subsets of $[d]$, where two sets A, B are adjacent if $|A \cap B| < m$.*

For this family of graphs, we prove the following upper bound on the minrank parameter over finite fields (recall Definition 8).

▶ **Theorem 18.** *For all integers $m \le s \le d$ and for every finite field $\mathbb{F}$,*

$$\mathrm{minrk}_{\mathbb{F}}(K^{<}(d,s,m)) \le \sum_{i=0}^{s-m} \binom{d}{i}.$$

Moreover, the bound on the minrank can be achieved by a symmetric matrix.

As in the previous section, we start with a simple linear algebra lemma, whose proof can be found in [19].

▶ **Lemma 19.** *For a graph G on the vertex set $[n]$, let $M \in \mathbb{Z}^{n\times n}$ be an integer matrix such that $M_{i,i} = 1$ for every $i \in [n]$, and $M_{i,j} = 0$ for every distinct non-adjacent vertices i and j in G. Then, for every finite field $\mathbb{F}$, $\mathrm{minrk}_{\mathbb{F}}(G) \le \mathrm{rank}_{\mathbb{R}}(M)$.*

Proof of Theorem 18. Consider the polynomial $q \in \mathbb{R}[x]$ defined by

$$q(x) = \binom{x-m}{s-m} = \frac{1}{(s-m)!} \cdot (x-m)(x-(m+1))\cdots(x-(s-1)).$$

Notice that q is an integer-valued polynomial of degree $s-m$. Let $f : \{0,1\}^d \times \{0,1\}^d \to \mathbb{R}$ be the function defined by

$$f(x,y) = q\Big(\sum_{i=1}^{d} x_i y_i\Big)$$

for every $x, y \in \{0,1\}^d$. Expanding f as a linear combination of monomials, the relation $z^2 = z$ for $z \in \{0,1\}$ implies that one can reduce to 1 the exponent of each variable occuring in a monomial. It follows that f can be represented as a multilinear polynomial in the $2d$ variables of x and y. By combining terms involving the same monomial in the variables of x, one can write f as

$$f(x,y) = \sum_{i=1}^{R} g_i(x)h_i(y)$$

for an integer R and functions $g_i, h_i : \{0,1\}^d \to \mathbb{R}$, $i \in [R]$, such that the g_i's are distinct multilinear monomials of total degree at most $s-m$ in d variables. It follows that $R \le \sum_{i=0}^{s-m} \binom{d}{i}$.

Now, let M_1 and M_2 be the $2^d \times R$ matrices whose rows are indexed by $\{0,1\}^d$ and whose columns are indexed by $[R]$, defined by $(M_1)_{x,i} = g_i(x)$ and $(M_2)_{x,i} = h_i(x)$. Then, the rank over $\mathbb{R}$ of the matrix $M = M_1 \cdot M_2^T$ is at most R and for every $x, y \in \{0,1\}^d$ it holds that $M_{x,y} = f(x,y)$. By the definition of f the matrix M is symmetric, and since q is an integer-valued polynomial, all of its entries are integer.

Finally, let V be the vertex set of $K^{<}(d,s,m)$, that is, the collection of all s-subsets of $[d]$, and identify every vertex $A \in V$ with an indicator vector $c_A \in \{0,1\}^d$ in the natural way. Observe that for every $A, B \in V$ we have

$$M_{c_A,c_B} = f(c_A, c_B) = q(|A \cap B|).$$

Hence, for every $A \in V$ we have $|A| = s$ and thus $M_{c_A,c_A} = q(s) = 1$, whereas for every distinct non-adjacent $A, B \in V$ we have $m \le |A \cap B| \le s-1$ and thus $M_{c_A,c_B} = q(|A \cap B|) = 0$. Since the restriction of M to $V \times V$ is symmetric and has rank at most R over the reals, Lemma 19 implies that $\mathrm{minrk}_{\mathbb{F}}(K^{<}(d,s,m)) \le R$ for every finite field $\mathbb{F}$ and that the bound can be achieved by a symmetric matrix, as desired. ◀

▶ Remark 20. Theorem 18 guarantees that the bound on the minrank can be achieved by a symmetric matrix. This will be crucial for one of our applications, namely, for a construction of triangle-free graphs whose complement has low orthogonality dimension over the binary field $\mathbb{F}_2$ (see Section 3.1.2). We remark, however, that for undirected graphs and for fields of characteristic different from 2, attaining the bound on the minrank by a symmetric matrix can be achieved easily with a factor of 2 worse bound on the minrank. Indeed, if a matrix M represents a graph G over a field $\mathbb{F}$ of characteristic different from 2 and satisfies $\text{rank}_{\mathbb{F}}(M) = r$ then the matrix $M + M^T$ also represents G and has rank at most $2r$ over $\mathbb{F}$. This argument does not hold over fields of characteristic 2, since in this case the diagonal entries of $M + M^T$ are all zeros.

3.1 Applications

We gather below several applications of Theorem 18.

3.1.1 The Odd Alternating Cycle Conjecture over Finite Fields

We turn to disprove Conjecture 9 over every finite field. We will use the simple fact that generalized Kneser graphs do not contain short odd cycles, as stated below (see, e.g., [13, 24]).

▶ **Lemma 21.** *Let $\ell \geq 3$ be an odd integer. For every even integer d and an integer $m \leq \frac{d}{2\ell}$, the graph $K^{<}(d, \frac{d}{2}, m)$ contains no odd cycle of length at most ℓ.*

We prove the following theorem, confirming Theorem 10.

▶ **Theorem 22.** *For every odd integer $\ell \geq 3$ there exists $\delta = \delta(\ell) > 0$ such that for every sufficiently large integer n, there exists an n-vertex graph G with no odd cycle of length at most ℓ such that for every finite field $\mathbb{F}$,*

$$\text{minrk}_{\mathbb{F}}(G) \leq n^{1-\delta}.$$

Moreover, the bound on the minrank can be achieved by a symmetric matrix.

Proof. Fix an odd integer $\ell \geq 3$. For an integer d divisible by 2ℓ, consider the graph $G = K^{<}(d, \frac{d}{2}, m)$ where $m = \frac{d}{2\ell}$. By Lemma 21, G contains no odd cycle of length at most ℓ. As for the minrank parameter, Theorem 18 implies that for every finite field $\mathbb{F}$,

$$\text{minrk}_{\mathbb{F}}(G) \leq \sum_{i=0}^{d/2-m} \binom{d}{i} \leq 2^{H(\frac{1}{2}-\frac{m}{d})\cdot d} = 2^{H(\frac{1}{2}-\frac{1}{2\ell})\cdot d},$$

where H stands for the binary entropy function. Since G has $|V| = \binom{d}{d/2} = 2^{(1-o(1))\cdot d}$ vertices, for any $\delta > 0$ such that $H(\frac{1}{2} - \frac{1}{2\ell}) < 1 - \delta$ we have $\text{minrk}_{\mathbb{F}}(G) \leq |V|^{1-\delta}$ for every sufficiently large integer d. The proof is completed by considering, for every sufficiently large integer n, some n-vertex subgraph of the graph defined above, where d is the smallest integer divisible by 2ℓ such that $n \leq \binom{d}{d/2}$. ◀

3.1.2 Triangle-free Graphs and the Orthogonality Dimension over the Binary Field

We turn to prove Theorem 11. Its proof adopts the following special case of a result due to Lempel [30].

▶ **Lemma 23** ([30]). *Let M by an n by n symmetric matrix over the binary field $\mathbb{F}_2$ with at least one nonzero diagonal entry and rank r. Then, there exists an n by r matrix B over $\mathbb{F}_2$ satisfying $M = B \cdot B^T$.*

Proof of Theorem 11. Apply Theorem 22 with $\ell = 3$ to obtain some $\delta > 0$ such that for every sufficiently large integer n, there exist a triangle-free n-vertex graph G and an n by n symmetric matrix M over $\mathbb{F}_2$ of rank $r = \text{rank}_{\mathbb{F}_2}(M) \leq n^{1-\delta}$ that represents G. By Lemma 23, there exists an n by r matrix B over $\mathbb{F}_2$ satisfying $M = B \cdot B^T$. By assigning the ith row of B to the ith vertex of G we get an r-dimensional orthogonal representation of $\overline{G}$ over $\mathbb{F}_2$, hence $\overline{\xi}(\overline{G}, \mathbb{F}_2) \leq r \leq n^{1-\delta}$. ◀

3.1.3 The Vector Chromatic Number vs. Minrank

The vector chromatic number of graphs, introduced by Karger, Motwani, and Sudan in [27], is defined as follows.

▶ **Definition 24** (Vector Chromatic Number). *For a graph $G = (V, E)$ the* vector chromatic number *of G, denoted by $\chi_v(G)$, is the minimal real value of $\kappa > 1$ such that there exists an assignment of a unit vector w_v to every vertex $v \in V$ satisfying the inequality $\langle w_v, w_{v'} \rangle \leq -\frac{1}{\kappa-1}$ whenever v and v' are adjacent in G.*

To prove Theorem 12, we need the following simple fact that relates the minrank of a graph to the minrank of its complement (see, e.g., [36, Remark 2.2]).

▶ **Fact 25.** *For every field $\mathbb{F}$ and an n-vertex graph G, $\text{minrk}_{\mathbb{F}}(G) \cdot \text{minrk}_{\mathbb{F}}(\overline{G}) \geq n$.*

Proof of Theorem 12. For an integer d divisible by 8, consider the graph $G = K^<(d, \frac{d}{2}, m)$ where $m = \frac{d}{8}$. We first claim that $\chi_v(G) \leq 3$. To see this, assign to every vertex A of G, representing a $\frac{d}{2}$-subset of $[d]$, the unit vector $w_A \in \mathbb{R}^d$ defined by $(w_A)_i = \frac{1}{\sqrt{d}}$ if $i \in A$ and $(w_A)_i = -\frac{1}{\sqrt{d}}$ otherwise. Observe that every two distinct vertices A and B that are adjacent in G satisfy $|A \cap B| < \frac{d}{8}$ and thus $|A \triangle B| > \frac{3d}{4}$, implying that $\langle w_A, w_B \rangle = \frac{d-2\cdot|A \triangle B|}{d} < -\frac{1}{2}$. This implies that $\chi_v(G) \leq 3$, as claimed. As for the minrank parameter, Theorem 18 implies that for every finite field $\mathbb{F}$,

$$\text{minrk}_{\mathbb{F}}(G) \leq \sum_{i=0}^{d/2-m} \binom{d}{i} \leq 2^{H(\frac{1}{2}-\frac{m}{d})\cdot d} = 2^{H(3/8)\cdot d},$$

where H stands for the binary entropy function. Since G has $n = \binom{d}{d/2} = 2^{(1-o(1))\cdot d}$ vertices, for any $\delta < 1 - H(3/8)$ we have $\text{minrk}_{\mathbb{F}}(G) \leq n^{1-\delta}$ assuming that d is sufficiently large. By Fact 25, this implies that $\text{minrk}_{\mathbb{F}}(\overline{G}) \geq n^{\delta}$, and we are done. ◀

References

1 Josh Alman and R. Ryan Williams. Probabilistic rank and matrix rigidity. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC'17)*, pages 641–652, 2017.

2 Noga Alon. The Shannon capacity of a union. *Combinatorica*, 18(3):301–310, 1998.

3 Noga Alon and Nabil Kahale. Approximating the independence number via the ϑ-function. *Math. Program.*, 80:253–264, 1998.

4 Jop Briët, Harry Buhrman, Debbie Leung, Teresa Piovesan, and Florian Speelman. Round elimination in exact communication complexity. In *Proceedings of the 10th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC'15)*, volume 44 of *LIPIcs*, pages 206–225, 2015.

5 Jop Briët and Jeroen Zuiddam. On the orthogonal rank of Cayley graphs and impossibility of quantum round elimination. *Quantum Information & Computation*, 17(1&2):106–116, 2017.

6 Boris Bukh and Christopher Cox. On a fractional version of Haemers' bound. *IEEE Trans. Inform. Theory*, 65(6):3340–3348, 2019.

7 Jakub Bulín, Andrei A. Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC'19)*, pages 602–613, 2019.

8 Peter J. Cameron, Ashley Montanaro, Michael W. Newman, Simone Severini, and Andreas J. Winter. On the quantum chromatic number of a graph. *Electr. J. Comb.*, 14(1), 2007.

9 Eden Chlamtáč and Ishay Haviv. Linear index coding via semidefinite programming. *Combinatorics, Probability & Computing*, 23(2):223–247, 2014. Preliminary version in SODA'12.

10 Vasek Chvátal, Michael R. Garey, and David S. Johnson. Two results concerning multicoloring. *Annals of Discrete Math.*, 2:151–154, 1978.

11 Bruno Codenotti, Pavel Pudlák, and Giovanni Resta. Some structural properties of low-rank matrices related to computational complexity. *Theor. Comput. Sci.*, 235(1):89–107, 2000. Preliminary version in ECCC'97.

12 Ronald de Wolf. Quantum Computing and Communication Complexity. PhD thesis, Universiteit van Amsterdam, 2001.

13 Tristan Denley. The odd girth of the generalised Kneser graph. *Eur. J. Comb.*, 18(6):607–611, 1997.

14 Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM J. Comput.*, 39(3):843–873, 2009. Preliminary version in STOC'06.

15 Zeev Dvir and Benjamin L. Edelman. Matrix rigidity and the Croot-Lev-Pach lemma. *Theory of Computing*, 15(1):1–7, 2019.

16 Zeev Dvir and Allen Liu. Fourier and circulant matrices are not rigid. In *34th Computational Complexity Conference (CCC'19)*, pages 17:1–17:23, 2019.

17 Uriel Feige. Randomized graph products, chromatic numbers, and the Lovász ϑ-function. *Combinatorica*, 17(1):79–90, 1997. Preliminary version in STOC'95.

18 Michael R. Garey and David S. Johnson. The complexity of near-optimal graph coloring. *J. ACM*, 23(1):43–49, 1976.

19 Alexander Golovnev and Ishay Haviv. The (generalized) orthogonality dimension of (generalized) Kneser graphs: Bounds and applications. *arXiv*, 2020. arXiv:2002.08580.

20 Alexander Golovnev, Oded Regev, and Omri Weinstein. The minrank of random graphs. *IEEE Trans. Inform. Theory*, 64(11):6990–6995, 2018. Preliminary version in RANDOM'17.

21 Willem H. Haemers. On some problems of Lovász concerning the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(2):231–232, 1979.

22 Ishay Haviv. On minrank and the Lovász theta function. In *International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'18)*, pages 13:1–13:15, 2018.

23 Ishay Haviv. Approximating the orthogonality dimension of graphs and hypergraphs. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS'19)*, pages 39:1–39:15, 2019.

24 Ishay Haviv. On minrank and forbidden subgraphs. *ACM Transactions on Computation Theory (TOCT)*, 11(4):20, 2019. Preliminary version in RANDOM'18.

25 Ishay Haviv. Topological bounds on the dimension of orthogonal representations of graphs. *Eur. J. Comb.*, 81:84–97, 2019.

26 Sihuang Hu, Itzhak Tamo, and Ofer Shayevitz. A bound on the Shannon capacity via a linear programming variation. *SIAM J. Discrete Math.*, 32(3):2229–2241, 2018. Preliminary version in ISIT'17.

27 David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998. Preliminary version in FOCS'94.

28 Donald E. Knuth. The sandwich theorem. *Electr. J. Comb.*, 1(A1):1–48, 1994.

29 Michael Langberg and Alexander Sprintson. On the hardness of approximating the network coding capacity. *IEEE Trans. Inform. Theory*, 57(2):1008–1014, 2011. Preliminary version in ISIT'08.

30 Abraham Lempel. Matrix factorization over $GF(2)$ and trace-orthogonal bases of $GF(2^n)$. *SIAM J. Comput.*, 4(2):175–186, 1975.

31 László Lovász. Kneser's conjecture, chromatic number, and homotopy. *J. Comb. Theory, Ser. A*, 25(3):319–324, 1978.

32 László Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(1):1–7, 1979.

33 László Lovász. *Graphs and Geometry*, volume 65. Colloquium Publications, 2019.

34 László Lovász, Michael Saks, and Alexander Schrijver. Orthogonal representations and connectivity of graphs. *Linear Algebra and its Applications*, 114/115:439–454, 1989. Special Issue Dedicated to Alan J. Hoffman.

35 Laura Mančinska and David E Roberson. Quantum homomorphisms. *Journal of Combinatorial Theory, Series B*, 118:228–267, 2016.

36 René Peeters. Orthogonal representations over finite fields and the chromatic number of graphs. *Combinatorica*, 16(3):417–431, 1996.

37 Søren Riis. Information flows, graphs and their guessing numbers. *Electr. J. Comb.*, 14(1), 2007.

38 Moshe Rosenfeld. Almost orthogonal lines in E^d. *DIMACS Series in Discrete Math.*, 4:489–492, 1991.

39 Giannicola Scarpa and Simone Severini. Kochen-Specker sets and the rank-1 quantum chromatic number. *IEEE Trans. Inform. Theory*, 58(4):2524–2529, 2012.

40 Saul Stahl. n-tuple colorings and associated graphs. *J. Comb. Theory, Ser. B*, 20(2):185–203, 1976.

41 Saul Stahl. The multichromatic numbers of some Kneser graphs. *Discrete Mathematics*, 185(1-3):287–291, 1998.

42 Claude Tardif and Xuding Zhu. A note on Hedetniemi's conjecture, Stahl's conjecture and the Poljak-Rödl function. *Electr. J. Comb.*, 26(4):P4.32, 2019.

43 Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *6th International Symposium on Mathematical Foundations of Computer Science (MFCS'77)*, pages 162–176, 1977.

44 Marcin Wrochna and Stanislav Živný. Improved hardness for H-colourings of G-colourable graphs. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1426–1435, 2020.

Shadows of Newton Polytopes

Pavel Hrubeš
Institute of Mathematics, The Czech Academy of Sciences, Prague, Czech Republic

Amir Yehudayoff
Department of Mathematics, Technion-IIT, Haifa, Israel

Abstract

We define the shadow complexity of a polytope P as the maximum number of vertices in a linear projection of P to the plane. We describe connections to algebraic complexity and to parametrized optimization. We also provide several basic examples and constructions, and develop tools for bounding shadow complexity.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Newton polytope, Monotone arithmetic circuit

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.9

Funding *Pavel Hrubeš*: Supported by the GACR grant 19-27871X.

Acknowledgements We thank Michael Forbes for pointing out the connection between shadow complexity and Conjecture 47.

1 Introduction

A *polytope* is the convex hull of a finite set of points in Euclidean space. Equivalently, it is a compact set that is defined by finitely many linear inequalities. Polytopes are central in convex geometry and linear optimization algorithms.

Our goal is to understand

how many vertices can a shadow of a polytope have?

A shadow of a polytope $P \subseteq \mathbb{R}^n$ is a set of the form $L(P)$, where $L : \mathbb{R}^n \to \mathbb{R}^2$ is a linear map. The shadows of P are two-dimensional polygons, and hence typically much simpler than P. The *shadow complexity* of P is

$$\sigma(P) = \max_L |\mathsf{vert}(L(P))|,$$

where L is a linear map and $\mathsf{vert}(Q)$ is the vertex set of the polytope Q.

The shadow problem is interesting already in three-dimensional space. Moser's shadow problem asks about the shadow complexity of three-dimensional polytopes [35]. Specifically, the question is what is the minimum of $\sigma(P)$ over all three dimensional polytopes P with n vertices. The solution is $\Theta(\log n/\log\log n)$; see [9, 33]. In other words, every n-vertex polytope in $\mathbb{R}^3$ has a projection to $\mathbb{R}^2$ with at least $\Omega(\log n/\log\log n)$ vertices, and there are polytopes where this is tight. The latter is quite surprising; in such a polytope, most vertices must disappear when projected to the plane.

Our main motivation comes from algebraic complexity theory. This is the study of computations of polynomials over a field. The connection between between polynomials and polytopes is via the notion of *Newton polytope*. Let $\mathbb{F}$ be a field. For a list of variables $x = (x_1, \ldots, x_n)$ and $\alpha \in \mathbb{N}^n$, let x^α be the monomial $\prod_{i=1}^n x_i^{\alpha_i}$. A polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is a formal sum of the form $\sum_{\alpha\in\mathbb{N}^n} a_\alpha x^\alpha$ where $\mathsf{supp}(f) := \{\alpha \in \mathbb{N}^n : a_\alpha \neq 0\}$ is finite. The *Newton* polytope of f is

$$\mathsf{Newt}(f) := \mathsf{conv}(\mathsf{supp}(f)),$$

where $\mathsf{conv}(\cdot)$ denotes the convex hull.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 9; pp. 9:1–9:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Koiran et al. [30] made a bold conjecture relating the complexity of $\mathsf{Newt}(f)$ with the computational complexity of f. The *τ-conjecture for Newton polygons* asserts, roughly speaking, that if a *bi-variate* polynomial f is easy to compute then $\mathsf{Newt}(f)$ has a small number of vertices. This conjecture has serious consequences. It implies that the permanent polynomial requires arithmetic circuit of exponential size. This is a central and long-standing open problem in algebraic complexity.

The Newton polytope of the permanent polynomial is the the *Birkhoff* polytope $\mathsf{DS}_n \subseteq \mathbb{R}^{n\times n}$; namely, the set of $n \times n$ doubly stochastic matrices. The vertices of DS_n are all $n \times n$ permutation matrices. This perspective leads us to the following question.

▶ **Problem 1.** *What is $\sigma(\mathsf{DS}_n)$?*

The Birkhoff polytope has the curious property that it is *both* the Newton polytope of the determinant *and* of the permanent polynomial. This creates friction in the context of the τ-conjecture. Determinant is easy to compute whereas permanent is largely believed to be hard. More specifically, it can be shown that the τ-conjecture implies $\sigma(\mathsf{DS}_n) \leq 2^{O(\sqrt{n}\log^2 n)}$. Proving that $\sigma(\mathsf{DS}_n) = 2^{\Omega(n)}$ refutes this τ-conjecture.[1]

Any non-trivial connection between the arithmetic complexity of f and some geometric complexity measure of $\mathsf{Newt}(f)$, such as shadow complexity, will be an exciting development.

We exhibit such a connection in the case of *monotone* computations. A *monotone* arithmetic circuit uses the operations $+, \times$ and only non-negative numbers so that no cancellations can occur in the course of a computation (for definitions see Section 5). They have been considered in the seminal papers of Valiant [45] and of Jerrum and Snir [23], and many others including a less known line of research by Kasim-Zade, Kuznetsev, Gashkov and Sergeev [26, 32, 17, 18]. We show that shadow complexity allows to prove hardness results for monotone computation.

▶ **Theorem 1.** *Every monotone formula computing f contains at least $\sigma(\mathsf{Newt}(f))$ leaves.*

What we are really interested in is understanding algebraic circuits, not formulas. We show that in some cases shadow complexity allows to lower bound monotone *circuit* complexity. A polynomial f is *transparent* if $|\mathsf{supp}(f)| = \sigma(\mathsf{Newt}(f))$. In other words, there is a linear map $L : \mathbb{R}^n \to \mathbb{R}^2$ which maps $\mathsf{supp}(f)$ to distinct convexly independent points in $\mathbb{R}^2$.

▶ **Theorem 2.** *If f is transparent then every monotone circuit computing f has size at least $\Omega(\sigma(\mathsf{Newt}(f)))$.*

Theorem 2 can be used to explicitly find a monotone multilinear polynomial in n variables which requires an arithmetic circuit of size $\Omega(2^{n/3})$; see Corollary 40. This surpasses the classical bounds from [45, 23] which are of the form $2^{\Omega(n^{1/2})}$, and matches the bounds from [26, 18] and [39] up to the constant in the exponent. The combinatorial essence of our argument resembles the arguments of Gashkov and Sergeev [18].

▶ Remark 3. The transparency assumption is unavoidable. There exists a bivariate polynomial f with a monotone circuit of size $O(n)$ such that $\mathsf{Newt}(f)$ has 2^n vertices (see Theorem 30).

Shadow complexity has an algorithmic perspective as well. A polytope naturally defines a linear optimization problem $\Phi(w) = \max_{x\in P} \langle x, w\rangle$, where $\langle x, w\rangle$ is the standard inner product. The maximizers of this optimization problem are vertices of P. The Birkhoff

[1] This observation came from Michael Forbes in a private conversation.

polytope, e.g., corresponds to the maximum weight bipartite perfect matching problem. Some additional examples of linear optimization problems include the shortest path problem or the maximum cut problem.

In parametrized complexity, one considers weights that come from a one dimensional space $w(t) = w_0 + tw_1$ parametrized by $t \in \mathbb{R}$. The map $t \mapsto \Phi(w(t))$ is a convex and piecewise linear function. A natural complexity measure for such a map is the number $\beta(P, w(t))$ of the breakpoints in $\Phi(w(t))$. The *parametrized complexity* of P now becomes

$$\beta(P) = \max_{w_0,w_1} \beta(P, w(t)).$$

The quantity $\beta(P)$ has been studied by Carstensen [7, 8], Mulmuley and Shah [36, 37], and many others. Carstensen [8] and later [37] showed that the shortest path problem in an n-vertex graph can have $2^{\Omega(\log^2 n)}$ breakpoints, and that the maximum cut problem can have $2^{\Omega(n)}$ breakpoints. In Section 3.4, we give an example of a polytope that corresponds to a linear optimization problem on n variables with $2^{\Omega(n)}$ breakpoints; the previous constructions gave only $2^{\Omega(\sqrt{n})}$ breakpoints.

We observe a fundamental connection between shadow complexity and parameterized complexity.

▶ **Theorem 4.** *If* $|\mathsf{vert}(P)| > 1$ *then* $\frac{\sigma(P)}{2} \leq \beta(P) \leq \sigma(P) - 1$.

This means that results from parametrized complexity translate to the language of shadows, and vice versa. Carstensen's lower bound for example implies that

$$\sigma(\mathsf{DS}_n) \geq 2^{\Omega(\log^2 n)}.$$

This is the best lower bound on $\sigma(\mathsf{DS}_n)$ we are aware of. The best upper bound we know is $\sigma(\mathsf{DS}_n) \leq 2^{O(n)}$. This is not entirely obvious and we shall explain this later on (see Proposition 23).

The connection between shadow and parametrized complexities leads to interesting conclusions. The idea, in a nutshell, is that if optimization over P is easy then $\beta(P)$ is low. For example, if we can optimize over P by a greedy algorithm then $\beta(P)$ is at most quadratic. We do not want to dive into the theory of greedy algorithms, or a formal definition for that matter. Edmonds and Rado [12, 15] proved that if $R \subseteq \{0,1\}^n$ is a matroid then the optimization problem over R can be solved by a greedy algorithm. Many generalizations of this theorem have been considered (see [46] and references within).

For our purposes, the following simple definition is sufficient. Let $P \subseteq \mathbb{R}^n$ be a polytope and $w \in \mathbb{R}^n$. We denote by $\mathrm{Opt}_P(w)$ the set of vertices v of P such that $\langle v, w\rangle = \max_{x \in P} \langle x, w\rangle$. Given $w, w' \in \mathbb{R}^n$, we say that they are *order-equivalent* if for every $i, j \in [n]$, we have $w_i \leq w_j$ iff $w'_i \leq w'_j$. The polytope P is *greedy-like*, if for every order-equivalent w and w', we have $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. In other words, P is greedy-like if for every weight function w, where the maximum for w is achieved on P depends only on the order induced by w.

▶ **Lemma 5.** *If* $P \subseteq \mathbb{R}^n$ *is a greedy-like polytope then* $\beta(P) \leq \binom{n}{2}$ *and* $\sigma(P) \leq n(n-1)$.

A more general link was established by Mulmuley [36]. He considers a model of computation called *PRAM model without bit operations* intended to solve decision problems or optimization problems. This model allows to use basic arithmetic operations such as $+, \times$ as well as $=, \leq$, but does not allow access to the individual bits of the inputs. Mulmuley showed[2] that a fast parallel algorithm for optimizing over P gives a small $\beta(P)$. This leads to several interesting lower bounds in this model.

[2] There is a technical issue of bit-lengths which we avoid.

The above can be further linked to our discussion concerning monotone arithmetic circuits. A monotone arithmetic formula can be interpreted as a computation over the semiring $(\mathbb{R}, \min, +, \infty, 0)$ which solves the optimization problem over $\mathsf{Newt}(f)$; see Section 5.1 for more details. This a particular instance of the PRAM model.

Are there general non trivial bounds on shadow complexity? Let $M_\sigma(n)$ be the maximum $\sigma(P)$ over all polytopes $P \subseteq \mathbb{R}^n$ with vertices in $\{0,1\}^n$. In [31], Kortenkamp et al. have shown the following:

▶ **Proposition 6** ([31]). *There exist constants $0 < c_1 < c_2 < 1$ such that for every n sufficiently large $2^{c_1 n} \leq M_\sigma(n) \leq 2^{c_2 n}$.*

An explicit construction yields $c_1 \geq 1/3$; see Remark 20.

1.1 Why the plane?

Why do we study projections of polytopes to two dimensions?

First, our results rely on the fact that in two dimensions Minkowski sum (defined in Section 2.3) is well-behaved with respect to the number of vertices. In $\mathbb{R}^2$, we have $|\mathsf{vert}(P+Q)| \leq |\mathsf{vert}(P)| + |\mathsf{vert}(Q)|$. Already in $\mathbb{R}^3$, only the trivial upper bound $|\mathsf{vert}(P+Q)| \leq |\mathsf{vert}(P)| \cdot |\mathsf{vert}(Q)|$ holds.

Second, there exists a polytope in $\mathbb{R}^3$ with k vertices such that every projection to $\mathbb{R}^2$ has only $O(\log k/\log\log k)$ vertices. Hence it may happen that a polytope in $\mathbb{R}^n$ has exponentially many vertices when projected to $\mathbb{R}^3$ but only polynomially many when projected to $\mathbb{R}^2$.

That said, there are non-trivial upper bounds on the number of vertices of $P_1 + \cdots + P_r$ in $\mathbb{R}^d$ if r is large. For the sake of simplicity, we discuss the case of $d = 3$. It follows from a result of Gritzman and Sturmfels [19] that, given polytopes $P_1, \ldots, P_r$ with $n_1, \ldots, n_r$ vertices in $\mathbb{R}^3$,

$$|\mathsf{vert}(P_1 + \cdots + P_r)| \leq O((n_1 + \cdots + n_r)^2)\,.$$

This beats the trivial bound $n_1 n_2 n_3$ already for $r = 3$. The improved bound could be used to derive non-trivial bounds on monotone computations of a bounded depth (see Remark 51).

1.2 Extension complexity

As a final remark, we briefly discuss a different possible connection between polytopes and algebraic complexity. The *extension complexity of* P, denoted $\mathrm{xc}(P)$, as the smallest r such that P is a linear projection of a polytope $Q \subseteq \mathbb{R}^m$ where Q can be defined using r inequalities and an arbitrary number of equalities; see [47, 40, 13] and references within. It is related to communication complexity and algorithms (see, e.g., [38]).

We observe that, like shadow complexity, extension complexity also allows to prove lower bounds on monotone computation. Namely, if f has monotone formula of size s then $\mathrm{xc}(\mathsf{Newt}(f)) \leq O(s)$. This uses simple properties of extension complexity together with a result of Balas [2].

Extension complexity, however, can not yield general lower bounds in the non-monotone setting. There exists a polynomial with a polynomial size arithmetic circuit, but whose Newton polytope has an exponential extension complexity. See Section 5.4 for more details.

2 Tools

We start by presenting several tools for bounding shadow complexity, including some elementary facts about Newton polytopes.

2.1 Parametrized complexity

Some of the bounds on shadow complexity we describe come from the algorithmic viewpoint. So, we first prove the connection between shadow complexity and parametrized complexity.

Proof of Theorem 4. It is convenient to argue about

$$B^*(P, w(t)) := \beta(P, w(t)) + 1,$$

which counts to the number of *pieces* of $\Phi(w(t))$. Given $w(t) = w_0 + tw_1$, define $L : \mathbb{R}^n \to \mathbb{R}^2$ by

$$L(x) = (\langle w_0, x\rangle, \langle w_1, x\rangle).$$

Because $\langle w(t), x\rangle = \langle (1,t), L(x)\rangle$, we see that

$$\max_{x\in P} \langle x, w(t)\rangle = \max_{y\in L(P)} \langle y, (1,t)\rangle \ .$$

Since the maximum is always achieved at a vertex of $L(P)$, we obtain $B^*(P, w(t)) \le \sigma(P)$.

To prove the other inequality, we first show that $B^*(Q) \ge k/2 + 1$ for every polytope Q in $\mathbb{R}^2$ with $k \ge 2$ vertices. Take non-parallel $w_0, w_1 \in \mathbb{R}^2$ so that $\langle v, w_1\rangle$ are distinct for distinct vertices v of Q. Let $w(t) = w_0 + tw_1$ and $\bar{w}(t) = -w_0 + tw_1$. Each vertex v of Q can be separated from the other vertices by a hyperplane (in two dimensions, a line), and a small perturbation of the hyperplane is still separating. Hence, there exists a non-empty open interval I such that either $\max_{x\in Q} \langle x, w(t)\rangle$ or $\max_{x\in Q} \langle x, \bar{w}(t)\rangle$ is achieved at $x = v$ on $t \in I$. (And v is the only such vertex.) Let v_1 be the vertex for which $\langle x, w_1\rangle$ is the largest, and v_2 the one where it is smallest. When $t \to \infty$, both $\max_{x\in Q} \langle x, w(t)\rangle$ and $\max_{x\in Q} \langle x, \bar{w}(t)\rangle$ is achieved at v_1; similarly for v_2 and $t \to -\infty$. It follows that $B^*(Q, w(t)) + B^*(Q, \bar{w}(t)) \ge k+2$ and so $B^*(Q) \ge k/2 + 1$.

Now, given $P \subseteq \mathbb{R}^n$, let $L : \mathbb{R}^n \to \mathbb{R}^2$ be a linear map so that $L(P)$ has $\sigma(P)$ vertices. By the above, there exists $w(t)$ in $\mathbb{R}^2$ so that $\max_{x\in L(P)} \langle x, w(t)\rangle$ has at least $\sigma(P)/2$ breakpoints. Maximizing $\langle x, w(t)\rangle$ on $L(P)$ is equivalent to maximizing some $w'(t)$ on P and so $\beta(P) \ge \sigma(P)/2$. ◀

2.2 Greedy polytopes

Our goal here is to prove that $\sigma(P)$ is small whenever P is greedy-like (Lemma 5).

Proof of Lemma 5. Let $w(t)$ be a line in $\mathbb{R}^n$. For a given t, the weight vector $w(t)$ defines a preorder on $[n]$ by $i \le_t j$ iff $w(t)_i \le w(t)_j$. Since P is greedy-like, every breakpoint of $\Phi(w(t)) = \max_{x\in P} \langle x, w(t)\rangle$ occurs at a time where the order $\le_t$ changes. Hence there exist $i \ne j$ such that the linear function $w(t)_i - w(t)_j$ changes sign. There are $\binom{n}{2}$ pairs, and a linear function can change sign at most once. So, $\Phi(w(t))$ has at most $\binom{n}{2}$ breakpoints. This means $\beta(P) \le \binom{n}{2}$ and $\sigma(P) \le n(n-1)$. ◀

We further show that the definition of greedy-like can be relaxed to weights for which the maximum is achieved at a unique vertex. This weaker notion can be easier to verify, as in the case of Kruskal's algorithm mentioned in Proposition 17.

▶ **Lemma 7.** *Let $P \subseteq \mathbb{R}^n$ be a polytope. Assume that for every order-equivalent $w, w' \in \mathbb{R}^n$, the equality $Opt_P(w) = Opt_P(w')$ holds whenever $|Opt_P(w)| = 1$. Then P is greedy-like.*

Proof. Let P be as in the assumption. Assume that $w, w' \in \mathbb{R}^n$ are order-equivalent with $|\mathrm{Opt}_P(w)| \geq 1$. We want to show that $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. Given $v \in \mathrm{Opt}_P(w)$, we can find $z \in \mathbb{R}^n$ such that $\mathrm{Opt}_P(z) = \{v\}$. Hence for every $\epsilon > 0$ we have $\mathrm{Opt}_P(w + \epsilon z) = \{v\}$. For $\epsilon > 0$ small enough, we also have that $w + \epsilon z$ and $w' + \epsilon z$ are order-equivalent. It follows that $v \in \mathrm{Opt}_P(w' + \epsilon z)$. Letting ϵ tend to zero, we can conclude $v \in \mathrm{Opt}_P(w')$.

We have shown $\mathrm{Opt}_P(w) \subseteq \mathrm{Opt}_P(w')$. By symmetry, we also have $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. ◀

2.3 Operations on polytopes

Given $A, B \subseteq \mathbb{R}^n$, their *Minkowski sum* is defined as

$$A + B := \{a + b : a \in A, b \in B\}.$$

If P and Q are polytopes then $P + Q$ is also a polytope. In two-dimensions, Minkowski sum has nice properties. Let P be a polytope in $\mathbb{R}^2$ with vertices $v_1, \ldots, v_k$ where $k > 1$. We can assume they are ordered so that P lies in the left closed half plane determined by the line going from v_i to v_{i+1} for $i < k$, and similarly for v_k and v_1. Let $E(P)$ be the collection of unit vectors in the direction of these k edges. That is, vectors of the form $(v_{i+1} - v_i)/\|v_{i+1} - v_i\|$ for $i < k$, and $(v_1 - v_k)/\|v_1 - v_k\|$. If $|\mathsf{vert}(P)| \leq 1$ then $E(P) := \emptyset$.

▶ **Lemma 8.** *Let $P_1, \ldots, P_r$ be non-empty polytopes in $\mathbb{R}^2$. Then $E(P_1 + \cdots + P_r) = \bigcup_{i=1}^r E(P_i)$. Consequently, $|\mathsf{vert}(P_1 + \cdots + P_r)| \leq \sum_{i=1}^r |\mathsf{vert}(P_i)|$. The latter holds for empty P_i's as well.*

The lemma is folklore. It can be inferred from Chapter 13.3 in [11], and we give only an outline of proof.

Proof sketch of Lemma 8. Given a non-empty polytope P and $w \in \mathbb{R}^2$, let $P^w := \{x \in P : \langle x, w\rangle = \max_{z \in P} \langle z, w\rangle\}$ be the set of extreme of points of P in the direction w. It is either a vertex or an edge of P. For a pair of polytopes we have $(P_1 + P_2)^w = P_1^w + P_2^w$. Every edge of P_1 yields an edge of $P_1 + P_2$ with the same direction. Conversely, every edge of $P_1 + P_2$ comes from one of P_1 or P_2. ◀

The second operation we use is

$$A \sqcup B := \mathsf{conv}(A \cup B).$$

If P and Q are polytopes then $P \sqcup Q$ is also a polytope.

▶ **Lemma 9.** *Let $L : \mathbb{R}^n \to \mathbb{R}^m$ be a linear map. Given polytopes $P, Q \subseteq \mathbb{R}^n$, $L(P + Q) = L(P) + L(Q)$ and $L(P \sqcup Q) = L(P) \sqcup L(Q)$.*

Proof. The first equality holds by linearity. The second one can be proved by

$$\begin{aligned} L(P \sqcup Q) &= L(\mathsf{conv}(P \cup Q)) = \mathsf{conv}(L(P \cup Q)) \\ &= \mathsf{conv}(L(P) \cup L(Q)) = L(P) \sqcup L(Q)\,. \end{aligned}$$ ◀

We next relate the shadow complexity of P with the shadow complexity of its faces. A *face* of a polytope P is the intersection of P with a hyperplane H such that P is completely contained in one of the two closed halfspaces determined by H. We stipulate that both $\emptyset$ and P are faces of P.

▶ **Lemma 10.** *Let F be a face of a polytope P. Then $\beta(F) \leq \beta(P)$ and $\sigma(F) \leq 2\sigma(P)$.*

For example, this implies $\beta(\mathsf{DS}_{n_1}) \leq \beta(\mathsf{DS}_{n_2})$ whenever $n_1 \leq n_2$. This reflects the fact that finding a maximum perfect matching is harder for larger graphs.

Proof. Without loss of generality, assume that $P \subset \mathbb{R}^n$ is contained in the halfspace $\{x \in \mathbb{R}^n : x_1 \geq 0\}$ and that $F \notin \{\emptyset, P\}$ is the intersection with the hyperplane $x_1 = 0$.

Let $w(t)$ be a line in $\mathbb{R}^n$ so that $\beta(F, w(t)) = \beta(F) = k$ with $w(t)_1 = 0$. Let $t_1 < t_2$ be such that the breakpoints of $\max_{x\in F} \langle x, w(t)\rangle$ are contained in the open interval (t_1, t_2). Let $V := \mathsf{vert}(P) \setminus \mathsf{vert}(F)$. Define

$$\mu_F := \min_{x\in F, t\in[t_1,t_2]} \langle x, w(t)\rangle$$

and

$$\mu_P := \max_{v\in V, t\in[t_1,t_2]} \langle v, w(t)\rangle \ .$$

Take $\lambda \in \mathbb{R}$ sufficiently small so that for every $v \in V$, we have $\lambda v_1 + \mu_P < \mu_F$. Define $\bar{w}(t)$ by changing the first coordinate of $w(t)$ to $\lambda + 0 \cdot t$. This means that

$$\max_{x\in F} \langle x, w(t)\rangle = \max_{x\in P} \langle x, \bar{w}(t)\rangle$$

holds on $[t_1, t_2]$. So, $\beta(P, \bar{w}(t)) = k$ and $\beta(P) \geq \beta(F)$.

If $|\mathsf{vert}(F)| \leq 2$, then $\sigma(F) \leq 2\sigma(P)$ holds trivially. Otherwise, $\sigma(F) \leq 2\sigma(P)$ follows from Theorem 4. ◀

2.4 Laurent polynomials

It is convenient to work with *Laurent polynomials* instead of polynomials. In a Laurent polynomial, variables are allowed to have negative integer exponents. The notions of $\mathsf{supp}(f)$ and Newton polytope of f are defined in the obvious manner. A Laurent polynomial over $\mathbb{R}$ is *monotone*, if all of its coefficients are non-negative.

▶ **Lemma 11.** *Let f, g be Laurent polynomials over $\mathbb{F}$.*

(i) *Then* $\mathsf{Newt}(fg) = \mathsf{Newt}(f) + \mathsf{Newt}(g)$.

(ii) $\mathsf{Newt}(f + g) = \mathsf{Newt}(f) \sqcup \mathsf{Newt}(g)$, *provided $\mathbb{F} = \mathbb{R}$ and both f and g are monotone.*

Proof. Part (i) can be found in [16] for polynomials; it extends to Laurent polynomials. Part (ii) is straightforward to verify. ◀

An application is that the shadow complexity of $\mathsf{Newt}(g)$ is at least the shadow complexity of any of its factors.

▶ **Lemma 12.** *Let g be a non-zero polynomial (over an arbitrary field). If f divides g then* $\sigma(\mathsf{Newt}(f)) \leq \sigma(\mathsf{Newt}(g))$.

Proof. Let L be such that $L(\mathsf{Newt}(f)) \subseteq \mathbb{R}^2$ has $\sigma(\mathsf{Newt}(f))$ vertices. By the assumption, we have $g = fh$ for some non-zero polynomial h and so $\mathsf{Newt}(g) = \mathsf{Newt}(f) + \mathsf{Newt}(h)$ by Lemma 11. By Lemma 9, we have $L(\mathsf{Newt}(g)) = L(\mathsf{Newt}(f)) + L(\mathsf{Newt}(h))$ and so $|\mathsf{vert}(L(\mathsf{Newt}(g)))| \geq |\mathsf{vert}(L(\mathsf{Newt}(f)))|$ by Lemma 8. ◀

3 Examples

We now describe some examples, and analyze the shadow complexity of several natural polytopes. We start with polytopes with small σ, continue with polytopes with large σ, and then discuss our favorites, the ones where we do not yet know.

3.1 The hypercube

Optimizing over the discrete cube $\{0,1\}^n \subset \mathbb{R}^n$ leads to the polytope $\mathsf{Q}_n = [0,1]^n$. The solid cube Q_n has 2^n vertices, but its shadow complexity is small.

▶ **Proposition 13.** *$\sigma(\mathsf{Q}_n) = 2n$ and $\beta(\mathsf{Q}_n) = n$.*

The proposition shows that the factor of two in Theorem 4 is necessary. The lower bound on σ also follows from a more general theorem of Klee [27].

Proof. Let $\ell_i \subseteq \mathbb{R}^n$ be the line segment joining the origin and the i-th unit vector for $i \in [n]$. The solid cube Q_n is the Minkowski sum of $\ell_1, \dots, \ell_n$. Given $L : \mathbb{R}^n \to \mathbb{R}^2$, the image $L(\mathsf{Q}_n)$ is the Minkowski sum of $L(\ell_1), \dots, L(\ell_n)$ by Lemma 9. Since $|\mathsf{vert}(L(\ell_i))| \leq 2$, Lemma 8 gives that $|\mathsf{vert}(L(\mathsf{Q}_n))| \leq 2n$. The bound $\sigma(\mathsf{Q}_n) \geq 2n$ is achieved by the same lemma. It is enough to take L so that $L(\ell_i)$ are not parallel to get $|\mathsf{vert}(L(\mathsf{Q}_n))| = 2n$.

The above and Theorem 4 imply that $\beta(\mathsf{Q}_n) \geq n$. It remains to prove $\beta(\mathsf{Q}_n) \leq n$. For every $w \in \mathbb{R}^n$, the maximum $\max_{x \in \mathsf{Q}_n} \langle x, w \rangle$ equals the sum of the positive entries in w, or zero if all entries are non-positive. A breakpoint of $\max_{x \in \mathsf{Q}_n} \langle x, w(t) \rangle$ can therefore occur only when some coordinate of $w(t)$ changes sign. A linear function can change sign at most once and there are n linear functions. ◀

▶ Remark 14. The solid cube Q_n is not greedy-like as defined above. This is because in the optimization algorithm, we must distinguish which entries are non-negative. Shifting all coordinates of w by λ does not change their order but may change where the maximum is achieved.

3.2 Permutahedra

Given $z = (z_1, \dots, z_n) \in \mathbb{R}^n$, let

$$P(z) := \mathsf{conv}\{(z_{\pi(1)}, \dots, z_{\pi(n)}) : \pi \in S_n\},$$

where S_n is the family of permutations of $[n]$. The *permutahedron* is usually defined using the point $z = (0, 1, \dots, n-1)$. However, we do not insist z to have distinct coordinates. Setting z to be a zero-one vector with k ones, $P(z)$ becomes the convex hull of Boolean vectors of Hamming weight k. For every z, the polytope $P(z)$ is a linear projection of DS_n. The polytope $P(z)$ typically has $n!$ vertices, but its shadow complexity is always small.

▶ **Proposition 15.** *For every $z \in \mathbb{R}^n$, $\sigma(P(z)) \leq n(n-1)$. The bound is attained for $z = (0, 1, \dots, n-1)$.*

Proof. Let $z := (0, 1, \dots, n-1)$. Let $e_i \in \mathbb{R}^n$ be the i-th unit vector. Let $\ell_{i,j}$ be the line segment joining e_i and e_j for $i \neq j$. We claim that the polytope $P(z)$ can be written as the following Mikowski sum

$$P(z) = \bigoplus_{i<j} \ell_{i,j}. \tag{1}$$

Indeed, let X be the $n \times n$ matrix such that $X_{i,j} = x_i^{z_j}$. Observe that

$$P(z) = \mathsf{Newt}(\mathsf{det}(X))\,.$$

The matrix X is a Vandermonde matrix whose determinant, over any field, factorizes as $\mathsf{det}(X) = \prod_{i<j}(x_j - x_i)$. Lemma 11 implies (1).

Now, given $L : \mathbb{R}^n \to \mathbb{R}^2$, we thus have $L(P(z)) = \bigoplus_{i<j} L(\ell_{i,j})$. By Lemma 8, if we choose L so that the lines $L(\ell_{i,j})$ are non-parallel, the number of vertices of $L(P)$ is $2 \cdot \binom{n}{2} = n(n-1)$.

The general upper bound is an application of Lemma 5. We claim that $P(z)$ is greedy-like. Permuting the entries of z does not changes σ. So, we can assume that $z_1 \leq z_2 \leq \dots z_n$. Given $w \in \mathbb{R}^n$,

$$\max_{x \in P(z)} \langle x, w \rangle = \max_{\pi \in S_n} \langle z, w_\pi \rangle\,,$$

where $w_\pi := (w_{\pi(1)}, \dots, w_{\pi(n)})$. The maximum is achieved iff $w_{\pi(1)} \leq w_{\pi(2)} \cdots \leq w_{\pi(n)}$. This means that $\mathrm{Opt}_w(P(z)) = \mathrm{Opt}_{w'}(P(z))$ whenever w and w' are order-equivalent. ◀

▶ Remark 16. Here we provide an additional algebraic proof. Consider $z = (z_1, \dots, z_n)$ with $z_i = 2^{i-1}$. The matrix X defined by $X_{i,j} = x_i^{z_j}$ is a *Moore matrix* [34]. Over $\mathbb{F} = GF(2)$, the polynomial $\mathsf{det}(X(z))$ factorizes as

$$\mathsf{det}(X(z)) = \prod_{A \subseteq [n]} \sum_{i \in A} x_i\,.$$

The number of factors is exponential but we can still get a quadratic upper bound. We have $P(z) = \bigoplus_{A \subseteq [n]} R_A$ where $R_A = \mathsf{conv}\{e_i : i \in A\}$. Given a projection $L : \mathbb{R}^n \to \mathbb{R}^2$, we have $L(P(z)) = \bigoplus_{A \subseteq [n]} L(R_A)$. The polytopes $L(R_A)$ contain at most $\binom{n}{2}$ non-parallel edges and hence $L(P(z))$ has again at most $n(n-1)$ vertices.

3.3 Spanning trees

Every $\alpha \in \{0,1\}^{\binom{n}{2}}$ can be interpreted as the incidence vector of an undirected graph on n vertices. Namely, $\alpha_{i,j} = 1$, if i, j are adjacent, and $\alpha_{i,j} = 0$ otherwise. The polytope TREE_n is defined as the convex hull of spanning trees of the complete n-vertex graph.

▶ **Proposition 17.** $\sigma(\mathsf{TREE}_n) \leq n^4$.

Proof. By Lemma 5, it is enough to show that $P = \mathsf{TREE}_n$ is greedy-like. Indeed, Kruskal's algorithm for finding a minimum weight spanning tree takes into account only the ordering of weights on the edges. That is, if w, w' are order-equivalent and $\mathrm{Opt}_P(w)$ is a singleton then $\mathrm{Opt}_P(w) = \mathrm{Opt}_P(w')$. Hence TREE_n is greedy-like by Lemma 7. . ◀

▶ Remark 18. This is interesting when contrasted with algebraic complexity. Consider the unique polynomial Tree_n with zero-one coefficients so that $\mathsf{Newt}(\mathsf{Tree}_n) = \mathsf{TREE}_n$. It is a homogeneous multilinear polynomial of degree $n-1$. Proposition 17 shows that the shadow complexity of its Newton polytope is polynomial. On the other hand, Jerrum and Snir showed that Tree_n requires exponential monotone arithmetic circuit [23]. They also pointed out that it has a non-monotone circuit of polynomial size. More surprisingly, Tree_n has a monotone circuit with division of polynomial size [14].

3.4 Cliques

The *correlation polytope* $\mathsf{COR}_n \subseteq \mathbb{R}^{n\times n}$ is the convex hull of all symmetric rank-one Boolean matrices:

$$\mathsf{COR}_n = \mathsf{conv}\{bb^t : b \in \{0,1\}^n\}\,.$$

▶ **Proposition 19.** *$\sigma(\mathsf{COR}_n) = 2^n$.*

Proof. Let $e_{i,j}$ be the $n \times n$ matrix whose (i,j) entry is one and every other entry is zero. The vertices of COR_n are of the form $v_A = \sum_{i,j\in A} e_{i,j}$ with $A \subseteq [n]$. Define

$$L(e_{i,j}) := \begin{cases} (2^i, 2^{2i}) & i = j, \\ (0, 2^{i+j}) & i \neq j, \end{cases}$$

and extend it linearly to $\mathbb{R}^{n\times n}$. Setting $n_A := \sum_{i\in A} 2^i$, this guarantees

$$L(v_A) = (\sum_{i\in A} 2^i, \sum_{i,j\in A} 2^{i+j}) = (n_A, n_A^2).$$

These 2^n points are convexly independent. ◀

▶ Remark 20. The polytope COR_n lives in dimension $N = n^2$, and so $\sigma(\mathsf{COR}) = 2^{\sqrt{N}}$. The polytope $\mathsf{ART}_n \subseteq \mathbb{R}^{3n}$, which we define next, has truly exponential shadow complexity. It is defined as the convex hull of

$$\Big\{(a_0, \ldots, a_{n-1}, b_0, \ldots, b_{2n-1}) \in \{0,1\}^{3n} : \sum_{i=0}^{2n-1} b_i 2^i = \Big(\sum_{i=0}^{n-1} a_i 2^i\Big)^2\Big\}.$$

In words, b is the binary representation of the square of the number represented by a. It follows that $\sigma(\mathsf{ART}_n) = 2^n$.

▶ Remark 21. The polynomial that corresponds to COR_n is

$$\mathsf{Clique}_n = \sum_{A\subseteq[n]} \prod_{i,j\in A} x_{i,j}\,.$$

It has n^2 variables and $\mathsf{Newt}(\mathsf{Clique}_n) = \mathsf{COR}_n$. We can interpret the polynomial as counting cliques of all sizes in a directed graph with loops, hence the name.

3.5 More graph-based polytopes

Consider a layered directed graph G_n as follows. The vertex-set of G_n is partitioned into layers $V_0, \ldots V_n$. The first and the last layer consist of a single vertex s and t. Every other layer has n vertices. The edges are all pairs from $V_i \times V_{i+1}$ directed from layer i to $i+1$. Overall, G_n has $n(n-1)+2$ vertices and $N := (n-2)n^2 + 2n$ edges. Let $\mathsf{CONN}_n \subseteq \mathbb{R}^N$ be the convex hull of incidence vectors of directed paths from s to t in G_n. The following proposition can be found in [8, 37], where the results are stated in terms of the parametrized complexity β, which is equivalent to the shadow complexity by Theorem 4.

▶ **Proposition 22.** *$\sigma(\mathsf{CONN}_n) = 2^{\Theta(\log^2 n)}$.*

We now deduce the best bound we are aware of for the Birkhoff polytope.

▶ **Proposition 23.** *$2^{\Omega(\log^2 n)} \leq \sigma(\mathsf{DS}_n) \leq 2^{O(n)}$.*

Proof. As pointed by Mulmuley and Shah in [37], the lower bound for CONN_n translates to DS_n. For the upper bound, we claim that

$$\sigma(\mathsf{DS}_{2n}) \leq 2\binom{2n}{n}\sigma(\mathsf{DS}_n)\,. \tag{2}$$

By induction, this indeed implies $\sigma(\mathsf{DS}_n) \leq 2^{O(n)}$.

Let us prove (2). Given $A \subseteq [2n]$ with $|A| = n$, let Π_A be the set of permutation matrices which, when viewed as a permutation on $[2n]$, map $\{1,\ldots,n\}$ to A. The set of all $2n \times 2n$ permutation matrices is the union of all Π_A with $|A| = n$. Hence,

$$\mathsf{DS}_{2n} = \mathsf{conv}\Big(\bigcup_{A:\,|A|=n} \Pi_A\Big).$$

We can view $\mathsf{conv}(\Pi_A)$ as the Minkowski sum of two copies of DS_n embedded into $\mathbb{R}^{2n\times 2n}$. Given $L : \mathbb{R}^{2n\times 2n} \to \mathbb{R}^2$ this gives, by Lemma 8, $|\mathsf{vert}(L(\mathsf{conv}(\Pi(A))))| \leq 2|\mathsf{vert}(L(\mathsf{DS}_n))|$. The bound in (2) follows. ◀

▶ Remark 24. The upper bound on DS_n is more exactly of the form $2^{(2-o(1))n}$. In the proof, we implicitly construct a monotone arithmetic formula for perm_n of this size. This matches the lower bound from [43]. Curiously, perm_n has a monotone circuit of size $O(n2^n)$ [23] and a (non-monotone) formula of size $O(n^2 2^n)$ [42].

▶ Remark 25. Let $\mathsf{Mat}_n := (X_0 \cdot X_1 \cdots X_n)_{1,1}$, where $X_0,\ldots,X_n$ are $n \times n$ matrices whose entries are distinct variables. Then $\mathsf{Newt}(\mathsf{Mat}_n) = \mathsf{CONN}_n$.

▶ Remark 26. The *perfect matching polytope* MATCH_n is the the convex hull of incidence vectors of perfect matchings in the complete (non-bipartite) graph on $2n$ vertices. A similar argument to the proof of Proposition 23 gives

$$\sigma(\mathsf{DS}_n) \leq \sigma(\mathsf{MATCH}_n) \leq \binom{2n}{n}\sigma(\mathsf{DS}_n) \leq 2^{O(n)}\,.$$

4 Projections

We now discuss some connection between algebraic projections of polynomials and linear projections of Newton polytopes. The results here shall also be used later on.

A *high power* projection (h.p.-projection for short) is a homomorphism

$$\pi : \mathbb{F}[x_1,\ldots,x_n] \to \mathbb{F}[y_1,\ldots,y_m,y_1^{-1},\ldots y_m^{-1}]$$

such that $\pi(x_i) = a_i y^{\alpha_i}$ for every x_i, where $a_i \in \mathbb{F}$ and $\alpha_i \in \mathbb{Z}^m$, and for every $f \in \mathbb{F}[x_1,\ldots,x_n]$,

$$\pi(f(x_1,\ldots,x_n)) = f(\pi(x_1),\ldots,\pi(x_n)).$$

The constants a_i are called the *coefficients* of π and α_i its *exponents*. If $\mathbb{F} = \mathbb{R}$ and π has non-negative coefficients, then π is called *monotone*.

An h.p.-projection π induces a linear map $L_\pi : \mathbb{R}^n \to \mathbb{R}^m$ by setting $L_\pi(e_i) = \alpha_i$ and extending it linearly to $\mathbb{R}^n$. It follows that $\mathsf{supp}(\pi(f)) \subseteq L_\pi(\mathsf{supp}(f))$. The inclusion may be strict, as some monomials of f can cancel out in the projection. If no cancellations occur, we indeed have $\mathsf{Newt}(\pi(f)) = L_\pi(\mathsf{Newt}(f))$. This is satisfied, e.g., if f is monotone and the coefficients of π are positive, or if the coefficients are algebraically independent.

In the other direction, take $L : \mathbb{R}^n \to \mathbb{R}^m$ a linear map defined by $m \times n$ matrix with integer coefficients. Consider a h.p.-projection π_L of the form $\pi(x_i) = a_i x_i^{L(e_i)}$. If we choose the coefficients a_i to be sufficiently independent, we again obtain $L(\mathsf{Newt}(f)) = \mathsf{Newt}(\pi_L(f))$.

The following we do not really need, but it puts things into perspective. A similar fact has been noted by Grochow [20].

▶ **Proposition 27.** *Let f be a monotone polynomial. Assume that a Laurent polynomial g is a monotone h.p.-projection of f. Then $\mathsf{Newt}(g)$ is a linear projection of some face of $\mathsf{Newt}(f)$. Hence $\sigma(\mathsf{Newt}(g)) \leq 2\sigma(\mathsf{Newt}(f))$.*

Proof. Assume $g = \pi(f)$ with π an h.p.-projection. Let $A \subseteq [n]$ be the set of $i \in [n]$ with $a_i = 0$. Let f^* be the polynomial obtained by substituting 0 for x_i for every $i \in A$. The polytope $\mathsf{Newt}(f^*)$ is a face of $\mathsf{Newt}(f)$. Indeed, since f has non-negative exponents, $\mathsf{Newt}(f^*) = \mathsf{Newt}(f) \cap H$ where H is the hyperplane defined by $\sum_{i \in A} z_i = 0$, and $\mathsf{Newt}(f)$ lies in the halfspace $\sum_{i \in A} z_i \geq 0$.

We can now write $\pi(f) = \pi^*(f^*)$ where π^* has positive coefficients. This means that $\mathsf{supp}(\pi(f)) = L_{\pi^*}(\mathsf{supp}(f^*))$ and hence $\mathsf{Newt}(\pi(f)) = L_{\pi^*}(\mathsf{Newt}(f^*))$. The bound on σ follows from Lemma 10 ◀

The following we do need:

▶ **Lemma 28.** *Let f be a polynomial over an infinite field $\mathbb{F}$. Assume that $\sigma(\mathsf{Newt}(f)) = k$. Then there exists a bivariate Laurent polynomial $g \in \mathbb{F}(y_1, y_2, y_1^{-1}, y_2^{-1})$ which is an h.p.-projection of f so that $\mathsf{Newt}(g)$ has k vertices. Moreover, if f is a homogeneous polynomial then g is a polynomial. If $\mathbb{F} = \mathbb{R}$, then the coefficients in the projection can be assumed positive.*

Proof. Let $L(z) = Az$ with $A \in \mathbb{R}^{2 \times n}$ be a linear map so that

$$|\mathsf{vert}(L(\mathsf{Newt}(f)))| = k.$$

We can assume that the entries of A are rational, because a small perturbation of A cannot decrease $|\mathsf{vert}(L(\mathsf{Newt}(f)))|$. Now, we can assume that the entries of A are integers, because we can multiply A by a suitable integer.

Moreover, when f is homogeneous of degree d, increasing all entries of A by λ corresponds to shifting $L(\mathsf{Newt}(f))$ by $(\lambda d, \lambda d)$, which does not change the number of vertices. Hence, in this case, A can be taken with non-negative integer entries.

Let us now consider a h.p.-projection π with $\pi(x_i) = a_i y^{L(e_i)}$. It follows that $\mathsf{supp}(\pi(f)) \subseteq L(\mathsf{supp}(f))$. Now, we claim that we can choose the coefficients a_i so that equality holds. This can be seen as follows. Given $\alpha \in \mathsf{supp}(f)$, the coefficient of $y^{L(\alpha)}$ in $\pi(f)$ is a non-zero polynomial h_α in the coefficients of π. Hence, if $\mathbb{F}$ is infinite, there exist non-zero coefficients for which h_α is non zero for every $\alpha \in \mathsf{supp}(f)$. If $\mathbb{F} = \mathbb{R}$, they can be taken positive. ◀

▶ Remark 29. We emphasize the difference between linear projections of polytopes and algebraic projections of polynomials. Since the permanent polynomial is VNP-complete, Clique_n from Remark 21 is a projection (in the common sense) of perm_m with m polynomial in n. However, $\mathsf{Newt}(\mathsf{Clique}_n)$ is not a linear projection of $\mathsf{Newt}(\mathsf{perm}_m)$, neither of any of its faces, unless m is exponentially large [20]. The idea is that DS_m has $O(m^2)$ facets whereas $\mathsf{Newt}(\mathsf{Clique}_n)$ is not a projection of any polytope with few facets. It follows that Clique_n is not a monotone projection of perm_m.

5 Monotone computation

As the standard model of computation of polynomials we take the *arithmetic circuit* model. It is a (finite) directed acyclic graph whose every node has in-degree zero or two. Nodes of in-degree zero (input nodes) are labelled with a constant or a variable. Nodes of in-degree two are labelled with operations + or ×. Every node of the circuit computes a polynomial in $\mathbb{F}$ in the natural way. As the *size* of the circuit, we take the number of nodes. A circuit is called a *formula* if its underlying graph is a tree. For more background and motivation, see [44] and references within.

Our focus is mainly on monotone computation. A polynomial over $\mathbb{R}$ is *monotone* if all of its coefficients are non-negative. Similarly, a *monotone arithmetic circuit* can use only non-negative constants.

5.1 Optimization problems

We start with a somewhat surprising connection between monotone computation and Newton polytopes. A monotone circuit over $\mathbb{R}$ computing f can be interpreted as a computation over the semi-ring $M = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$. That is, replace "+" by "min", replace "×" by "+", replace "0" by "∞", and replace every positive constant "a" by "0". A circuit with operations from M has also been called a *tropical* circuit [24]. The resulting circuit computes the function $f^* : \mathbb{R}^n \to \mathbb{R}$ which turns out to be precisely

$$f^*(w) = \min_{x \in \mathsf{Newt}(f)} \langle x, w \rangle \, .$$

For example, any monotone circuit for the permanent polynomial can also be viewed as a tropical circuit for the minimum weight perfect matching in a bipartite graph. Computations over general semi-rings have been considered in [23, 24], where the reader can find more details.

5.2 Shadows and monotone computations

We explore some connections between the structure of the Newton polytope of f and monotone arithmetic computations. We prove that shadow complexity allows to prove lower bounds on monotone complexity (Theorems 1 and 2). We also show that in general Theorem 1 does not hold for circuits instead of formulas and so the assumption of transparency in Theorem 2 cannot be removed.

▶ **Theorem 30.** *For every n, there exists a monotone bivariate polynomial f_n such that f_n has a monotone arithmetic circuit of size $O(n)$ and $\mathsf{Newt}(f)$ has 2^n vertices.*

Theorem 30 is proved in Section 8. The construction is reminiscent of that in [3] of a univariate polynomial with circuit of size s and $2^{\Omega(s)}$ real roots (cf. Chapter 12 of [6]). A weaker bound can also be deduced as follows:

▶ Remark 31. Recall the polynomial Mat_n from Remark 25. Then Mat_n has a monotone circuit of size $O(n^4)$ whereas $\sigma(\mathsf{Newt}(\mathsf{Mat}_n)) = 2^{\Omega(\log^2 n)}$.

▶ Remark 32. When a monotone arithmetic formula is interpreted as a tropical formula (cf. Section 5.1), it becomes an instance of parallel computation in the PRAM model without bit operations of Mulmuley [36]. Hence Theorem 1 can be seen as special case[3] of Theorem 3.3 from [36].

[3] This is not a "black box" reduction. Mulmuley's result has an additional parameter representing bit size of the input, whereas we have no such thing.

5.3 Monotone formulas

Here we show that shadow complexity allows to lower bound monotone formula complexity.

A *high powered* circuit (h.p.-circuit for short) is an arithmetic circuit in which every input node is labelled by a term $ax_1^{k_1} \cdots x_n^{k_n}$ with $a \in \mathbb{F}$ and $k_1, \ldots, k_n \in \mathbb{Z}$. The size of the circuit is the number of its nodes.

In other words, we have given the circuit a power to compute every term ax^α at a unit cost. This is especially important in the case of h.p.-formula. An arithmetic formula of size s can compute a polynomial of degree at most s, whereas there is no such restriction in an h.p.-formula. Furthermore, we have allowed the variables to have negative exponents and hence an h.p.-circuit computes a Laurent polynomial instead of a polynomial. But this is only a cosmetic detail.

▶ **Theorem 33.** *Let f be a monotone bivariate Laurent polynomial such that $\mathsf{Newt}(f)$ has k vertices. Then every monotone h.p.-formula computing f has at least k leaves.*

Proof. Straightforward induction using Lemma 11 and 8. ◀

We can now prove that every monotone formula computing f contains at least $\sigma(\mathsf{Newt}(f))$ leaves.

Proof of Theorem 1. By Lemma 28 there exists a bivariate g which is a monotone h.p.-projection of f so that $\mathsf{Newt}(g)$ has k vertices. The projection also transforms a monotone formula for f to a monotone h.p.-formula for g. ◀

5.4 Lower bounds from extension complexity

As mentioned in Section 1.2, one can obtain monotone formula lower bounds also from extensions complexity of Newton polytopes. The main ingredient is the following lemma.

▶ **Lemma 34.** *For polytopes $P, Q \subseteq \mathbb{R}^n$ we have*

$$\mathrm{xc}(P+Q) \leq \mathrm{xc}(P) + \mathrm{xc}(Q) \quad \textit{and} \quad \mathrm{xc}(P \sqcup Q) \leq \mathrm{xc}(P) + \mathrm{xc}(Q) + 2.$$

Proof. The first inequality is rather obvious. The second follows from a theorem of Balas [2], see also [10]. ◀

The lower bound is now proved by a straightforward induction.

▶ **Theorem 35.** *Assume that f has a monotone formula of size s. Then $\mathrm{xc}(\mathsf{Newt}(f)) \leq O(s)$.*

▶ Remark 36. The Pfaffian Pf_n is the polynomial so that $\mathsf{Pf}_n^2 = \mathsf{det}(X)$, where X is the $2n \times 2n$ antisymmetric matrix with $X_{i,i} = 0$ and $X_{i,j} = -X_{j,i} = x_{i,j}$ if $i < j$. The Pfaffian has an arithmetic circuit of size polynomial in n, and a formula of size $2^{O(\log^2 n)}$; see [45]. The Newton polytope $\mathsf{Newt}(\mathsf{Pf}_n)$ is the perfect matching polytope MATCH_n, as described in Remark 26. By a result of Rothvoss [41], MATCH_n has extension complexity $2^{\Omega(n)}$.

5.5 Monotone circuits

We move to proving the circuit lower bound stated in Theorem 2. We first observe that Minkowski sum typically can not avoid convex independence.

▶ **Lemma 37.** *Let $A, B \subseteq \mathbb{R}^2$ be non-empty sets such that $A + B$ is a convexly independent set with $|A| \geq |B|$. Then either $|A| \leq 2$ or $|B| \leq 1$.*

Proof. For the sake of contradiction, assume that $A+B$ is convexly independent, $|A| \geq 3$ and $|B| \geq 2$. By Lemma 8, the convex hull of $A+B$ has at most $|A|+|B|$ vertices. By the size assumption, there exist $a_1 \neq a_2 \in A$ and $b_1 \neq b_2 \in B$ with $a_1+b_1 = a_2+b_2$. The point a_1+b_1 is the average of a_1+b_2 and a_2+b_1 and it is distinct from them, a contradiction. ◀

▶ **Theorem 38.** *Let f be a monotone bivariate Laurent polynomial such that $\mathsf{supp}(f)$ is convexly independent and $|\mathsf{supp}(f)| = k$. Then f requires monotone h.p.-circuit with $k/4$ gates.*

Theorem 38 implies Theorem 2 via Lemma 28.

Proof. The lower bound is proved using the following "progress" measure. Given $A \subseteq \mathbb{R}^2$ and $\epsilon \in \{0,1\}$, let $A^\epsilon := A$ if $\epsilon = 1$ and $A^\epsilon := \emptyset$ if $\epsilon = 0$. Given $v \in \mathbb{R}^2$, let $v + A := \{v\} + A$. Let $\mathcal{A}$ be a finite set of finite subsets of $\mathbb{R}^2$. For functions $\epsilon : \mathcal{A} \to \{0,1\}$ and $v : \mathcal{A} \to \mathbb{R}^2$, let

$$\mathcal{A}_{\epsilon,v} = \bigcup_{A \in \mathcal{A}} (v(A) + A)^{\epsilon(A)}.$$

Let

$$\mu(\mathcal{A}) = \max_{\epsilon,v}\{|\mathcal{A}_{\epsilon,v}| : \ \mathcal{A}_{\epsilon,v} \text{ is convexly independent}\}.$$

▷ Claim. Let $\mathcal{A}' = \mathcal{A} \cup \{B\}$ and $A_1, A_2 \in \mathcal{A}$. Then

$$\begin{aligned}
\mu(\mathcal{A}') \leq & \mu(\mathcal{A}) + |B|\,, && (3)\\
\mu(\mathcal{A}') \leq & \mu(\mathcal{A}) + 2\,, \text{ if } B = u + A_1 \text{ for some } u \in \mathbb{R}^2\,, && (4)\\
\mu(\mathcal{A}') \leq & \mu(\mathcal{A}) + 4\,, \text{ if } B = A_1 \cup A_2\,, && (5)\\
\mu(\mathcal{A}') \leq & \mu(A) + 4\,, \text{ if } B = A_1 + A_2\,. && (6)
\end{aligned}$$

Proof of Claim. Inequality (3) is straightforward.

To prove (4), suppose that ϵ, v are such that $\mathcal{A}'_{\epsilon,v}$ is convexly independent. Suppose $\epsilon(A_1) = \epsilon(B) = 1$ and $v(A_1) + A_1 \neq v(B) + B$; otherwise we have $|\mathcal{A}'_{\epsilon,v}| \leq \mu(\mathcal{A})$. Then $(v(A_1) + A_1) \cup (v(B) + B) = \{v(A_1), v(B) + u\} + A_1$ is convexly independent. Since $|\{v(A_1), v(B) + u\}| = 2$, by Lemma 37, A_1 has size at most 2. This means $\mu(\mathcal{A}') \leq \mu(\mathcal{A}) + 2$ by (3).

For (5), observe that $\mu(\mathcal{A}') \leq \mu(\mathcal{A} \cup \{u_1 + A_1, u_2 + A_2\})$ whenever $u_1, u_2 \neq 0$ are distinct and apply (4) twice.

Finally, we prove (6). If $B = A_1 + A_2$ is not convexly dependent, it contributes nothing to μ. Assume that B is convexly independent and $|A_1| \geq |A_2| > 0$. By Lemma 37, either $|A_1 + A_2| \leq 4$ or $|A_2| = 1$. In the former case, $\mu(\mathcal{A}') \leq \mu(\mathcal{A}) + 4$ by (3). In the latter, $A_2 = \{u\}$ for some u and $\mathcal{A}' = \mathcal{A} \cup \{u + A_1\}$ and we can apply (4). ◁

Let us call a h.p.-circuit *transparent*, if every gate in the circuit computes a polynomial with convexly independent support. Given a circuit Ψ and a node u, let $\mathsf{supp}(u)$ be the support of the Laurent polynomial computed by u. Let $\mathcal{A}_\Psi$ be the set $\{\mathsf{supp}(u) : u \in \Psi\}$.

Using the Claim, we can show that whenever a transparent and monotone Ψ has s gates then $\mu(\mathcal{A}_\Psi) \leq 4s$. The proof is by induction. The induction base $s = 1$ trivially holds. It remains to verify the induction step. Let u be an output gate of Ψ. If u is also an input gate, apply (3). If $u = u_1 + u_2$ then $\mathsf{supp}(u) = \mathsf{supp}(u_1) \cup \mathsf{supp}(u_2)$ and (5) completes the proof. If $u = u_1 \times u_2$ then $\mathsf{supp}(u) = \mathsf{supp}(u_1) + \mathsf{supp}(u_2)$ and (6) completes the proof.

Finally, consider a monotone circuit Ψ for f of minimal size s. No gate in the circuit computes the zero polynomial (unless f itself the zero polynomial). The circuit is transparent because a monotone computation does not cancel monomials unless multiplying by zero, and because $+, \times$ can not "undo" convex dependence. This means that $\mu(\mathcal{A}_\Psi) \leq 4s$. On the other hand, since $\mathsf{supp}(f)$ consists of k convexly independent points, we have $\mu(\mathcal{A}_\Psi) \geq |\mathsf{supp}(f)| = k$. ◀

Other illustrative consequences are the following:

▶ **Corollary 39.** $\sum_{k=0}^{n} x^k y^{k^2}$ *requires monotone h.p.-arithmetic circuit of size* $\Omega(n)$.

Recall the Clique_n polynomial from Remark 21 and the polytope ART_n from Remark 20. Let Art_n be the unique polynomial with zero-one coefficients so that $\mathsf{Newt}(\mathsf{Art}_n) = \mathsf{ART}_n$.

▶ **Corollary 40.** *Both* Clique_n *and* Art_n *require monotone arithmetic circuits of size* $\Omega(2^n)$.

Proof. Proposition 19 and Remark 20 show that Clique_n and Art_n are transparent with shadow complexity 2^n. ◀

5.6 Generalizations

The results of this section can be strengthened in several ways. First, one could extend the notion of monotone computation to any field. A monotone circuit would be such that for every sum gate $f_1 + f_2$, no monomial can vanish[4]: $\mathsf{supp}(f_1 + f_2) = \mathsf{supp}(f_1) \cup \mathsf{supp}(f_2)$. Then Theorem 1 goes through.

Second, one may consider circuits with high-power gates. This would be an arithmetic circuit which, apart from the $+, \times$ gates, can use also unary gates of the form $(\ )^k$ which raises its input to a power of $k \in \mathbb{N}$. A similar notion has appeared in the context of additive complexity of a polynomial and counting real roots of univariate polynomials (see Section 12.3 of [6] and references within). Our lower bounds hold also in this setting. This is because $\mathsf{Newt}(f^k)$ with $k > 0$ is merely a scaling of $\mathsf{Newt}(f)$.

Finally, our results extend to other semi-rings as well. For definitions of polynomials over semi-rings and their computations see, e.g., [23, 24]. Let $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ be the Boolean semi-ring.

▶ **Proposition 41.** *Theorems 1 and 2 hold also over* $\mathbb{B}$.

Proof. Given a circuit over $\mathbb{B}$ computing f, we can interpret it as a computation over $\mathbb{R}$ by replacing $\wedge$ by $\times$ and replacing $\vee$ by $+$. The circuit then computes a polynomial f^* over $\mathbb{R}$ with $\mathsf{supp}(f^*) = \mathsf{supp}(f)$. Since the two theorems take into account only $\mathsf{supp}(f^*)$, they hold over $\mathbb{B}$ as well. ◀

6 Divisions

The model of monotone circuits can be extended to include *division gates*. We may allow the circuit to use an extra gate computing f/g. A monotone circuit with divisions can compute a non-monotone polynomial; e.g., $x^2 - x + 1 = \frac{x^3+1}{x+1}$.

Monotone circuits with divisions were extensively studied by Fomin et al. [14]. They proved, among other nice things, a separation between monotone circuits and monotone circuits with division. The *Spanning Tree* polynomial (see Section 3.3) has a polynomial size

[4] Monomials can however vanish on a product as in $(x+y)(x-y) = x^2 - y^2$.

monotone circuit with divisions but requires an exponential size monotone circuit by [23]. This is in sharp contrast with the result of Strassen that division gates cannot help in the general arithmetic setting[5].

Super-polynomial lower bounds on monotone circuits with division computing a *monotone* polynomial f are not known. In [14], strong lower bounds were given for a non-monotone f. The non-monotonicity, however, is more than a subtlety. Their proof hinges on the fact that $(x-1)^2+2^{-2^{n+1}}$ can be written as f/g with f,g monotone, whereas they require degrees 2^{2^n}.

This question can be phrased more generally. If f can be computed by a monotone circuit with divisions of size s then we can find non-zero h and g with monotone circuit size $O(s)$ such that $fh = g$. In other words, f divides g.

▶ **Problem 2.** *Find an explicit monotone f_n (with polynomially many variables and of a polynomial degree) such that g requires superpolynomial monotone circuit whenever $g \neq 0$ and f_n divides g.*

A seminal result of Kaltofen [25], see also [5], states the following: if f of degree d can be computed by a circuit of size s, we can compute each factor of f by a (non-monotone) circuit of size polynomial in s and d. We believe that in fact d can be replaced by the degree of the factor. This means that in the non-monotone setting, Problem 2 is equivalent to proving lower bound on f_n.

Shadow complexity gives a partial solution to Problem 2.

▶ **Theorem 42.** *Let f be a (not necessarily monotone) real polynomial such that $\sigma(\mathsf{Newt}(f)) = k$. Assume that $g \neq 0$ is a monotone polynomial such that f divides g. Then every monotone formula computing g contains at least k leaves.*

Proof. Lemma 12 gives $\sigma(\mathsf{Newt}(g)) \geq \sigma(\mathsf{Newt}(f))$, and we can apply Theorem 1. ◀

Shadow complexity also provides lower bounds on monotone *circuit* complexity provided the degree is not too large. This is another partial solution to Problem 2.

▶ **Proposition 43.** *Let f be either Clique_n or Art_n. Let $g \neq 0$ be a monotone polynomial such that f divides g.*

1. *g requires monotone formula with 2^n leaves.*
2. *If g has degree $d \leq 2^{o(n^{\frac{1}{2}})}$, then g requires monotone circuit of size $2^{\Omega(n^{\frac{1}{2}})}$.*
3. *If $g = \alpha f$ with α a monomial of an arbitrary degree, then g requires monotone arithmetic circuit of size $\Omega(2^n)$.*

Proof. 1 follows from Theorem 42 and the fact that f is transparent (see Proposition 19 and Remark 20). Similarly, $\mathsf{Newt}(\alpha f)$ is merely a shift of $\mathsf{Newt}(f)$ and hence it remains transparent, which gives 3.

For 2 we use a result of Hyafil [22]: If g has a monotone circuit of size s, then it has a monotone formula of size $2^{O(\log s \log d + \log^2 d)}$. Part 1 completes the proof. ◀

The degree assumption in 2 is rather artificial. A monotone circuit with divisions can result in g with an exponential degree, as is the case in the circuit from [14] computing the spanning tree polynomial. Nevertheless, this yields lower bounds at least for monotone formulas with division.

[5] This holds for polynomials of low degree; the spanning tree polynomial indeed has this property.

▶ **Theorem 44.** *The polynomials* Clique_n *and* Art_n *require monotone formula with division of size* $2^{\Omega(n)}$*.*

Proof. Brent's [4] argument that formulas with division can be balanced implies that if f has monotone formula with divisions of size s, then $f = g/h$ where both g and h have monotone formulas of size polynomial in s. Proposition 43 part 1 completes the proof ◀

▶ Remark 45. Transparency is fragile. If f is transparent then f^2 is not necessarily so. In fact, if f is monotone then f^2 is *never* transparent unless $|\mathsf{supp}(f)| \leq 1$. Hence, the techniques from Proposition 43 do not give anything when $g = f^m$ and m is exponentially large.

▶ Remark 46. A different partial solution to Problem 2 can be inferred from monotone *Boolean* lower bounds. Let $\mathsf{Clique}_{k,n}$ be the polynomial $\sum_A \prod_{i,j\in A} x_{i,j}$, where A ranges over k-element subsets of $[n]$. For $k := \lfloor (n/\log n)^{2/3}/4 \rfloor$, and for every m, the polynomial $(\mathsf{Clique}_{k,n})^m$ requires a monotone arithmetic circuit of size $2^{n^{\Omega(1)}}$.

Indeed, a monotone arithmetic can be interpreted as a monotone Boolean circuit (cf. Section 5.1). Hence, a monotone arithmetic circuit for $(\mathsf{Clique}_{k,n})^m$ translates to a monotone Boolean circuit deciding whether a graph has a k-clique. This requires an exponential circuit by a result of Alon and Boppana [1].

7 τ-Conjecture for Newton polygons

Koiran et al. made the following conjecture [30].

▶ **Conjecture 47** ([30])**.** *Let* $\mathbb{F}$ *be a field. Let* $f \in \mathbb{F}[x_1, x_2]$ *be a bivariate polynomial which can be written as*

$$f = \sum_{i=1}^{p} \prod_{j=1}^{q} f_{i,j}\,, \quad \textit{where } |\mathsf{supp}(f_{i,j})| \leq r\,, \tag{7}$$

then $\mathsf{Newt}(f)$ *has at most* $O((pqr)^c)$ *vertices (for some absolute constant* c*).*

The authors of [30] have shown that Conjecture 47 implies VP≠VNP over the field in question. The conjecture is related to a similar conjecture by Koiran from [28] about the number of real roots of *univariate* polynomials. In [21], it was shown that the conjecture from [28] in fact implies Conjecture 47. Theorem 33 validates the conjecture in the monotone setting:

▶ Remark 48. Let f be as in (7) with f_{ij} monotone. Then $\mathsf{Newt}(f)$ has at most pqr vertices.

The conjecture can be used to upper-bound the shadow complexity.

▶ **Proposition 49.** *Let* $\mathbb{F}$ *be an infinite field. Assume Conjecture 47 holds over* $\mathbb{F}$*. Assume that a polynomial* f *of degree* d *has an arithmetic circuit of size* s*. Then* $\sigma(\mathsf{Newt}(f)) \leq s^{O(\sqrt{d}\log d)}$*.*

Proof. First, observe that if Conjecture 47 is true, it is also true when f and f_{ij} in (7) are allowed to be Laurent polynomials.

Now, if f has a circuit of size s, then f has a depth-four circuit of size $s^{O(\sqrt{d}\log d)}$; see [29] and references within. This means that we can write

$$f = \sum_{i=1}^{p} \prod_{j=1}^{q} f_{i,j}\,, \quad \text{where } |\mathsf{supp}(f_{i,j})| \leq r\,,$$

with $pqr \leq s^{O(\sqrt{d}\log d)}$.

Suppose that $\sigma(\mathsf{Newt}(f)) = k$. By Lemma 28, there is a h.p.-projection π so that the Newton polytope of the bivariate Laurent polynomial $\pi(f)$ has k vertices. Hence $\pi(f) = \sum_{i=1}^{p} \prod_{j=1}^{q} \pi(f_{i,j})$. Since $|\mathsf{supp}(\pi(f_{i,j}))| \leq r$, Conjecture 47 implies $k \leq O((pqr)^c)$ and hence $k \leq s^{O(\sqrt{d}\log d)}$. ◀

This gives quantitative bounds for some specific polytopes, mainly the Birkhoff polytope and the Matching polytope from Remark 26:

▶ **Corollary 50.** *Assume that Conjecture 47 holds over* some *infinite field. Then both* $\sigma(\mathsf{DS}_n)$ *and* $\sigma(\mathsf{MATCH}_n)$ *are at most* $2^{O(\sqrt{n}\log^2 n)}$.

Proof. DS_n is the Newton polytope of the determinant polynomial which has an arithmetic circuit of size $s = n^{O(1)}$. For MATCH_n, the same holds by Remark 36. ◀

We do not know whether these conclusions hold or not. Another implication of Conjecture 47 is that $\sigma(\mathsf{Q}_{k,n}) \leq n^{O(1)}$, where $\mathsf{Q}_{k,n}$ is the convex hull of vectors in $\{0,1\}^n$ of Hamming weight k. It follows from Proposition 15 that this is actually true: $\sigma(\mathsf{Q}_{k,n}) \leq n^2$.

▶ Remark 51. Results of Gritzman and Sturmfels [19] (cf. Section 1.1) imply the following monotone three-dimensional version. Let f be as in (7), where $f_{ij} \in \mathbb{R}[x_1, x_2, x_3]$ are monotone. Then $\mathsf{Newt}(f) \subseteq \mathbb{R}^3$ has at most $O(p(qr)^2)$ vertices.

8 An easy polynomial with many vertices

Here we construct a bivariate polynomial with a monotone arithmetic circuit of linear size, but whose Newton polytope has exponentially many vertices. This proves Theorem 30.

We use the following notation. Given $(a, b) \in \mathbb{R}^2$,

$$(a, b) \cdot P := \{(ax, by) : (x, y) \in P\}.$$

Given $a \in \mathbb{R}$,

$$aP := (a, a) \cdot P.$$

▶ **Observation 52.** *For a bivariate polynomial* $f(x, y)$,

$$\mathsf{Newt}(f(x^a, y^b)) = (a, b)\mathsf{Newt}(f(x, y)) \text{ and } \mathsf{Newt}(f^a) = a\mathsf{Newt}(f).$$

The building block of the polynomial are the following two polytopes. Let P_n be the polytope with vertices $\{(k, k^2) : 0 \leq k \leq n-1\}$. Let Q_n be the polytope with vertices $\{(k, k^2 + k) : 0 \leq k \leq n-1\}$. These polytopes can be constructed inductively as follows.

▶ **Lemma 53.** *For every* $n \geq 1$,

$$\begin{aligned} P_{2n} &= (2,4) \cdot P_n \sqcup ((1,1) + (2,4) \cdot Q_n)) && (8) \\ Q_{2n} &= (1,2) \cdot (P_n + Q_n) \sqcup \{(2n-1, 2n(2n-1))\}\,. && (9) \end{aligned}$$

Proof.

Part (8). Let $0 \leq k \leq 2n-1$. If $k = 2r$ is even then $r \leq n-1$ and

$$(k, k^2) = (2, 4)(r, r^2)$$

with (r, r^2) a vertex of P_n. If $k = 2r+1$ is odd then $r \leq n-1$ and

$$(k, k^2) = (2r+1, 4r^2 + 4r + 1) = (1, 1) + (2, 4) \cdot (r, r^2 + r)\,,$$

where $(r, r^2 + r)$ is a vertex of Q_n. This shows the containment $\subseteq$ in (8). The other direction holds since $P_n \sqcup Q_n$ can have at most $2n$ vertices.

Part (9). We first describe the vertices of $(1, 2)(P_n + Q_n)$. We claim that

$$\mathsf{vert}((1,2)(P_n{+}Q_n)) = \{v_0, v_1, \ldots, v_{2n-2}, u\}, \qquad (10)$$
$$\text{where } v_k := (k, k^2 + k),\ u := (n-1, 2n(n-1)).$$

Given $0 \leq k \leq 2n-2$, let us show that v_k is a vertex of $(1, 2)(P_n + Q_n)$. If $k = 2r$ is even, we have $r \leq n-1$ and

$$(k, k^2 + k) = (2r, 4r^2 + 2r) = (1, 2)(r, r^2) + (1, 2)(r, r^2 + r).$$

If $k = 2r + 1$ is odd, we have $r \leq n-2$ and

$$(k, k^2 + k) = (2r + 1, 4r^2 + 6r + 2) = (1, 2)(r + 1, (r + 1)^2) + (1, 2)(r, r^2 + r).$$

This means that $v_k \in (1, 2)(P_n + Q_n)$. Now, every $(z_1, z_2) \in (1, 2)(P_n + Q_n)$ satisfies $z_2 \geq z_1^2 + z_1$, because

$$2r_1^2 + 2(r_2^2 + r_2) - (r_1 + r_2)^2 - (r_1 + r_2) = (r_1 - r_2)^2 - (r_1 - r_2) \geq 0.$$

Since v_k lies on the curve $z_2 = z_1^2 + z_1$, and the curve is strictly convex, v_k cannot be convex combination of other points in $(1, 2)(P_n + Q_n)$. So, v_k is indeed a vertex. To show that u is a vertex, note that both $(1, 2)P_n$ and $(1, 2)Q_n$ are contained in the halfplane $\{(z_1, z_2) \in \mathbb{R}^2 : z_2 \leq 2nz_1\}$. On the boundary $z_2 = 2nz_1$, $(1, 2)Q_n$ has vertices $(0, 0)$ and u, and $(1, 2)P_n$ only the vertex $(0, 0)$. This implies u is a vertex of $(1, 2)(P_n + Q_n)$. This proves the containment $\subseteq$ in (10). Equality holds since $P_n + Q_n$ can have at most $2n$ vertices.

To infer (9) from (10), note that u lies on the line connecting the origin and $v_{2n-1} = (2n - 1, 2n(2n - 1))$. ◀

Proof of Theorem 30. Inductively define a sequence of bivariate polynomials. The base case is

$$p_0 = 1 \text{ and } q_0 = 1.$$

The inductive step is

$$p_{n+1} = p_n(x^2, y^4)^2 + x^N y^N q_n(x^2, y^4)^2$$

and

$$q_{n+1} = p_n(x^2, y^4)q_n(x^2, y^4) + x^{N(N-1)}y^{N^2(N-1)}$$

where $N = 2^{n+1}$.

We claim that for every $n \geq 0$,

$$\mathsf{Newt}(p_n) = 2^n P_{2^n} \text{ and } \mathsf{Newt}(q_n) = 2^n Q_{2^n}\,. \qquad (11)$$

For $n = 0$, this follows from $\mathsf{Newt}(p_0) = \mathsf{Newt}(q_0) = \{(0, 0)\} = P_1 = Q_1$. The induction step uses Lemma 11 and Observation 52. Assume that (11) holds for a given $n \geq 0$. Then

$$\mathsf{Newt}(p_n(x^2, y^4)) = 2^n(2, 4)P_{2^n} \text{ and } \mathsf{Newt}(q_n(x^2, y^4)) = 2^n(2, 4)Q_{2^n}.$$

Using (8),

$$\begin{aligned}\mathsf{Newt}(p_{n+1}) =& 2\cdot 2^n(2,4)P_{2^n} \sqcup ((N,N)+2\cdot 2^n(2,4)Q_{2^n}))\\ =&2^{n+1}((2,4)P_{2^n}\sqcup((1,1)+(2,4)Q_{2^n})))\\ =&2^{n+1}P_{2^{n+1}}\,.\end{aligned}$$

Similarly, part (9) gives

$$\begin{aligned}\mathsf{Newt}(q_{n+1}) =& 2^n(2,4)(P_{2^n}+Q_{2^n})\sqcup\{(N(N-1),N^2(N-1))\}\\ =&2^{n+1}\left((1,2)(P_{2^n}+Q_{2^n})\sqcup\{(N-1,N(N-1))\}\right)\\ =&2^{n+1}Q_{2^{n+1}}\,.\end{aligned}$$

This proves (11).

To compute p_n, q_n, first construct a circuit of size $O(n)$ that simultaneously computes $x^M, x^{M(M-1)}$, $y^M, y^{M^2(M-1)}$ for every $M = 2^m$ with $m \leq n$. Now, construct a circuit for p_n and q_n inductively. Given a circuit for p_n and q_n, we can construct a new one computing p_{n+1}, q_{n+1} by introducing a constant number of extra gates. ◀

9 Open problems

We conclude with the main open problems of this paper.

▶ **Open Problem 1.** *Is $\sigma(\mathsf{DS}_n)$ or $\sigma(\mathsf{MATCH}_n)$ exponential in n?*

▶ **Open Problem 2.** *Is Conjecture 47 true? If not, is it true when f in (7) is required to have convexly independent support?*

▶ **Open Problem 3.** *Find an explicit monotone f_n (with polynomially many variables and of a polynomial degree) such that g requires superpolynomial monotone arithmetic circuit whenever $g \neq 0$ and f_n divides g.*

References

1 Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

2 E. Balas. Disjunctive programming: properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44, 1998.

3 A. Borodin and S. Cook. On the number of additions needed to cumpute specific polynomial. *SIAM J. Comput.*, 5:146–157, 1976.

4 R. P. Brent. The parallel evaluation of general arithmetic expressions. *J. ACM*, 21:201–206, 1974.

5 P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and Computation in Mathematics.* Springer, 2000.

6 P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *A series of comprehensive studies in mathematics.* Springer, 1997.

7 P. Carstensen. Complexity of some parametric integer and network programming problems. *Math. Programming*, 26:64–75, 1983.

8 P. Carstensen. *The complexity of some problems in parametric linear and combinatorial programming.* PhD thesis, Univ. of Michigan, 1983.

9 B. Chazelle, H. Edelsbrunner, and L. J. Guibas. The complexity of cutting complexes. *Discrete Comput Geom*, 4:139–181, 1989.

10 M. Confronti, M. D. Summa, and Y. Faenza. Balas formulation for the union of polytopes is optimal. *Math. Programming*, 180:311–326, 2020.

11 M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer, 2 edition, 2000.

12 J. Edmonds. Matroids and the greedy algorithm. *Math. Programming* 1, pages 127–136, 1971.

13 Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Linear vs. semidefinite extended formulations: Exponential separation and strong lower bounds. *CoRR*, abs/1111.0837, `arXiv:1111.0837`, 2011.

14 S. Fomin, D. Grigoriev, and G. Koshevoy. Subtraction-free complexity, cluster transformations, and spanning trees. *Found Comput Math*, 16:1–31, 2016.

15 D. Gale. Optimal assignments in an ordered set: an application of matroid theory. *J. Combin. Theory* 4, pages 1073–1082, 1968.

16 S. Gao. Absolute irreducibility of polynomials via newton polytopes. *Journal of Algebra*, 237(2):501–520, 2001.

17 S.B. Gashkov. The complexity of monotone computations of polynomials. *Mosc. Univ. Math. Bull.*, 42(5):1–8, 1987.

18 S.B. Gashkov and I.S. Sergeev. A method for deriving lower bounds for the complexity of monotone arithmetic circuits computing real polynomials. *Sbornik: Mathematics*, 203(10):33–70.

19 P. Gritzmann and B. Sturmfels. Minkowski addition of polytopes: Computational complexity and applications to Gröbner bases. *SIAM J. Disc. Math.*, 6(2), 1993.

20 J. A. Grochow. Monotone projection lower bounds from extended formulation lower bounds. *Theory of Computing*, 13:1–15, 2017.

21 P. Hrubeš. On the distribution of runners on a circle. *European Journal of Combinatorics*, 89, 2020.

22 L. Hyafil. On the parallel evaluation of multivariate polynomials. *SIAM J. Comput.*, 8(2):120–123, 1979.

23 M. Jerrum and M. Snir. Some exact complexity results for straight-line computations over semirings. *Journal of the ACM*, 1982.

24 S. Jukna. Lower bounds for tropical circuits and dynamic programs. *Theory of Computing Systems*, 57:160–194, 2015.

25 E. Kaltofen. Uniform closure properties of p-computable functions. In *STOC*, pages 330–337, 1987.

26 O. M. Kasim-Zade. Arithmetic complexity of monotone polynomials. In *Theoretical Problems in Cybernetics. Abstracts of lectures.* Saratov State University Publishing House, Saratov, 1986.

27 V. Klee. On a conjecture of Lindenstrauss. *Israel Journal of Mathematics*, 1:1–4, 1963.

28 P. Koiran. Shallow circuits with high-powered inputs. In *Symposium on Innovations in Computer Science*. Tsingua University Press, Beijing, 2011.

29 P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012.

30 P. Koiran, N. Portier, S. Tavenas, and S. Thomassé. A τ-conjecture for Newton polygons. *Foundations of computational mathematics*, 15(1):187–197, 2015.

31 U. H. Kortenkamp, J. Richter-Gebert, A. Sarangajan, and G. M. Ziegler. Extremal properties of 0/1-polytopes. *Discrete and Computational Geometry*, 17:439–448, 1997.

32 S. E. Kuznetsov. Monotone computations of polynomials without zero chains. In *VII All-Union Conference on Problems in Theoretical Cybernetics*, pages 108–109. Irkutsk, 1985.

33 J. G. Lagarias, Y. Luo, and A. Padrol. Moser's shadow problem. *ArXiv*, `arXiv:1310.4345`, 2013.

34 E. H. Moore. A two-fold generalization of fermat's theorem. *Bull. Amer. Math. Soc.*, 2(7):189–199, 1896.

35 L. Moser. Poorly formulated unsolved problems in combinatorial geometry. In *mimeographed notes.* (East Lansing conference), 1966.

36 K. Mulmuley. Lower bounds in a parallel model without bit operations. *SIAM J. Comput.*, 28(4):1460–1509, 1999.

37 K. Mulmuley and P. Shah. A lower bound for the shortest path problem. *Journal of Computer and System Sciences*, 62(2):253–267, 2001.

38 A. Rao and A. Yehudayoff. Communication Complexity: And Applications. Cambridge University Press. doi:10.1017/9781108671644

39 R. Raz and A. Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. J. Comput. Syst. Sci. 77(1), pages 167–190, 2011.

40 Thomas Rothvoß. Some 0/1 polytopes need exponential size extended formulations. *CoRR*, abs/1105.0036, arXiv:1105.0036, 2011.

41 Thomas Rothvoß. The matching polytope has exponential extension complexity the matching polytope has exponential extension complexity. *J. ACM*, 2017.

42 H. J. Ryser. *Combinatorial Mathematics.* Mathematical Association of America, 1963.

43 E. Shamir and M. Snir. On the depth complexity of formulas. *Journal Theory of Computing Systems*, 13(1):301–322, 1979.

44 A. Shpilka and A. Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science,* 5(3-4), 2010.

45 L. G. Valiant. Negation can be exponentially powerful. *Theoretical Computer Science*, 12:303–314, 1980.

46 A. Vince. A framework for the greedy algorithm. *Discrete Applied Mathematics* 121, pages 247–260, 2002.

47 Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.

Fractional Pseudorandom Generators from Any Fourier Level

Eshan Chattopadhyay ✉
Department of Computer Science, Cornell University, Ithaca, NY, USA

Jason Gaitonde ✉
Department of Computer Science, Cornell University, Ithaca, NY, USA

Chin Ho Lee ✉
Department of Computer Science, Columbia University, New York City, NY, USA

Shachar Lovett ✉
Department of Computer Science, University of California, San Diego, CA, USA

Abhishek Shetty ✉
Department of Computer Science, University of California, Berkeley, CA, USA

Abstract

We prove new results on the polarizing random walk framework introduced in recent works of Chattopadhyay et al. [4, 6] that exploit L_1 Fourier tail bounds for classes of Boolean functions to construct pseudorandom generators (PRGs). We show that given a bound on the k-th level of the Fourier spectrum, one can construct a PRG with a seed length whose quality scales with k. This interpolates previous works, which either require Fourier bounds on all levels [4], or have polynomial dependence on the error parameter in the seed length [6], and thus answers an open question in [6]. As an example, we show that for polynomial error, Fourier bounds on the first $O(\log n)$ levels is sufficient to recover the seed length in [4], which requires bounds on the entire tail.

We obtain our results by an alternate analysis of fractional PRGs using Taylor's theorem and bounding the degree-k Lagrange remainder term using multilinearity and random restrictions. Interestingly, our analysis relies only on the *level-k unsigned Fourier sum*, which is potentially a much smaller quantity than the L_1 notion in previous works. By generalizing a connection established in [5], we give a new reduction from constructing PRGs to proving correlation bounds. Finally, using these improvements we show how to obtain a PRG for $\mathbb{F}_2$ polynomials with seed length close to the state-of-the-art construction due to Viola [26].

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Derandomization, pseudorandomness, pseudorandom generators, Fourier analysis

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.10

Funding *Eshan Chattopadhyay*: Supported by NSF grant CCF-1849899 and NSF CAREER award 2045576.
Jason Gaitonde: Supported by NSF grant CCF-1408673 and AFOSR grant FA9550-19-1-0183.
Chin Ho Lee: Supported by the Croucher Foundation and the Simons Collaboration on Algorithms and Geometry.
Shachar Lovett: Supprted by NSF grants CCF-2006443 and DMS-1953928.
Abhishek Shetty: Supported by a JP Morgan Chase Faculty Fellowship.

Acknowledgements We thank the anonymous reviewers for their helpful comments and suggestions.

COMPUTATIONAL
COMPLEXITY
CONFERENCE

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 10; pp. 10:1–10:24
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A central pursuit in complexity theory is to understand the need of randomness in efficient computation. Indeed there are important conjectures (such as $\mathbf{P} = \mathbf{BPP}$) in complexity theory which state that one can completely remove the use of randomness without losing much in efficiency. While we are quite far from proving such results, a rich line of work has focused on *derandomizing* simpler models of computation (see [25] for a survey of prior work on derandomization). A key tool for proving such derandomization results is through the notion of a *pseudorandom generator* defined as follows.

▶ **Definition 1.** *Let $\mathcal{F}$ be a class of n-variate Boolean functions. A* pseudorandom generator *(PRG) for $\mathcal{F}$ with error $\varepsilon > 0$ is a random variable $\mathbf{X} \in \{-1,1\}^n$ such that for all $f \in \mathcal{F}$,*

$$\left|\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] - \mathbb{E}_{\mathbf{U}_n}[f(\mathbf{U}_n)]\right| \leq \varepsilon,$$

where $\mathbf{U}_n$ is the uniform distribution on $\{-1,1\}^n$. We also say that $\mathbf{X}$ fools *$\mathcal{F}$ with error ε. If $\mathbf{X} = G(\mathbf{U}_s)$ for some explicit function $G : \{-1,1\}^s \to \{-1,1\}^n$, then $\mathbf{X}$ has* seed length *s.*

There is a long line of research on explicit constructions of PRGs for various classes of Boolean functions in the literature and it is well beyond our scope to survey prior work here. We focus on a recent line of works initiated by Chattopadhyay et al. [4, 6] that provide a framework for constructing pseudorandom generators for any Boolean function classes that exhibit *Fourier tail bounds* (we will define and discuss this in more details in the next subsection; see Section 2.1 for a brief introduction to Fourier analysis of Boolean functions). This provides a unified PRG for several well-studied function classes such as small-depth circuits, low-sensitivity functions, and read-once branching programs that exhibit such Fourier tails.

We now briefly discuss this new framework, and then in Section 1.2 we present our new results, which significantly generalize this approach.

1.1 The Polarizing Random Walk Framework

The *polarizing random walk* framework was introduced by Chattopadhyay, Hatami, Hosseini, and Lovett [4]. The authors showed that for any class of n-variate Boolean functions that is closed under restrictions, one can flexibly construct pseudorandom generators via the following local-to-global principle: it suffices to construct *fractional pseudorandom generators (fractional PRGs)*, a notion that generalizes PRGs to allow the random variable $\mathbf{X}$ (in Definition 1) to be supported on the solid cube $[-1,1]^n$ instead of $\{-1,1\}^n$, while still requiring that $\mathbf{X}$ fools (the multilinear extension) of each Boolean function in the class. Ideally, the variance of each coordinate of $\mathbf{X}$ should be as large as possible. Towards this, we define a fractional PRG $\mathbf{X}$ to be p-noticeable if the variance in each of its coordinates is least p (See Definition 13 for a formal definition of a fractional PRG).

To obtain a genuine pseudorandom generator from a fractional PRG, the authors give a random walk gadget that composes together independent copies of the fractional PRG in a random walk that polarizes $\mathbf{X}$ quickly to take values from the Boolean hypercube $\{-1,1\}^n$. The analysis for how the error accumulates in this process relies on interpreting the intermediate points of $\mathbf{X}$ in this random walk as an average of *random restrictions* of the original Boolean function. As the fractional PRG locally fools the class by definition, this analysis shows that the random walk does not incur much additional error at each intermediate step and the rapid polarization shows that it does not take too many steps. Taken together, these two facts imply that the final random variable (supported on $\{-1,1\}^n$) successfully fools the class.

Through this construction, the design of pseudorandom generators reduces to the easier task of designing fractional pseudorandom generators. It is easier as such random variables need not be Boolean-valued. The authors further construct such fractional pseudorandom generators for any class of functions satisfying *Fourier tail bounds*, that is, every function in the class is such that the L_1 Fourier mass at each level $1 \leq k \leq n$ is at most b^k for some fixed $b \geq 1$. For error ε, their fractional pseudorandom generators have seed length $O(\log\log n + \log(1/\varepsilon))$ and variance $\Theta(b^{-2})$ in each coordinate. Combining this fractional pseudorandom generator with their random walk gadget yields a pseudorandom generator with seed length $b^2 \cdot \text{polylog}(n/\varepsilon)$ for *any* class with such Fourier tail bounds.

As a result, if one can show that a function class admits nontrivial Fourier tail bounds (and is closed under restriction), then the construction in [4] immediately implies a pseudorandom generator for this class. Some examples of Boolean functions that exhibit such tail bounds include $\mathbf{AC}^0$ circuits with the parameter $b = \text{poly}(\log n)$ [13, 23], constant width read-once branching programs with $b = \text{poly}(\log n)$ [7], s-sensitive functions with $b = O(s)$ [11, 10], and product tests [12]. Using these tail bounds, [4] immediately gave PRGs for these function classes. It was also conjectured in [4] that the class of n-variate degree-d polynomials over $\mathbb{F}_2$ satisfy such tail bounds. We discuss this in more detail in Section 1.2.

A natural question is whether the complete control on the entire Fourier tail of a class is necessary to obtain a PRG in this framework. In the subsequent work by Chattopadhyay, Hatami, Lovett, and Tal [6], the authors show how to construct fractional pseudorandom generators using different pseudorandom primitives whose seed length depends on just the *second Fourier level* of the class. They construct their fractional PRGs by derandomizing the celebrated work of Raz and Tal [18], which establishes an oracle separation of **BQP** and **PH**. Raz and Tal show that classes of multilinear functions with small level-two Fourier mass cannot significantly distinguish between a suitable variant of the Forrelation distribution and the uniform distribution.[1] However, this construction incurs exponentially worse dependence on the error parameter in each fractional step to sample sufficiently good approximations to Gaussian random variables. The final seed length given by this construction has the form $O((b^2/\varepsilon)^{2+o(1)}\text{polylog}(n))$, where b^2 is the level-two Fourier mass of the class. This yields exponentially worse dependence on the error compared to the generator of [4], as well as quadratically worse dependence on the level-two mass (though without assumptions on the rest of the Fourier levels).

1.2 Our Contribution

In this paper, we address several open questions in this framework by leveraging a novel connection between polarizing random walk and the classical theory of polynomial approximation. Given these prior works, a very natural question (also explicitly asked in [6]) is whether it is possible to interpolate between these previous constructions by assuming Fourier bounds on an intermediate level. Concretely, can this framework still succeed if one has Fourier control at just level k? If the class further has such Fourier bounds up to and including level k, can one interpolate between the seed lengths of [4] and [6]? Given Fourier bounds from level 1 up to level k, what range of error $\varepsilon > 0$ can the resulting PRG tolerate while maintaining polylogarithmic dependence on $1/\varepsilon$ in the seed length (or equivalently, given a desired error $\varepsilon > 0$, how many levels of Fourier bounds are sufficient to ensure that the seed length remains polylogarithmic in $1/\varepsilon$)?

[1] It turns out that this fact can be interpreted via Itô's Lemma, which shows that the local behavior of a smooth function of Brownian motion is essentially determined by the first two derivatives [28].

Moreover, it was previously not known whether L_1 control of Fourier tails is really necessary for this framework to yield effective PRGs, or whether weaker Fourier quantities would suffice. To this end, define

$$L_{1,k}(f) \triangleq \sum_{S\subseteq[n]:|S|=k} |\hat{f}(S)|$$

to be the *level-k L_1 Fourier mass* of f, and

$$M_k(f) \triangleq \max_{\mathbf{x}\in[-1,1]^n} \left| \sum_{S\subseteq[n]:|S|=k} \hat{f}(S)\mathbf{x}^S \right| = \max_{\mathbf{x}\in\{-1,1\}^n} \left| \sum_{S\subseteq[n]:|S|=k} \hat{f}(S)\mathbf{x}^S \right|.$$

to be the *level-k absolute Fourier sum* of f. For a function class $\mathcal{F}$, we define $L_{1,k}(\mathcal{F})$ and $M_k(\mathcal{F})$ as the maximum of $L_{1,k}(f)$ and $M_k(f)$ taken over $f \in \mathcal{F}$. The recent work by Chattopadhyay, Hatami, Hosseini, Lovett, and Zuckerman [5] considers the weaker quantity of the level-two *unsigned Fourier sum*, defined as the absolute value of the sum of the Fourier coefficients rather than the sum of their absolute values that is considered in [4, 6]. The authors show that the problem of bounding the level-two unsigned Fourier sum corresponds to the problem of bounding the covariance of the function class and the XOR of shifted majority functions. For a class that is closed under negations of the variables, the level-two unsigned Fourier sum is precisely the quantity $M_2(\mathcal{F})$. In particular, using this connection to this weaker object, the authors explicitly ask whether bounding the weaker Fourier quantity $M_2(\mathcal{F})$ (or more generally, $M_k(\mathcal{F})$) yields pseudorandom generators.

In this work, we positively resolve all of these questions. To do so, we establish novel connections between the polarizing random walk framework and the classical theory of polynomial approximations of Boolean functions. We show that the seed length of a fractional PRG for a given class of functions $\mathcal{F}$ is intimately connected to the uniform error of low-degree approximations of functions on *subcubes* of the form $[-c, c]^n$ for some $c < 1$.

Our main technical result provides an upper bound on this quantity in terms of $M_k(\mathcal{F})$ for every function f in a class $\mathcal{F}$ that is closed under restrictions. For any multilinear polynomial $f : \{-1, 1\}^n \to \mathbb{R}$, define $f_{\geq k}$ to be component of f with monomials of degree at least k. Then our main result asserts the following bound:

▶ **Theorem 2.** *Let $f \in \mathcal{F}$ with $\mathcal{F}$ closed under restrictions. Then for all $c \in (0, 1)$, we have*

$$\max_{\mathbf{x}\in[-c,c]^n} |f_{\geq k}(\mathbf{x})| \leq \left(\frac{c}{1-c}\right)^k M_k(\mathcal{F}).$$

For intuition, recall that by Parseval's identity in Fourier analysis the low-degree Fourier expansion of any Boolean function f is provably the best ℓ_2-approximator on $\{-1, 1\}^n$. Conversely, from elementary analysis, one can show that the best uniform (i.e. ℓ_∞) low-degree approximators of f converge, coefficient-by-coefficient, to the low-degree expansion of f as the domain converges to $\mathbf{0}$. Our main result shows that one can strongly quantify the ℓ_∞ error of the low-degree approximator of Boolean functions on subcubes so long as c is not too close to 1 (compare this bound to when f has degree exactly k).

We complement this result with a corresponding lower bound on the best attainable uniform error for *any* low-degree approximation on these subcubes that will be comparable for sufficiently small values of c (see Theorem 23). These results combined together imply that the low-order expansion of a Boolean function is a reasonable uniform approximation for small domains. Note that the properties of low-degree approximations on subcubes with $c \ll 1$ can be quite different than for $c = 1$; for instance the PARITY function on n bits is well-known to be inapproximable on $\{-1, 1\}^n$ to constant error unless the approximating polynomial has degree $\Omega(n)$, but is trivially approximable for any c bounded away from 1.

From this main result, we can positively resolve the above open questions in the polarizing random walk framework as a nearly immediate corollary. To do so, we provide a new analysis of the fractional pseudorandom generator of [4] that views fractional pseudorandom generators as fooling a low-degree part of a function on $[-c, c]^n$ for some $c < 1$, where the high-degree part has small ℓ_∞ norm on $[-c, c]^n$. Recall that the seed length of the final generator depends on the variance of the constituent fractional generator; the connection to the above result is that for a given error ε, the largest subcube on which the above approximation holds can be lower-bounded using just the weaker $M_k(\mathcal{F})$ quantity. Leveraging this insight, our main result in the polarizing random walk framework is the following analysis of a fractional pseudorandom generator:

▶ **Theorem 3.** *Let $\mathcal{F}$ be any class of n-variate Boolean functions that is closed under restrictions. Suppose $M_k(\mathcal{F}) \le b^k$ for some $b \ge 1$ and $k \ge 1$. Then for any $\varepsilon > 0$, there exists an explicit $\Omega(\varepsilon^{2/k}/b^2)$-noticeable fractional PRG for $\mathcal{F}$ with error ε and seed length $O(k \cdot \log n)$.*[2]

Further, if it holds that $L_{1,i}(\mathcal{F}) \le b^i$ for all $1 \le i < k$, then the seed length can be improved to $O(\log\log n + \log k + \log(1/\varepsilon))$.

Using the fractional pseudorandom generator from Theorem 3, we obtain the following consequences almost immediately from the random walk gadget of [4] (see Theorem 14):

1. **Pseudorandom Generators from Fourier Bounds at Level k**: From our fractional pseudorandom generator, we show that the random walk framework yields nontrivial pseudorandom generators assuming Fourier bounds *just at* level k of the associated class, with improvements if we assume bounds from level 1 *up to* level k. The informal statement is the following:

   ▶ **Theorem 4.** *Let $\mathcal{F}$ be any class of n-variate Boolean functions that is closed under restrictions. Suppose that $\mathcal{F}$ satisfies $M_k(\mathcal{F}) \le b^k$ for some $b \ge 1$ and $k \ge 3$. Then there exists an explicit pseudorandom generator for $\mathcal{F}$ for error ε with seed length $k \cdot b^{2+4/(k-2)}\mathrm{polylog}(n/\varepsilon)/\varepsilon^{2/(k-2)}$. The seed length can be improved if $L_{1,i}(\mathcal{F}) \le b^i$ for all levels $i \le k$.*

   See Theorem 27 for the precise statement. One immediate consequence is that if one has a non-trivial bound on $M_3(\mathcal{F})$, then the seed length of our PRG has the same dependence on the error ε as the one in [6]. Further, given $M_4(\mathcal{F}) \le b^4$, one obtains better seed length than [6]; in particular it has quadratically better dependence on $1/\varepsilon$ in the seed length (as well as polylogarithmic factors in n/ε). More generally, given an appropriate Fourier bound of b^k on just some level $k \le \mathrm{polylog}(n)$, one obtains a pseudorandom generator with error ε with seed length $O(b^{2+4/(k-2)}\mathrm{polylog}(n/\varepsilon)/\varepsilon^{2/(k-2)})$.

   We note that the fractional PRG from Theorem 3 cannot be converted into a PRG for $k = 1, 2$. Informally, this is because of the following reason: the number of steps one needs to take in the random walk gadget of [4] (with each step using an independent copy of the fractional PRG) scales roughly with the variance of the fractional PRG, and the error adds up in each step. As is clear from Theorem 3, for the variance of the fractional PRG to scale sublinearly with the error, one requires $k > 2$. See Remark 28 for more discussion.

[2] We remark that at this level of generality, this linear dependence on k is essentially necessary. Indeed, any Boolean function on n-variables has L_1 level-n mass at most 1, but one cannot hope to generically fool all Boolean functions simultaneously without using n bits.

2. **Pseudorandom Generators with Polylogarithmic Error Dependence from Up-to-level-k Bounds**: A simple corollary of our fractional pseudorandom generator is that one can recover the polylogarithmic dependence on $1/\varepsilon$ from [4] if $\varepsilon \geq b \cdot \log n \cdot 2^{-O(k)}$ and we have Fourier bounds *up to* level k.

   ▶ **Corollary 5.** *Let $\mathcal{F}$ be any class of n-variate Boolean functions that is closed under restrictions. Suppose that for some level $k \geq 3$ and $b \geq 1$, we have $M_k(\mathcal{F}) \leq b^k$ and $L_{1,i}(\mathcal{F}) \leq b^i$ for $i < k$. Then, for any $\varepsilon \geq b \cdot \log n \cdot 2^{-O(k)}$, there exists an explicit pseudorandom generator for $\mathcal{F}$ with error ε and seed length $O(b^2 \text{polylog}(n/\varepsilon))$.*

   This actually subsumes the analysis of [4] without requiring anything on the full Fourier tail, and addresses an open question of [6] asking how many levels of Fourier bounds one needs control of to regain polylogarithmic dependence on ε. In particular, if one requires error $\varepsilon = 1/\text{poly}(n)$, then it suffices to have Fourier bounds up to level $\Theta(\log n)$ to get the same dependence.

We view this work as a proof of concept that it is indeed possible to interpolate between the two extremes of [4, 6] in the polarizing random walk framework and obtain better results using weakened Fourier assumptions. We prove Theorem 3 in Section 4, from which Theorem 4 and Corollary 5 follow without much difficulty using the existing random walk gadget of [4].

Note that for some Boolean classes of great interest such as the class of low-degree $\mathbb{F}_2$-polynomials, Fourier tail bounds as required by [4] are not yet known and thus Theorem 3 allows us to leverage potentially much weaker bounds proved in [4] to construct a PRG with polylogarithmic dependence on n/ε in the seed length (see Theorem 6). This almost matches the best known PRG due to Viola [26]. In particular, we show the following:

▶ **Theorem 6.** *Let $\mathcal{F}$ be the class of degree-d polynomials over $\mathbb{F}_2$ on n variables. Then there exists an explicit pseudorandom generator for $\mathcal{F}$ with error ε and seed length $2^{O(d)}\text{polylog}(n/\varepsilon)$.*

We present the proof of Theorem 6 in Section 5. While this result does not quite match the current state-of-the-art PRG for this class due to Viola [26] (and therefore fails to give anything nontrivial for $d = \Omega(\log n)$), we view this as a conceptual contribution that the random walk framework can yield an explicit pseudorandom generator with seed length that is polylogarithmic in n/ε, which was not known from previous works [4, 6]. As we discuss below, the results in [4, 6] do not give a PRG for the class of $\mathbb{F}_2$-polynomials with polylogarithmic error dependence using known Fourier tail bounds.

As a concrete application of this approach which would dramatically improve the state-of-the-art PRGs for $\mathbb{F}_2$-polynomials, both [4] and [6] conjecture Fourier bounds on the L_1 mass of the class of degree-d $\mathbb{F}_2$ polynomials. The former conjectures that this class satisfies a tail bound of the form c_d^k for some constant c_d at all levels $1 \leq k \leq n$ (so as to apply their approach), while the latter conjectures just that the level-two L_1 mass is $O(d^2)$. While neither conjecture seems close to being resolved, our work shows that one can instead prove bounds for the smaller quantities $M_k(\mathcal{F})$ for any $k \geq 3$. If one could prove such bounds of the form $(\text{poly}(d, \log n))^k$ for some level $k = \Omega(1)$, or even more optimistically, for some $k = \Omega(\log n)$, this would immediately imply a breakthrough pseudorandom generator for $\mathbf{AC^0}[\oplus]$ using the results of Razborov [19] and Smolensky [21, 22] (see the discussion in [6]).

To our knowledge, our application of $M_k(\mathcal{F})$ bounds is new to the pseudorandomness literature. There are several advantages to proving $M_k(\mathcal{F})$ bounds over $L_{1,k}(\mathcal{F})$ bounds. For one, from the definition we clearly have $M_k(\mathcal{F}) \leq L_{1,k}(\mathcal{F})$ for any class $\mathcal{F}$. This improvement alone potentially gives smaller seed length for any class. From an analytical perspective, we believe that the quantity $M_k(\mathcal{F})$ is easier to estimate. Specifically, for a class $\mathcal{F}$ that is closed under negation of input variables, $M_k(\mathcal{F})$ is precisely an *unsigned Fourier sum* and can be bounded via the recent connections established by Chattopadhyay et al. [5], which reduces $M_2(\mathcal{F})$ bounds to proving correlation bounds against certain resilient functions. We straightforwardly generalize their reduction to $M_k(\mathcal{F})$ bounds in Section 6.

1.3 Overview of Our Approach

To prove Theorem 2, we rely on Taylor's theorem, as well as multilinearity and the random restriction trick of [4]. Recall that Taylor's theorem, when applied to a sufficiently smooth function $h\colon [-1,1] \to \mathbb{R}$, asserts that the Taylor expansion at 0 can be expressed in terms of its first $(k-1)$-th order derivatives at 0 along with a Lagrange error term that depends on its k-th order derivatives at some intermediate point in our domain. In doing so, the higher-order components of the function "collapse" down to the k-th order term. While Taylor's theorem has been extensively applied in the construction of pseudorandom generators, often in tandem with *invariance principles*, we somewhat counterintuitively apply it to the *multilinear expansion of the Boolean functions* themselves.

To apply Taylor's theorem here, we consider one-dimensional restrictions of (the multilinear extension) of a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$. While the full Taylor expansion of a polynomial is trivially the same polynomial, the Lagrange error term eliminates the dependence on the high order Fourier coefficients (corresponding to the terms of degree $> k$). Moreover, the low-order terms of the Taylor expansion of f at 0 are precisely the original low-degree part of its Fourier expansion. However, the Lagrange error term requires the derivatives to be evaluated at a point away from 0. While the derivatives of f at a nonzero point are related to the *biased* Fourier coefficients of f, it is not clear how to estimate these quantities. To overcome this difficulty, recall that we are interested in bounds on $|f_{\geq k}(\mathbf{x})|$ for $\mathbf{x} \in \{-c,c\}^n$ where $c < 1$. In Lemma 22, we show that by "recentering" $\mathbf{x}$ using the random restriction technique of [4], we can write the error term as an average of the k-th order derivatives *at 0* of some random restrictions of our original function f, up to a multiplicative factor depending on c.[3] We can then apply multilinearity to bound these error terms using $M_k(\mathcal{F})$ to obtain Theorem 17.

While Theorem 17 shows that the low-order Taylor expansion of a Boolean function is a decent *uniform* approximator on subcubes $[-c,c]^n$ for some sufficiently small c that depends on the class $\mathcal{F}$, it is natural to wonder if one can obtain a better low-order approximation. Using our upper bound along with Chebyshev polynomials on the univariate restrictions, we give a lower bound showing that no low-order approximator can give significantly smaller error over $[-c,c]^n$ for any c less than some quantity depending on the ratio $M_k(\mathcal{F})/M_{k+1}(\mathcal{F})$ for some k. This quantifies the intuition that the low-degree Fourier expansion is a near optimal uniform approximator of f over small enough neighborhoods of $\mathbf{0}$. These arguments are formally carried out in Section 3.

To prove our results in the polarizing random walk framework, we rely on an alternate, simple analysis of fractional pseudorandom generators. The original analysis in [4] assumes control of $L_{1,k}(\mathcal{F})$ at all levels of the Fourier spectrum. We now explain how these assumptions can be weakened using Theorem 17. Consider a candidate fractional PRG $\mathbf{X} \in [-1,1]^n$. We first decompose the multilinear (Fourier) expansion of $f \in \mathcal{F}$ in the same manner as [4]:

$$\left|\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] - \mathbb{E}_{\mathbf{U}}[f(\mathbf{U})]\right| \leq \underbrace{\sum_{i=1}^{k-1} \sum_{S\subseteq[n]:|S|=i} \left|\hat{f}(S)\right| \left|\mathbb{E}_{\mathbf{X}}[\mathbf{X}^S]\right|}_{\text{low-order terms}} + \underbrace{\left|\mathbb{E}_{\mathbf{X}}[f_{\geq k}(\mathbf{X})]\right|}_{\text{high-order term}}. \tag{1}$$

[3] We note that similar ideas for the $k = 1$ case also appeared in [1] (attributed to Avishay Tal).

[4] requires bounding $L_{1,\ell}(\mathcal{F})$ for all $\ell \geq k$ to give a uniform bound on the high-order term. Using Theorem 17, we can obtain small error in the high-order term so long as we choose $\mathbf{X} \in [-c, c]^n$ for sufficiently small c depending on ε and $M_k(\mathcal{F})$. To handle the low-order terms, we consider two cases: if we further have $L_{1,\ell}(\mathcal{F})$ bounds for $\ell < k$, then we may choose $\mathbf{X}$ to be a scaled $(k-1)$-wise δ-biased distribution to nearly fool each of the low-order terms as in [4]. Otherwise, we may choose $\mathbf{X}$ to be a scaled $(k-1)$-wise independent distribution to incur zero error from the low-order terms. Note that the latter pseudorandom primitives are more expensive in terms of seed length. Finally, to obtain pseudorandom generators, we then simply apply the random walk gadget of [4] to our fractional PRGs as a blackbox. We refer the reader to Section 4 for formal proofs of the ideas in this section.

We immediately leverage this newfound flexibility to construct new pseudorandom generators for $\mathbb{F}_2$-polynomials of degree $d = O(\log n)$. We do this using known $L_{1,k}(\mathcal{F})$ bounds derived in [4]. Previously these bounds were not sufficient to give PRGs with polylogarithmic error dependence as their analysis of fractional PRGs either required control of the entire Fourier tail or could not leverage higher Fourier levels, but they can be employed here due to our more flexible analysis. This result is given in Section 5. Finally, we show how $M_k(\mathcal{F})$ bounds can be obtained using correlation bounds with shifted majority functions in Section 6. This is done by straightforwardly generalizing the analysis of [5], which shows how such correlation bounds can be used to bound the bulk of the terms in the definition of $M_k(\mathcal{F})$.

1.4 Other Related Work

To our knowledge, our use of $M_k(\mathcal{F})$ bounds is new to the derandomization literature. As mentioned earlier, the stronger and better-known $L_{1,k}(\mathcal{F})$ notion has been extensively studied in recent years. In addition to derandomization, a recent line of work [24, 3, 20] has used $L_{1,k}$ bounds for decision trees to obtain an optimal separation of quantum and classical query complexity. Among these works, the work of Bansal and Sinha [3] generalizes the results of Raz and Tal [18] by considering a k-generalization of their Forrelation distribution and bounding the distinguishing advantage of any function with small $L_{1,\ell}$ bounds for $\ell = 1, \ldots, k$. Much as how the results of Chattopadhyay et al. [6] derandomize the result of Raz and Tal, we believe that their construction can be derandomized for pseudorandomness purposes, but appears to give significantly worse seed length, nor obtains bounds in terms of $M_k(\mathcal{F})$. A related work by Girish, Raz, and Zhan [9] establishes a similar result with a different generalization of the Forrelation distribution, but we do not know how to use their construction for pseudorandom generators.

The relationship between $M_k(\mathcal{F})$ and $L_{1,k}(\mathcal{F})$ has been of intense study in the mathematics literature due to renewed interest in *Bohnenblust–Hille* inequalities (see, for instance, the breakthrough work of Defant, Frerick, Ortega-Cerdà, Ounaïes, and Seip [8]). The optimal constant $C_{n,k}$ satisfying $L_{1,k}(f) \leq C_{n,k} M_k(f)$ for any polynomial $f \colon \mathbb{C}^n \to \mathbb{C}$ is known as the *Sidon constant.* It is known that $C_{n,k}$ is, up to small exponential factors in k, proportional to roughly $n^{\frac{k-1}{2}}$, and its tightness is witnessed by a random function with high probability. The quantity $M_k(\mathcal{F})$ also has applications in other areas in theoretical computer science, such as quantum information theory (see for instance the survey of Montanaro [14]) and Boolean function analysis [2].

Subsequent to our work, Viola [27] observed that $M_k(\mathcal{F})$ bounds imply correlation bounds between $\mathcal{F}$ and an explicit function.

2 Preliminaries

As in [4] and [6], we study PRGs for classes $\mathcal{F}$ of n-variate Boolean functions that are closed under restriction (that is, fixing any subset of the input variables of a function in the class yields a function that remains in the class).

2.1 Fourier Analysis

We briefly recall basic Fourier analysis: any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ admits a unique multilinear expansion, also known as the *Fourier expansion*, given by

$$f(\mathbf{x}) = \sum_{S\subseteq[n]} \hat{f}(S)\mathbf{x}^S, \tag{2}$$

where we write $\mathbf{x}^S \triangleq \prod_{i\in S} x_i$. The Fourier coefficient $\hat{f}(S)$ is given by

$$\hat{f}(S) = \mathbb{E}_{\mathbf{X}\sim\{-1,1\}^n}[f(\mathbf{X})\mathbf{X}^S].$$

For more on Fourier analysis of Boolean functions, see the excellent book by O'Donnell [16]. One may thus extend the domain of f to $[-1,1]^n$, where $f(\mathbf{x})$ for arbitrary $\mathbf{x}$ is evaluated according to the expression in Equation (2). Note that in this case, $f(\mathbf{0}) = \hat{f}(\emptyset) = \mathbb{E}_{\mathbf{U}_n}[f(\mathbf{U}_n)]$. One of the main parameters of interest from the Fourier expansion for this framework is the following:

▶ **Definition 7.** *The* level-k mass of a Boolean function f *is*

$$L_{1,k}(f) \triangleq \sum_{S\subseteq[n]:|S|=k} |\hat{f}(S)|,$$

and the level-k mass of a class $\mathcal{F}$ *is* $L_{1,k}(\mathcal{F}) \triangleq \max_{f\in\mathcal{F}} L_{1,k}(f)$.

In this work, we will show how to construct PRGs whose seed length depends on the following, smaller quantity:

▶ **Definition 8.** *For any multilinear polynomial $f : \mathbb{R}^n \to \mathbb{R}$ given by $f(\mathbf{x}) = \sum_{S\subseteq[n]} \hat{f}(S)\mathbf{x}^S$, define the level-$k$ part by*

$$f_k(\mathbf{x}) \triangleq \sum_{S\subseteq[n]:|S|=k} \hat{f}(S)\mathbf{x}^S,$$

and further define $f_{<k}(\mathbf{x}) \triangleq \sum_{i=0}^{k-1} f_i(\mathbf{x})$ and $f_{\geq k}(\mathbf{x}) \triangleq \sum_{i=k}^{n} f_i(\mathbf{x})$. Then we define the level-k absolute Fourier sum of f *by*

$$M_k(f) \triangleq \max_{\mathbf{x}\in[-1,1]^n} \left| \sum_{S\subseteq[n]:|S|=k} \hat{f}(S)\mathbf{x}^S \right| = \max_{\mathbf{x}\in\{-1,1\}^n} \left| \sum_{S\subseteq[n]:|S|=k} \hat{f}(S)\mathbf{x}^S \right|$$

and analogously define $M_k(\mathcal{F}) \triangleq \max_{f\in\mathcal{F}} M_k(f)$ for a class $\mathcal{F}$.

Note that the equality arises by multilinearity, and clearly we have $M_k(f) \leq L_{1,k}(f)$ by the triangle inequality. Without loss of generality, we may further assume that our class is closed under flipping the image, i.e. we may suppose that $f \in \mathcal{F}$ if and only if $-f \in \mathcal{F}$; this transformation does not change either $L_{1,k}(f)$ or $M_k(f)$, and therefore the same bound on the class still holds when completing it to include all such functions. If this is the case, we get the more striking identity:

▶ **Lemma 9.** *Suppose that $\mathcal{F}$ is closed under negation of variables and that $f \in \mathcal{F}$ implies $-f \in \mathcal{F}$. Then*

$$M_k(\mathcal{F}) = \max_{f\in\mathcal{F}} \sum_{S\subseteq[n]:|S|=k} \hat{f}(S) = \max_{f\in\mathcal{F}} f_k(\mathbf{1}).$$

To see why this holds, simply note that if $(f, \mathbf{z}) \in \mathcal{F} \times \{-1,1\}^n$ is a maximizer in the definition of $M_k(\mathcal{F})$ (where we may now assume that the sign is positive), then by replacing the function $f(\mathbf{x})$ with $g(\mathbf{x}) = f(\mathbf{x} \circ \mathbf{z})$, where $\circ$ denotes componentwise multiplication, we have

$$M_k(\mathcal{F}) = \left|\sum_{S\subseteq[n]:|S|=k} \hat{f}(S)\mathbf{z}^S\right| = \sum_{S\subseteq[n]:|S|=k} \hat{g}(S) = \max_{h\in\mathcal{F}} \sum_{S\subseteq[n]:|S|=k} \hat{h}(S).$$

In particular, it suffices to bound the *unsigned level-k Fourier sum* of such a class.

Lastly, we require the following notion:

▶ **Definition 10.** *Let $\mathcal{F}$ be a class of n-variate multilinear polynomials that is closed under restrictions. Define* conv$(\mathcal{F})$ *as the convex closure of $\mathcal{F}$,*

$$\operatorname{conv}(\mathcal{F}) \triangleq \left\{ \sum_{f\in\mathcal{F}} \lambda_f f \,\middle|\, \sum_{f\in\mathcal{F}} \lambda_f = 1, \lambda_f \geq 0\ \forall f \in \mathcal{F} \right\}.$$

We briefly note the following two elementary facts: first, by the assumption that $\mathcal{F}$ is closed under restrictions, the same is true of conv$(\mathcal{F})$. The second is the following simple claim:

▶ **Lemma 11.** *For any class $\mathcal{F}$ of Boolean functions, $M_k(\mathcal{F}) = M_k(\operatorname{conv}(\mathcal{F}))$.*

Proof. One direction is obvious: as $\mathcal{F} \subseteq \operatorname{conv}\mathcal{F}$, clearly $M_k(\mathcal{F}) \leq M_k(\operatorname{conv}(\mathcal{F}))$. In the other direction, let $g = \sum_{f\in\mathcal{F}} \lambda_f f$ be an arbitrary element of conv$(\mathcal{F})$, where $\lambda_f \geq 0$ and $\sum_{f\in\mathcal{F}} \lambda_f = 1$. Then

$$\begin{aligned}
M_k(g) &= \max_{\mathbf{x}\in\{-1,1\}^n} \left|\sum_{S\subseteq[n]:|S|=k} \widehat{g}(S)\mathbf{x}^S\right| \\
&= \max_{\mathbf{x}\in\{-1,1\}^n} \left|\sum_{S\subseteq[n]:|S|=k} \Big(\sum_{f\in\mathcal{F}} \lambda_f \widehat{f}(S)\Big)\mathbf{x}^S\right| \\
&\leq \sum_{f\in\mathcal{F}} \lambda_f \max_{\mathbf{x}\in\{-1,1\}^n} \left|\sum_{S\subseteq[n]:|S|=k} \widehat{f}(S)\mathbf{x}^S\right| \\
&\leq \max_{f\in\mathcal{F}} M_k(f).
\end{aligned}$$

The reverse inequality immediately follows. ◀

2.2 (Fractional) Pseudorandom Generators

We now recall the (well-known) definition of a pseudorandom generator, as well as the generalization of a fractional pseudorandom generator as introduced by [4]:

▶ **Definition 12.** *Let $\mathcal{F}$ be a class of n-variate Boolean functions. Then a* pseudorandom generator *(PRG) for $\mathcal{F}$ with error $\varepsilon > 0$ is a random variable $\mathbf{X} \in \{-1,1\}^n$ such that for all $f \in \mathcal{F}$,*

$$|\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] - \mathbb{E}_{\mathbf{U}_n}[f(\mathbf{U}_n)]| \leq \varepsilon,$$

where $\mathbf{U}_n$ is the uniform distribution on $\{-1,1\}^n$. If $\mathbf{X} = G(\mathbf{U}_s)$ for some explicit function $G : \{-1,1\}^s \to \{-1,1\}^n$, then $\mathbf{X}$ has seed length *s.*

▶ **Definition 13.** *A* fractional pseudorandom generator *(fractional PRG) for $\mathcal{F}$ with error $\varepsilon > 0$ is a random variable $\mathbf{X} \in [-1,1]^n$ such that for all $f \in \mathcal{F}$ (identifying f with its multilinear expansion)*

$$|\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] - f(\mathbf{0})| \leq \varepsilon,$$

*where the definition of seed length is the same. A fractional PRG is p-*noticeable *if for each $i \in [n]$, $\mathbb{E}[\mathbf{X}_i^2] \geq p$.*

We now state the main results of [4] and [6] that show how to construct PRGs from suitably combining noticeable fractional PRGs. This is done by the following *amplification theorem*, which roughly composes fractional random variables into a random walk inside the Boolean hypercube:

▶ **Theorem 14.** *Suppose $\mathcal{F}$ is class of n-variate Boolean functions that is closed under restrictions, and that $\mathbf{X}$ is an explicit p-noticeable fractional PRG with error ε and seed length s. Then there exists an explicit PRG for $\mathcal{F}$ with seed length $O(s\log(n/\varepsilon)/p)$ and error $O(\varepsilon\log(n/\varepsilon)/p)$.*

Using this result, [4] proved the following theorem that exploits strong L_1 control of each Fourier level:

▶ **Theorem 15.** *Let $\mathcal{F}$ be any class of n-variate Boolean functions that is closed under restrictions. Suppose that $L_{1,k}(\mathcal{F}) \leq b^k$ for some $b \geq 1$ and all $1 \leq k \leq n$. Then for any $\varepsilon > 0$, there exists an explicit PRG for $\mathcal{F}$ with error ε and seed length $b^2 \cdot \mathrm{polylog}(n/\varepsilon)$.*

This is achieved by constructing a fractional PRG that is a scaled version of a $\log(1/\varepsilon)$-wise nearly unbiased distribution. As we will be analyzing a similar fractional PRG, we defer the details to next section. To lessen the requisite assumptions on the Fourier spectrum, Chattopadhyay et al. [6] derandomize a construction of Raz and Tal [18] to prove the following result that requires only level-two control, albeit at a cost of exponentially worse dependence on the error ε, and quadratically worse dependence on the level-two mass:

▶ **Theorem 16.** *Let $\mathcal{F}$ be any class of n-variate Boolean functions that is closed under restrictions. Suppose that $L_{1,2}(\mathcal{F}) \leq b^2$ for some $b \geq 1$. Then for any $\varepsilon > 0$, there exists an explicit PRG for $\mathcal{F}$ with error ε and seed length $O((b^2/\varepsilon)^{2+o(1)}\mathrm{polylog}(n))$.*

3 Low-Degree Polynomial Approximations on Subcubes

Throughout this section, we assume that $\mathcal{F}$ is a class of n-variate Boolean functions closed under restrictions. As mentioned above, the main result from which we derive our improvements in constructing pseudorandom generators is essentially a statement about low-degree polynomial approximations on subcubes $[-c,c]^n$ for $c < 1$. We remark that this setting is equivalent to approximating *noisy* versions $T_c f$ on $[-1,1]^n$, where T_ρ is the ρ-noise operator. This is because for any $\mathbf{y} \in [-c,c]^n$, we can write $\mathbf{y} = c\mathbf{x}$ for some $\mathbf{x} \in [-1,1]^n$ and

$$f(\mathbf{y}) = f(c\mathbf{x}) = \sum_{S\subseteq[n]} \hat{f}(S)(c\mathbf{x})^S = \sum_{S\subseteq[n]} c^{|S|}\hat{f}(S)\mathbf{x}^S = T_c f(\mathbf{x}).$$

In general, given any $k \leq n$, $c \geq 0$, and any $f \in \mathcal{F}$, let $\varepsilon_{c,k}(f)$ be defined by

$$\varepsilon_{c,k}(f) \triangleq \inf_{g:\deg(g)<k} \max_{\mathbf{x}\in[-c,c]^n} |f(\mathbf{x}) - g(\mathbf{x})|, \tag{3}$$

and extend the definition to function classes by

$$\varepsilon_{c,k}(\mathcal{F}) \triangleq \max_{f \in \mathcal{F}} \varepsilon_{c,k}(f).$$

Now, given $\varepsilon > 0$, $k \leq n$, and the class $\mathcal{F}$, define $c_k(\mathcal{F}, \varepsilon)$ by

$$c_k(\varepsilon, \mathcal{F}) \triangleq \max\{c \geq 0 : \varepsilon_{c,k}(\mathcal{F}) \leq \varepsilon\}.$$

In words, $c_k(\varepsilon, \mathcal{F})$ measures how small a hypercube we must take to ensure that for every function in our class, there exists a degree-$(k-1)$ approximating polynomial that agrees with f up to a uniform ε error on the subcube $[-c, c]^n$; by multilinearity, it actually suffices that this holds at the extreme points $\{-c, c\}^n$. Note that Equation (3) can be formulated as a linear program and its optimal solution is the best low-degree ℓ_∞-approximation to f.

The main technical claim in this section is that we bound $c_k\,(\varepsilon, \mathcal{F})$ in terms of $M_k(\mathcal{F})$. Specifically, we show that for any class $\mathcal{F}$ that is closed under restrictions, truncating the Fourier expansion of a function $f \in \mathcal{F}$ to its first $(k-1)$ levels serves as a good approximation to f on a sufficiently small hypercube around the origin.

▶ **Theorem 17.** *Let $f \in \mathcal{F}$ that is closed under restrictions. Then for all $c \in (0, 1)$, we have*

$$\max_{\mathbf{x} \in [-c,c]^n} |f_{\geq k}(\mathbf{x})| \leq \left(\frac{c}{1-c}\right)^k M_k(\mathcal{F}).$$

In particular, it follows that

$$\varepsilon_{c,k}(\mathcal{F}) \leq \left(\frac{c}{1-c}\right)^k M_k(\mathcal{F}).$$

From Theorem 17, one immediately obtains a lower bound on $c_k(\varepsilon, \mathcal{F})$:

▶ **Corollary 18.** *For any class $\mathcal{F}$ that is closed under restrictions, and any $\varepsilon > 0$ and $k \leq n$,*

$$c_k(\varepsilon, \mathcal{F}) = \Omega\left(\left(\frac{\varepsilon}{M_k(\mathcal{F})}\right)^{1/k}\right)$$

Proof. Observe that by setting $c = \Omega\left(\left(\frac{\varepsilon}{M_k(\mathcal{F})}\right)^{1/k}\right)$ in Theorem 17, the right side is bounded by ε. Because $f_{\geq k} = f - f_{<k}$ and $f_{<k}$ has degree strictly less than k, it follows immediately from the definition of $c_k(\varepsilon, \mathcal{F})$ that $c_k(\varepsilon, \mathcal{F})$ is at least c. ◀

We now return to the proof of Theorem 17. To prove this result, we require the following intermediate claims. The first simply shows that we may always bound the contribution of the level-k part of any function in $\mathcal{F}$ by simply rescaling the argument:

▶ **Lemma 19.** *Let $f \in \operatorname{conv}(\mathcal{F})$. Then, for all $c \in (0, 1)$ and $\mathbf{x} \in [-c, c]^n$, we have*

$$|f_k(\mathbf{x})| \leq c^k M_k(\mathcal{F}).$$

Proof. Observe that $c^{-1}\mathbf{x} \in [-1, 1]^n$ by assumption, and by homogeneity of f_k as a polynomial, we have

$$|f_k(\mathbf{x})| = c^k |f_k(c^{-1}\mathbf{x})| \leq c^k M_k(\operatorname{conv}(\mathcal{F})) = c^k M_k(\mathcal{F}). \quad ◀$$

The next simple yet powerful claim shows that one can “recenter” functions in $\mathcal{F}$ and they remain in $\mathrm{conv}(\mathcal{F})$ (and therefore, enjoy the same Fourier bounds). This random restriction technique is a key tool in [4].

▶ **Lemma 20.** *Let $f \in \mathrm{conv}(\mathcal{F})$, $\mathbf{a} \in [-1,1]^n$ and $\mathbf{b} \in [0,1]$ such that $|a_i| + b_i \leq 1$ for all $i \in [n]$. Define $\tilde{f}$ by $\tilde{f}(\mathbf{x}) = f(\mathbf{a} + \mathbf{b} \circ \mathbf{x})$, where $\circ$ denotes componentwise multiplication. Then, $\tilde{f} \in \mathrm{conv}(\mathcal{F})$.*

Proof. Given $\mathbf{a}$ and $\mathbf{b}$, define a distribution D_i on $Z_i = \{-1, 1, x_i\}$ where x_i is treated as formal variable, such that $\mathbb{E}_{y_i \sim D_i}[y_i] = a_i + b_i x_i$; note that this is possibly by the assumption that $|a_i| + b_i \leq 1$. Let $D = \prod_i D_i$ be the product distribution of the D_i. For any $\mathbf{z} \in \prod_i Z_i$, define $f_{\mathbf{z}}(\mathbf{x})$ as the function obtained by setting $x_i = z_i$ for each i; in particular, each variable gets set to ±1 or remains a formal variable. By our assumption on the closure of $\mathcal{F}$, we clearly have $f_{\mathbf{z}} \in \mathcal{F}$ for any $\mathbf{z}$. By multilinearity and independence of the product distribution, we have $f(\mathbf{a} + \mathbf{b} \circ \mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim D}[f_{\mathbf{z}}(\mathbf{x})]$. Thus $\tilde{f} \in \mathrm{conv}(\mathcal{F})$. ◀

As mentioned before, our approach will be to bound the higher-order terms of the Fourier expansion at the fractional points of the fractional PRG via the error term that arises in Taylor's theorem. Denote by $h^{(k)}$ the k-th derivative of any C^k function $h : \mathbb{R} \to \mathbb{R}$. We then have the following claim:

▶ **Lemma 21.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be multilinear and let $\mathbf{x} \in \mathbb{R}^n$. Define $g : \mathbb{R} \to \mathbb{R}$ by $g(t) = f(t\mathbf{x})$. Then,*

$$g^{(k)}(0) = k! \cdot f_k(\mathbf{x}).$$

Proof. From the definition, it follows that

$$g(t) = \sum_{S \subseteq [n]} t^{|S|} \hat{f}(S) \mathbf{x}^S.$$

Differentiating g with respect to t, we get

$$g^{(k)}(t) = \sum_{S \subseteq [n] : |S| \geq k} \left(\prod_{i=0}^{k-1} (|S| - i) \right) t^{|S|-k} \hat{f}(S) \mathbf{x}^S.$$

Setting $t = 0$ eliminates all of the monomials with $|S| > k$, giving us the required bound. ◀

The last intermediate result we require connects the function defined in the previous part with our assumed Fourier bounds:

▶ **Lemma 22.** *Let $f \in \mathrm{conv}(\mathcal{F})$, $c \in (0,1)$ and $\mathbf{x} \in [-c,c]^n$. Define g as in Lemma 21. Then,*

$$\max_{s \in [0,1]} \left| g^{(k)}(s) \right| \leq \left(\frac{c}{1-c} \right)^k \cdot k! \cdot M_k(\mathcal{F})$$

Proof. Fix $s \in [0,1]$ and let $\lambda = 1 - c \in [0,1]$. Define the auxiliary function $\tilde{f}(\mathbf{y}) = f(s\mathbf{x} + \lambda \mathbf{y})$. Writing $\mathbf{a} = s\mathbf{x}$ and $\mathbf{b} = (\lambda, \ldots, \lambda)$, we clearly have $s|x_i| + \lambda \leq 1$, so we may apply Lemma 20 to see that $\tilde{f} \in \mathrm{conv}(\mathcal{F})$. Now writing $\tilde{g}(t) = \tilde{f}(t\mathbf{x}) = f(s\mathbf{x} + \lambda t\mathbf{x})$, we also have $\tilde{g}(t) = g(s + t\lambda)$. By the chain rule, differentiating both sides k times and then setting $t = 0$, we have

$$\lambda^k g^{(k)}(s) = \tilde{g}^{(k)}(0).$$

On the other hand, by Lemma 21, we have $\tilde{g}^{(k)}(0) = k! \cdot \tilde{f}_k(\mathbf{x})$, and as $\tilde{f} \in \operatorname{conv}(\mathcal{F})$ by Lemma 20, we conclude using Lemma 19 that

$$\left|g^{(k)}(s)\right| = \left|\frac{\tilde{g}^{(k)}(0)}{\lambda^k}\right| \leq \left(\frac{c}{1-c}\right)^k \cdot k! \cdot M_k(\mathcal{F}). \qquad ◀$$

With these intermediate claims taken care of, we may now put them together to obtain Theorem 17.

Proof of Theorem 17. The second statement follows immediately from the first by setting $g = f_{<k}$ for any given f, and noticing that $f - g = f_{\geq k}$. Therefore, we focus on the first statement.

Let $f \in \mathcal{F}, \mathbf{x} \in [-c, c]^n$ and define $g(t) = f(t\mathbf{x})$. Then, by Taylor expanding g about $t = 0$ and evaluating g at $t = 1$, we have

$$g(1) = \sum_{i<k} \frac{g^{(i)}(0)}{i!} + R_k, \tag{4}$$

where R_k is the error term and is given in Lagrange form by

$$R_k = \frac{g^{(k)}(s)}{k!}$$

for some $s \in (0, 1)$. By Lemma 21, we easily see that the first term in the right hand side of Equation (4) is precisely $f_{<k}(\mathbf{x})$, and as $g(1) = f(\mathbf{x})$, we clearly then must have $R_k = f_{\geq k}(\mathbf{x})$. Therefore, by Lemma 22, we obtain

$$|f_{\geq k}(\mathbf{x})| = \left|\frac{g^{(k)}(s)}{k!}\right| \leq \left(\frac{c}{1-c}\right)^k M_k(\mathcal{F}),$$

as desired. ◀

3.1 Lower Bounds via Chebyshev Polynomials

In this subsection, we show that our bounds on the uniform error of any low-degree polynomial approximator are essentially tight for a reasonable range of $c < 1$. Recall that Theorem 17 shows that the low-degree Fourier expansion is an excellent approximator to the original function for c small enough; we now show that this bound cannot be significantly improved for a reasonable range of c using *any* approximator. Our main result of this section is the following converse:

▶ **Theorem 23.** *Let $\mathcal{F}$ be any class of n-variate multilinear functions that are closed under restrictions. Then for any $c \leq \min\left(\frac{1}{3}, 3^{-k}\frac{M_k(\mathcal{F})}{M_{k+1}(\mathcal{F})}\right)$, we have*

$$\varepsilon_{c,k}(\mathcal{F}) \geq \left(\frac{c}{2}\right)^k M_k(\mathcal{F}).$$

Recall that on the interval $[-1, 1]$, the Chebyshev polynomials give the minimum ℓ_∞ norm among all polynomials with same leading coefficient in magnitude:

▶ **Fact 24** (Theorem 1.5.4 of [17]). *If a polynomial $f : \mathbb{R} \to \mathbb{R}$ is monic of degree n, then $\max_{x\in[-1,1]} |f(x)| \geq 2^{-n+1}$, with equality if and only if $f = T_n$, the normalized n-th Chebyshev polynomial.*

Proof of Theorem 23. Let $(f, \mathbf{x})$ attain the maximum in the definition of $M_k(\mathcal{F})$, namely

$$M_k(\mathcal{F}) = \left| \sum_{S \subseteq [n] : |S| = k} \widehat{f}(S)\mathbf{x}^S \right|.$$

First, note that the claim is trivial if every function in $\mathcal{F}$ is of degree at most k, because then $f_{\geq k}$ is a homogeneous polynomial of degree k and this lower bound is trivial. Under this assumption, $M_{k+1}(\mathcal{F}) > 0$. Fix $c \in (0, 1)$ and let $p : [-1, 1]^n \to \mathbf{R}$ be any multilinear polynomial of degree strictly less than k. Define the univariate function $g : [-1, 1] \to \mathbb{R}$ by

$$g(t) = f(tc\mathbf{x}) - p(tc\mathbf{x}).$$

By taking the Fourier expansion of f, it is easy to see that the coefficient of t^ℓ for $\ell \geq k$ is precisely

$$c^\ell \sum_{S \subseteq [n] : |S| = \ell} \widehat{f}(S)\mathbf{x}^S,$$

so that the coefficient of t^k is equal to $c^k M_k(\mathcal{F})$ in magnitude. We then have

$$\begin{aligned} \sup_{\mathbf{z} \in [-c,c]^n} |f(\mathbf{z}) - p(\mathbf{z})| &\geq \max_{\mathbf{z} \in [-c\mathbf{x}, c\mathbf{x}]} |f(\mathbf{z}) - p(\mathbf{z})| \\ &= \sup_{t \in [-1,1]} |g(t)| \\ &\geq \sup_{t \in [-1,1]} |g_{\leq k}(t)| - \sup_{t \in [-1,1]} |g_{\geq k+1}(t)|. \end{aligned}$$

By Fact 24, the first term is at least $c^k M_k(\mathcal{F})/2^{k-1}$. On the other hand, the second term can be bounded using Theorem 17 by

$$\sup_{t \in [-1,1]} |g_{\geq k+1}(t)| \leq \left(\frac{c}{1-c}\right)^{k+1} M_{k+1}(\mathcal{F}).$$

Therefore, we obtain

$$\sup_{\mathbf{z} \in [-c,c]^n} |f(\mathbf{z}) - p(\mathbf{z})| \geq 2\left(\frac{c}{2}\right)^k M_k(\mathcal{F}) - \left(\frac{c}{1-c}\right)^{k+1} M_{k+1}(\mathcal{F}).$$

It is straightforward to verify that for $c \leq \min\left(1/3, 3^{-k} \frac{M_k(\mathcal{F})}{M_{k+1}(\mathcal{F})}\right)$, the second term is bounded by half of the first. Because p was an arbitrary low-degree multilinear polynomial, the claim follows. ◀

4 From Polynomial Approximations to PRGs

4.1 From Polynomial Approximations to Fractional PRGs

From Theorem 17, we now show how the construction of fractional PRGs from level-k bounds reduces to efficient polynomial approximation on "large" subcubes.

▶ **Theorem 25.** *Let $\mathcal{F}$ be closed under restrictions. Then there exists a fractional PRG for $\mathcal{F}$ with error ε and seed length $O(k \log n)$ that is $(c_k(\varepsilon/2, \mathcal{F}))^2$-noticeable. In particular, if $M_k(\mathcal{F}) = b^k$, there exists such a fractional PRG that is $\Omega\left(\frac{\varepsilon^{2/k}}{b^2}\right)$-noticeable with seed length $O(k \log n)$.*

Proof. The second statement follows immediately from the first using Corollary 18, so we focus on the first statement.

Fix $f \in \mathcal{F}$, $\varepsilon > 0$, and let $\mathbf{X}$ be a $(k-1)$-wise independent random variable over $\{-1,1\}^n$ such that $|\mathbf{X}_i| = c \le 1/2$ for all $i \in [n]$ for some $c > 0$ we specify momentarily. It is well-known that $\mathbf{X}$ can be sampled efficiently with seed length $O(k \log n)$ [25]. By definition of $c := c_k(\varepsilon/2, \mathcal{F})$, there exists a degree-$(k-1)$ multilinear polynomial $\widetilde{f}$ which ε-approximates f on the subcube $[-c,c]^n$, i.e.

$$\max_{y \in [-c,c]^n} \left|f(y) - \widetilde{f}(y)\right| \le \varepsilon/2. \tag{5}$$

Then we have, via the Fourier expansion of f,

$$\begin{aligned}\left|\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] - f(\mathbf{0})\right| &\le \frac{\varepsilon}{2} + \left|\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] - \widetilde{f}(\mathbf{0})\right| \\ &= \frac{\varepsilon}{2} + \left|\mathbb{E}_{\mathbf{X}}\left[f(\mathbf{X}) - \widetilde{f}(\mathbf{X})\right]\right| \\ &\le \frac{\varepsilon}{2} + \mathbb{E}_{\mathbf{X}}\left[\left|f(\mathbf{X}) - \widetilde{f}(\mathbf{X})\right|\right] \\ &\le \varepsilon.\end{aligned}$$

The first inequality applies Equation (5) at the point $\mathbf{x} = \mathbf{0}$, and the second uses the fact that $\mathbf{X}$ is $(k-1)$-wise independent and $\widetilde{f}$ has degree at most $k-1$. The final inequality holds because of (5) and the fact that $\mathbf{X} \in [-c,c]^n$. Therefore, $\mathbf{X}$ satisfies the definition of a fractional PRG. Note that by construction, $\mathbf{X}$ is c^2-noticeable since it takes values in $\{-c,c\}^n$. ◀

Although it does not fit so neatly in this approximation framework, one can essentially recover the improved seed length of [4] (which we recall assumes $L_{1,i}(\mathcal{F})$ bounds for $i = 1,\dots,n$) if one further has $L_{1,i}(\mathcal{F})$ bounds just up to level $k-1$:

▶ **Theorem 26.** *Let $\mathcal{F}$ be closed under restrictions, and suppose that $M_k(\mathcal{F}) \le b^k$ for some $b \ge 1$, $k \ge 2$. If it further holds that $L_{1,i}(\mathcal{F}) \le b^i$ for all $1 \le i < k$, then there exists a $\Theta(\varepsilon^{2/k}/b^2)$-noticeable fractional pseudorandom generator for $\mathcal{F}$ with error ε and seed length $O(\log\log n + \log k + \log(1/\varepsilon))$.*

Proof. Fix $f \in \mathcal{F}$, and let $\mathbf{X}$ be a random variable such that $|\mathbf{X}_i| = c$ for all $i \in [n]$ for some $c > 0$ we specify momentarily. Then we have, via the Fourier expansion of f,

$$\left|\mathbb{E}_{\mathbf{X}}[f(\mathbf{X})] - f(\mathbf{0})\right| = \left|\mathbb{E}_{\mathbf{X}}\left[\sum_{S \subseteq [n] : 1 \le |S| \le k-1} \hat{f}(S)\mathbf{X}^S\right]\right| + \left|\mathbb{E}_{\mathbf{X}}[f_{\ge k}(\mathbf{X})]\right|.$$

We first deal with the second term on the right hand side. By Theorem 17 we have

$$\left|\mathbb{E}_{\mathbf{X}}[f_{\ge k}(\mathbf{X})]\right| \le \left(\frac{c}{1-c}\right)^k M_k(\mathcal{F}).$$

By assumption, $M_k(\mathcal{F}) \le b^k$ for some $b \ge 1$; therefore, by taking $c = \Theta(\varepsilon^{1/k}/b)$, this term is at most $\varepsilon/2$. To deal with the first term, we take the same approach as [4]. Under the assumption $L_{1,i}(\mathcal{F}) \le b^i$ for all $i < k$, one may apply their analysis by letting $\mathbf{X} = c \cdot \mathbf{Y}'$, where $\mathbf{Y}'$ is an $(k-1)$-wise $(\varepsilon/2)$-biased independent random variable over $\{-1,1\}^n$. It is clear that $\mathbf{X}$ is $c^2 = \Theta(\varepsilon^{2/k}/b^2)$-noticeable. Moreover, exactly as in [4], we have

$$\left|\mathbb{E}_{\mathbf{X}}\left[\sum_{S \subseteq [n] : 1 \le |S| \le k-1} \hat{f}(S)\mathbf{X}^S\right]\right| \le \sum_{i=1}^{k-1} c^i \sum_{S \subseteq [n] : |S| = i} |\hat{f}(S)||\mathbb{E}[\mathbf{Y}'^S]| \le (\varepsilon/2)\sum_{i=1}^{k-1}(bc)^i \le \varepsilon/2,$$

because by our choice of c we have $bc \leq 1/2$. By standard constructions, $\mathbf{Y}'$ can be efficiently sampled with seed length $O(\log\log n + \log k + \log(1/\varepsilon))$ [15]. Combining these two errors proves the theorem. ◀

4.2 From Fractional PRGs to PRGs

Using Theorem 25 and Theorem 26 in tandem with Theorem 14, it is fairly immediate to obtain PRGs that rely only on a bound on some k-th Fourier level. Similarly, bounds on levels up to k can be leveraged to get an improved seed length.

▶ **Theorem 27** (Theorem 4, restated). *Let $\mathcal{F}$ be any class of n-variate Boolean functions that is closed under restrictions. Suppose that $M_k(\mathcal{F}) \leq b^k$ for some $b \geq 1$ and $k \geq 3$. Then for any $\varepsilon > 0$, there exists an explicit PRG for $\mathcal{F}$ with error ε with seed length*

$$O\left(\frac{b^{2+\frac{4}{k-2}} \cdot k \log n \cdot \log^{1+\frac{2}{k-2}}(n/\varepsilon)}{\varepsilon^{\frac{2}{k-2}}}\right).$$

If it further holds that $L_{1,i}(\mathcal{F}) \leq b^i$ for all $1 \leq i < k$, then the seed length can be improved to

$$O\left(\frac{b^{2+\frac{4}{k-2}} \cdot (\log\log n + \log k + \log(b/\varepsilon)) \cdot \log^{1+\frac{2}{k-2}}(n/\varepsilon)}{\varepsilon^{\frac{2}{k-2}}}\right).$$

Proof. By Theorem 14, given an explicit p-noticeable fractional PRG for $\mathcal{F}$ with error δ and seed length s, one immediately obtains an explicit PRG for $\mathcal{F}$ with error $O(\delta \log(n/\delta)/p)$ and seed length $O(s\log(n/\delta)/p)$.

For the first statement, by our assumption and using the fractional PRG guaranteed by Theorem 25, for any $\delta > 0$, we immediately obtain an explicit PRG for $\mathcal{F}$ with error $O(b^2\delta^{1-2/k}\log(n/\delta))$ and seed length $O(b^2 k \log(n)\log(n/\delta)/\delta^{2/k})$. To get the error below ε, we set

$$\delta = \Theta\left(\left(\frac{\varepsilon}{b^2\log(n/\varepsilon)}\right)^{\frac{k}{k-2}}\right)$$

(the astute reader may notice we implicitly use $b \leq n$ here). This yields a PRG with error ε and seed length

$$O\left(\frac{b^{2+\frac{4}{k-2}} \cdot k \log n \cdot \log^{1+\frac{2}{k-2}}(n/\varepsilon)}{\varepsilon^{\frac{2}{k-2}}}\right).$$

The second statement follows in an identical manner from the improved seed length given in the second part of Theorem 26 in the case that one has control on the L_1 Fourier mass on the lower levels. ◀

Corollary 5 is now an immediate consequence of Theorem 27; for any desired $\varepsilon > b \cdot \log(n) \cdot 2^{-O(k)}$, one can simply apply Theorem 27 using level $k = \Theta(\log(b\log(n)/\varepsilon))$ to obtain a PRG for $\mathcal{F}$ with error at most ε with seed length

$$O(b^2 \cdot \log(b\log(n)/\varepsilon) \cdot \log(n/\varepsilon)).$$

Note that for error $\varepsilon = 1/\mathrm{poly}(n)$, one needs bounds only up to level $\Theta(\log n)$ (again, using the fact that $b \leq n$). This also partially answers an open question of [6], which asks how many levels of Fourier bounds suffice to recover polylogarithmic dependence in $1/\varepsilon$.

▶ Remark 28. Note that this Taylor's theorem approach does not yield anything nontrivial given bounds just on the second level, unlike the fractional PRG in [6]. This is actually a necessary byproduct of combining this approach with the random walk gadget of [4]. Given only level-two bounds, this approach attempts to use j-wise independence for $j < k = 2$ and smallness to deal with errors on the high degree terms ($k \geq 2$). However, the trivial random variable that is $\pm\mathbf{1}$ with equal probability is trivially 1-wise independent, as each component is a uniform random bit, albeit completely correlated. No matter how we scale them, one can show that composing arbitrarily many independent copies of this random variable via the random walk gadget must necessarily polarize to $\pm\mathbf{1}$ at termination, which clearly cannot fool any nontrivial functions.

5 Low-degree Polynomials over $\mathbb{F}_2$

Our analysis recovers all the existing applications of [4] (among them, $\mathbf{AC}^0$ circuits, low-sensitivity functions, and read-once branching programs); indeed, all the classes considered there satisfy L_1 Fourier bounds on the entire tail. To our knowledge, our new analysis does not immediately improve the seed lengths obtained there, though it shows that (i) *the seed lengths there can potentially be improved using stronger bounds on M_k*, and (ii) *the PRGs there would still have fooled those classes had these Fourier bounds been known only up to some level k.*

However, the generality afforded to us by this new analysis allows us to obtain a new PRG for low-degree polynomials over $\mathbb{F}_2$, which addresses an open question of [4] by showing that this framework can handle this class. Indeed, let $\mathcal{F}$ be the set of n-variate, degree-d polynomials over $\mathbb{F}_2$. As a preliminary step towards deriving Fourier tail bounds that would imply a nontrivial PRG for this class using their framework, [4] proves the following Fourier bounds:

▶ **Proposition 29** (Theorem 6.1 of [4]). *Let $p\colon \mathbb{F}_2^n \to \mathbb{F}_2$ be a degree-d polynomial, and let $f(\mathbf{x}) = (-1)^{p(\mathbf{x})}$. Then $L_{1,k}(f) \leq (k \cdot 2^{3d})^k$.*

Note that this result cannot be applied to their original analysis, for they require a nontrivial bound at all levels, while this bound is trivial for $k = \Omega(\sqrt{n})$ and any d. While Theorem 16 can yield a nontrivial PRG by just applying the level-two bound, the dependence on $1/\varepsilon$ is at least quadratic.[4] However, using our new, more flexible analysis, one can obtain a nontrivial PRG with polylogarithmic dependence on the error parameter. Our formal result is the following:

▶ **Theorem 30.** *Let $\mathcal{F}$ be the class of degree-d polynomials over $\mathbb{F}_2$ on n variables. Then there exists an explicit pseudorandom generator for $\mathcal{F}$ with error ε and seed length*

$$2^{O(d)} \cdot \log^3(\log(n)/\varepsilon) \cdot \log(n/\varepsilon).$$

Proof. Fix $\varepsilon > 0$ and let $k = \Theta(\log(\log(n)/\varepsilon))$. By Proposition 29, we have that for all $j \leq k$,

$$L_{1,j}(\mathcal{F}) \leq \Theta\big(\log(\log(n)/\varepsilon) \cdot 2^{3d}\big)^j.$$

[4] By applying this Fourier bound at level-two, one can use the fractional PRG of [6] to obtain seed length $2^{O(d)}\mathrm{polylog}(n)/\varepsilon^{2+o(1)}$ using the random walks framework. This gives exponentially worse error dependence compared to our approach.

By setting $b = \Theta(\log(\log(n)/\varepsilon) \cdot 2^{3d})$, we may apply Theorem 27 for $\mathcal{F}$ and error ε. Note that $\varepsilon^{-\Theta(1/\log(1/\varepsilon))} = O(1)$, so plugging in this value of b, we immediately obtain the desired pseudorandom generator. ◀

For comparison, the best known construction by Viola [26], obtained by summing d independent copies of a sufficiently good small-bias space, attains seed length $d \cdot \log n + O(d \cdot 2^d \log(1/\varepsilon))$, which for constant ε and d is within a constant factor of the optimal seed length. The generator implied by our analysis recovers this polylogarithmic dependence in n/ε, although with slightly worse dependence on $\log n$ and polynomially worse dependence in $\log(1/\varepsilon)$. Neither generator can handle superlogarithmic degree. While this result clearly falls short of the state-of-the-art, we emphasize that this generator is conceptually distinct from the existing constructions, and yet belongs to this generic random walk framework.

Our analysis allows us to exploit known Fourier bounds that are too weak for the existing analyses to obtain polylogarithmic error dependence. In particular, to get a nontrivial pseudorandom generator for polynomials of superlogarithmic degree with nontrivial seed length, our work shows that the following weaker conjecture would suffice to break the logarithmic degree barrier and still achieve polylogarithmic (in n) seed length for $\varepsilon = 1/\text{poly}(n)$:

▶ **Conjecture 31.** *Let $\mathcal{F}$ be the class of degree-d polynomials over $\mathbb{F}_2$ on n variables. Then*

$$M_k(\mathcal{F}) \leq (\text{poly}(k, \log n) \cdot 2^{o(d)})^k$$

for $k \leq O(\log n)$.

In fact, we observe that to break the logarithmic degree barrier, it actually suffices that this holds just at level $k = 3$, though with poor dependence on ε. Note that this is a significantly weaker conjecture than positing that the same bounds hold for $L_{1,k}(\mathcal{F})$. Moreover, as we explain in the next section, $M_k(\mathcal{F})$ can be controlled using correlation bounds, which are much better studied than L_1 Fourier bounds.

6 Bounds on $M_k(\mathcal{F})$ via Correlation with Shifted Majorities

As we have seen, our new analysis lets one construct PRGs from the weaker quantity $M_k(\mathcal{F})$. In this section, we extend the argument of Chattopadhyay, Hatami, Hosseini, Lovett, and Zuckerman [5] to show how bounds on $M_k(\mathcal{F})$ follow from covariance bounds with certain resilient functions (in particular, shifted majorities). In their paper, they deal with the case of $k = 2$; we rather straightforwardly generalize this argument, but stress that the approach is the same as in Section 6 of their paper. To that end, for convenience and consistency with their argument, we adopt their conventions and requisite definitions just for this section. We will now consider Boolean functions written as $f : \{0,1\}^n \to \{0,1\}$. Translating to this notation, for any such Boolean function f, let $e(f)(\mathbf{x}) \triangleq (-1)^{f(\mathbf{x})}$. Then, letting $F = e(f)$, we now have $\hat{F}(S) = \mathbb{E}_{\mathbf{x}}[F(\mathbf{x})e(\sum_{i \in S} x_i)]$.

▶ **Definition 32.** *The* covariance between f and g, *where f, g are Boolean is*

$$\text{cov}(f, g) \triangleq \left|\mathbb{E}[e(f(\mathbf{x}))e(g(\mathbf{x}))] - \mathbb{E}[e(f(\mathbf{x}))]\mathbb{E}[e(g(\mathbf{x}))]\right|.$$

The covariance between a function f and a class $\mathcal{G}$ is defined as $\text{cov}(f, \mathcal{G}) \triangleq \max_{g \in \mathcal{G}} \text{cov}(f, g)$.

For any $\mathbf{x} \in \{0,1\}^n$, we write $|\mathbf{x}|$ for its Hamming weight, i.e. $\sum_{i=1}^n x_i$. For any $a \in \{0,1,\dots,n\}$, [5] defines Maj_a by

$$\mathrm{Maj}_a(\mathbf{x}) \triangleq \begin{cases} 1 & \text{if } |\mathbf{x}| > a \\ 0 & \text{otherwise,} \end{cases}$$

as well as the following associated functions for any $\theta \in [n/2]$:

$$\mathrm{Thr}_\theta(x) \triangleq \begin{cases} (-1)^{\mathrm{Maj}_{n/2}(\mathbf{x})} & \text{if } \big||\mathbf{x}| - n/2\big| > \theta \\ 0 & \text{otherwise.} \end{cases}$$

We now prove the following lemma relating $M_k(\mathcal{F})$ with covariance bounds against the k-XORs of these functions:

▶ **Lemma 33** (Lemma 6.1 of [5], adapted). *Let $\mathcal{F}$ be any family of (kn)-variate Boolean functions that is closed under relabeling and negation of input variables. Suppose that for any $a_1,\dots,a_k$ such that $|a_i - n/2| = O(\sqrt{kn\log n})$ for all $i \in [k]$, and all $f \in \mathcal{F}$, we have for some $t \geq 1$*

$$\mathrm{cov}\big(f, \oplus_{i=1}^k \mathrm{Maj}_{a_i}\big) \leq \left(\sqrt{\frac{t}{n}}\right)^k,$$

where $\oplus$ denotes the XOR *function. Then,*

$$M_k(\mathcal{F}) \leq O\big(\sqrt{tk\log n}\big)^k.$$

To prove this lemma, [5] uses the following sequence of claims.

▶ **Fact 34** (Claim 6.2 in [5]). *For any $f \in \mathcal{F}$, let $F(\mathbf{x}_1,\dots,\mathbf{x}_k) = e(f(\mathbf{x}_1,\dots,\mathbf{x}_k))$. Under the hypotheses of Lemma 33, for any $1 \leq a_1,\dots,a_k \leq O(\sqrt{kn\log n})$,*

$$\left|\mathbb{E}_{\mathbf{x}_1,\dots,\mathbf{x}_k}\left[\big(F(\mathbf{x}_1,\dots,\mathbf{x}_k) - \mathbb{E}[F]\big)\prod_{i=1}^k \mathrm{Thr}_{a_i}(\mathbf{x}_i)\right]\right| \leq \left(\sqrt{\frac{t}{n}}\right)^k.$$

▶ **Fact 35** (Claim 6.3 of [5]). *For any $\mathbf{x} \in \{0,1\}^n$, $\sum_{i=1}^n e(\mathbf{x}_i) = 2\sum_{1\leq a\leq n/2} \mathrm{Thr}_a(\mathbf{x})$.*

▶ **Fact 36** (Claim 6.4 of [5], adapted). *For any Boolean function $f : \{0,1\}^{kn} \to \{0,1\}$, there exists a k-equipartition of $[kn]$ into disjoint sets $S_1,\dots,S_k$ such that*

$$\left|\sum_{S\subseteq[kn]:|S|=k} \hat{f}(S)\right| \leq C^k \left|\sum_{i_j\in S_j\ \forall j\in[k]} \hat{f}(\{i_1,\dots,i_k\})\right|$$

for some absolute constant $C > 0$.

As this fact is not quite identical to that in [5], we give an argument here:

Proof. We use the probabilistic method: let $\mathcal{P}$ be the set of k-equipartitions of $[kn]$. Let $T \subseteq [kn]$ of size k be arbitrary; without loss of generality, suppose $T = [k]$. Consider a uniformly random k-equipartition $P = S_1 \sqcup \cdots \sqcup S_k \in \mathcal{P}$. The probability that each $i \in T$ belongs to a distinct S_j is easily seen to be

$$\prod_{i=1}^{k-1} \frac{(k-i)\cdot n}{kn-i} \geq \frac{(k-1)!\, n^{k-1}}{(kn)^{k-1}} = \frac{(k-1)!}{k^{k-1}} = e^{-O(k)},$$

where the last equality uses Stirling's approximation. By symmetry, let $\alpha \in \mathbb{N}$ be the number of k-equipartitions that any arbitrary subset T is in. Then we have

$$\begin{aligned}\alpha\left|\sum_{S\subseteq[kn]:|S|=k}\hat{f}(S)\right| &= \left|\sum_{P\in\mathcal{P}}\sum_{i_j\in S_j\ \forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right|\\ &\le \sum_{P\in\mathcal{P}}\left|\sum_{i_j\in S_j\ \forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right|\\ &\le |\mathcal{P}|\max_{P\in\mathcal{P}}\left|\sum_{i_j\in S_j\ \forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right|. \qquad \blacktriangleleft\end{aligned}$$

The first line follows from simple counting, while the second is the triangle inequality. Rearranging, we deduce that (writing T as a generic subset of size k)

$$\begin{aligned}\left|\sum_{S\subseteq[kn]:|S|=k}\hat{f}(S)\right| &\le \frac{|\mathcal{P}|}{\alpha}\max_{P\in\mathcal{P}}\left|\sum_{i_j\in S_j\ \forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right|\\ &= \Pr_{P\sim\mathcal{P}}(T\in P)^{-1}\max_{P\in\mathcal{P}}\left|\sum_{i_j\in S_j\ \forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right|\\ &\le e^{O(k)}\max_{P\in\mathcal{P}}\left|\sum_{i_j\in S_j\ \forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right|.\end{aligned}$$

The last fact that is needed can be deduced from the Chernoff bound:

▶ **Fact 37** (Claim 6.5 of [5], adapted). *For any $a \ge \Omega(\sqrt{kn\log n})$, $\mathbb{E}[|\mathrm{Thr}_a|] \le O(1/n^k)$.*

With these facts, we can now prove Lemma 33 in an entirely analogous fashion to [5]:

Proof of Lemma 33. Fix $f \in \mathcal{F}$, and again write $F(\mathbf{x}_1,\dots,\mathbf{x}_k) = e(f(\mathbf{x}_1,\dots,\mathbf{x}_k))$. Let $F' = F - \mathbb{E}[F]$. Let $U_j = \{i : (j-1)n+1 \le i \le jn\}$. Then, possibly after relabelling variables, we have by Fact 36 that

$$\left|\sum_{S\subseteq[kn]:|S|=k}\hat{f}(S)\right| \le C^k\left|\sum_{i_j\in U_j,\forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right|,$$

so we may turn to bounding this latter term. We have

$$\begin{aligned}\left|\sum_{i_j\in U_j,\forall j\in[k]}\hat{f}(\{i_1,\dots,i_k\})\right| &= \left|\sum_{i_j\in U_j,\forall j\in[k]}\mathbb{E}\Big[F'(\mathbf{x}_1,\dots,\mathbf{x}_k)\prod_{j=1}^{k}e\Big((\mathbf{x}_j)_{i_j}\Big)\Big]\right|\\ &= \left|\mathbb{E}\Big[F'(\mathbf{x}_1,\dots,\mathbf{x}_k)\prod_{j=1}^{k}\Big(\sum_{i_j\in U_j}e\Big((\mathbf{x}_j)_{i_j}\Big)\Big)\Big]\right|\\ &\le 2^k\sum_{1\le a_i\le n/2,\forall i\in[k]}\left|\mathbb{E}\Big[F'(\mathbf{x}_1,\dots,\mathbf{x}_k)\prod_{i=1}^{k}\mathrm{Thr}_{a_i}(\mathbf{x}_i)\Big]\right|\\ &\le 2^k\left(\sum_{1\le a_i\le O(\sqrt{kn\log n}),\forall i\in[k]}\left|\mathbb{E}\Big[F'(\mathbf{x}_1,\dots,\mathbf{x}_k)\prod_{i=1}^{k}\mathrm{Thr}_{a_i}(\mathbf{x}_i)\Big]\right| + O(1)\right)\\ &\le 2^k\cdot O\Big(\sqrt{kn\log n}\Big)^k\cdot\left(\sqrt{\frac{t}{n}}\right)^k\\ &= O\Big(\sqrt{tk\log n}\Big)^k.\end{aligned}$$

The first inequality follows from Fact 35, the second from Fact 37, and the last from Fact 34. Because we assumed that $\mathcal{F}$ is closed under negations of input variables and $f \in \mathcal{F}$ was arbitrary, we obtain the desired claim from Lemma 9 after absorbing the constant C above into the implicit constant in this bound. ◀

7 Discussion and Open Questions

In this work, we have given a nearly complete interpolation between the previous PRGs obtained in the polarizing random walk framework by exploiting level-k bounds on the class of functions, thus answering an open question from [6]. We do so by exploiting an alternate Fourier analysis via Taylor's theorem and utilizing multilinearity and random restrictions. This new analysis enables us to construct PRGs from bounds on the potentially much smaller and better-understood Fourier quantity $M_k(\mathcal{F})$, for any $k \geq 3$. By generalizing the connection established in [5], this reduces the problem of constructing PRGs in this framework to proving correlation bounds. Further, we show how to get a PRG with an improved seed length if we have bounds on $L_{1,i}(\mathcal{F})$, for all $i \leq k$, where $k \geq 3$. A natural open question along these lines is to obtain the improved seed length using bounds on $M_i(\mathcal{F})$ (instead of $L_{1,i}(\mathcal{F})$) for all $i \leq k$. Another natural question is to construct a PRG using bounds on just M_2 (recall that [6] gives such a construction using bounds on $L_{1,2}(\mathcal{F})$ and our analysis only gives a non-trivial PRG from bounds on $M_k(\mathcal{F})$ when $k \geq 3$).

Finally, exploiting known level-k bounds for $\mathbb{F}_2$ polynomials, our approach shows that the polarizing random walk framework can yield pseudorandom generators for the class of $\mathbb{F}_2$ polynomials that is competitive with the state of the art. As mentioned, we hope this paper gives evidence that stronger Fourier control (perhaps via proving the required correlation bounds) can give better PRGs using this framework, and can also handle classes that were previously not known to be possible. In particular, we emphasize that proving Conjecture 31 even for the case of $k = 3$ will lead to PRGs for $\mathbb{F}_2$-polynomials with degree $\omega(\log n)$, a longstanding problem in complexity theory.

References

1 Rohit Agrawal. Coin theorems and the Fourier expansion. *Chicago Journal of Theoretical Computer Science*, 2020(4), August 2020.

2 Srinivasan Arunachalam, Sourav Chakraborty, Michal Koucký, Nitin Saurabh, and Ronald de Wolf. Improved bounds on Fourier entropy and min-entropy. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, volume 154 of *LIPIcs*, pages 45:1–45:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.STACS.2020.45.

3 Nikhil Bansal and Makrand Sinha. k-forrelation optimally separates quantum and classical query complexity. *CoRR*, abs/2008.07003, 2020. arXiv:2008.07003.

4 Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. *Theory of Computing*, 15(10):1–26, 2019. doi:10.4086/toc.2019.v015a010.

5 Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, Shachar Lovett, and David Zuckerman. XOR lemmas for resilient functions against polynomials. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 234–246, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384242.

6 Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. Pseudorandom Generators from the Second Fourier Level and Applications to AC0 with Parity Gates. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ITCS.2019.22.

7 Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 363–375, 2018. doi:10.1145/3188745.3188800.

8 Andreas Defant, Leonhard Frerick, Joaquim Ortega-Cerdà, Myriam Ounaïes, and Kristian Seip. The Bohnenblust-Hille inequality for homogeneous polynomials is hypercontractive. *Annals of mathematics*, pages 485–497, 2011.

9 Uma Girish, Ran Raz, and Wei Zhan. Lower bounds for XOR of forrelations. *CoRR*, abs/2007.03631, 2020. arXiv:2007.03631.

10 Parikshit Gopalan, Rocco A. Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions, 2016. arXiv:1604.07432.

11 Parikshit Gopalan, Rocco A. Servedio, and Avi Wigderson. Degree and sensitivity: Tails of two distributions. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 13:1–13:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.13.

12 Chin Ho Lee. Fourier bounds and pseudorandom generators for product tests. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 7:1–7:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.7.

13 Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. In *30th Annual Symposium on Foundations of Computer Science*, pages 574–579. IEEE, 1989.

14 Ashley Montanaro. Some applications of hypercontractive inequalities in quantum information theory. *Journal of Mathematical Physics*, 53(12):122206, 2012.

15 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 213–223. ACM, 1990. doi:10.1145/100216.100244.

16 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi:10.1017/CBO9781139814782.

17 Qazi Ibadu Rahman and Gerhard Schmeisser. *Analytic theory of polynomials*, volume 26 of *London Mathematical Society Monographs. New Series*. The Clarendon Press, Oxford University Press, Oxford, 2002.

18 Ran Raz and Avishay Tal. Oracle separation of BQP and PH. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 13–23. ACM, 2019. doi:10.1145/3313276.3316315.

19 Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41(4):333–338, April 1987. doi:10.1007/BF01137685.

20 Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. An optimal separation of randomized and quantum query complexity. *Electron. Colloquium Comput. Complex.*, 27:128, 2020. URL: https://eccc.weizmann.ac.il/report/2020/128.

21 Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 77–82, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/28395.28404.

22 Roman Smolensky. On representations by low-degree polynomials. In *Proceedings of the 1993 IEEE 34th Annual Foundations of Computer Science*, SFCS '93, page 130–138, USA, 1993. IEEE Computer Society. doi:10.1109/SFCS.1993.366874.

23 Avishay Tal. Tight bounds on the Fourier spectrum of AC0. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 15:1–15:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.CCC.2017.15.

24 Avishay Tal. Towards optimal separations between quantum and randomized query complexities. *Electron. Colloquium Comput. Complex.*, 26:179, 2019. URL: https://eccc.weizmann.ac.il/report/2019/179.

25 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1–3):1–336, 2012.

26 Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. *Computational Complexity*, 18(2):209–217, 2009. doi:10.1007/s00037-009-0273-5.

27 Emanuele Viola. Fourier conjectures, correlation bounds, and majority. *Electron. Colloquium Comput. Complex.*, 27:175, 2020. URL: https://eccc.weizmann.ac.il/report/2020/175.

28 Xinyu Wu. A stochastic calculus approach to the oracle separation of BQP and PH. *CoRR*, abs/2007.02431, 2020. arXiv:2007.02431.

Deterministic Identity Testing Paradigms for Bounded Top-Fanin Depth-4 Circuits

Pranjal Dutta
Chennai Mathematical Institute, India
Department of Computer Science & Engineering, IIT Kanpur, India

Prateek Dwivedi
Department of Computer Science & Engineering, IIT Kanpur, India

Nitin Saxena
Department of Computer Science & Engineering, IIT Kanpur, India

Abstract

Polynomial Identity Testing (PIT) is a fundamental computational problem. The famous depth-4 reduction (Agrawal & Vinay, FOCS'08) has made PIT for depth-4 circuits, an enticing pursuit. The largely open special-cases of sum-product-of-sum-of-univariates ($\Sigma^{[k]}\Pi\Sigma\wedge$) and sum-product-of-constant-degree-polynomials ($\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$), for constants k, δ, have been a source of many great ideas in the last two decades. For eg. depth-3 ideas (Dvir & Shpilka, STOC'05; Kayal & Saxena, CCC'06; Saxena & Seshadhri, FOCS'10, STOC'11); depth-4 ideas (Beecken,Mittmann & Saxena, ICALP'11; Saha,Saxena & Saptharishi, Comput.Compl.'13; Forbes, FOCS'15; Kumar & Saraf, CCC'16); geometric Sylvester-Gallai ideas (Kayal & Saraf, FOCS'09; Shpilka, STOC'19; Peleg & Shpilka, CCC'20, STOC'21). We solve two of the basic underlying open problems in this work.

We give the *first* polynomial-time PIT for $\Sigma^{[k]}\Pi\Sigma\wedge$. Further, we give the *first* quasipolynomial time *blackbox* PIT for both $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$. No subexponential time algorithm was known prior to this work (even if $k = \delta = 3$). A key technical ingredient in all the three algorithms is how the *logarithmic derivative*, and its power-series, modify the top Π-gate to $\wedge$.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Polynomial identity testing, hitting set, depth-4 circuits

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.11

Funding *Pranjal Dutta*: Google Ph. D. Fellowship
Nitin Saxena: DST (DST/SJF/MSA-01/2013-14) and N. Rama Rao Chair

Acknowledgements Pranjal thanks CSE, IIT Kanpur for the hospitality.

1 Introduction: PIT & beyond

Algebraic circuits are natural algebraic analog of boolean circuits, with the logical operations being replaced by + and × operations over the underlying field. The study of algebraic circuits comprise the large study of algebraic complexity, mainly pioneered (and formalized) by Valiant [87]. A central problem in algebraic complexity is an algorithmic design problem, known as Polynomial Identity Testing (PIT): given an algebraic circuit $\mathcal{C}$ over a field $\mathbb{F}$ and input variables $x_1, \dots, x_n$, determine whether $\mathcal{C}$ computes the identically zero polynomial. PIT has found numerous applications and connections to other algorithmic problems. Among the examples are algorithms for finding perfect matchings in graphs [59, 62, 24], primality testing [4], polynomial factoring [52, 19], polynomial equivalence [21], reconstruction algorithms [48, 83, 44] and the existence of algebraic natural proofs [16, 53]. Moreover, efficient design of PIT algorithms is intrinsically connected to proving strong lower bounds [39, 1, 42, 23, 29, 17, 20]. Interestingly, PIT also emerges in

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 11; pp. 11:1–11:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

many fundamental results in complexity theory such as $\mathsf{IP} = \mathsf{PSPACE}$ [82, 60], the PCP theorem [10, 11], and the overarching Geometric Complexity Theory (GCT) program towards $\mathsf{P} \neq \mathsf{NP}$ [64, 63, 32, 41].

There are broadly two settings in which the PIT question can be framed. In the *whitebox* setup, we are allowed to look inside the wirings of the circuit, while in the *blackbox* setting we can only evaluate the circuit at some points from the given domain. There is a very simple randomized algorithm for this problem - evaluate the polynomial at a random point from a large enough domain. With very high probability, a nonzero polynomial will have a nonzero evaluation; this is famously known as the Polynomial Identity Lemma [66, 18, 89, 81]. It has been a long standing open question to derandomize this algorithm.

For many years, blackbox identity tests were only known for depth-2 circuits (equivalently sparse polynomials) [13, 49]. In a surprising result, Agrawal and Vinay [7] showed that a complete derandomization of blackbox identity testing for just depth-4 algebraic circuits ($\Sigma\Pi\Sigma\Pi$) already implies a near complete derandomization for the general PIT problem. More recent depth reduction results [50, 36], and the bootstrapping phenomenon [2, 55, 34, 9] show that even PIT for very restricted classes of depth-4 circuits (*even* depth-3) would have very interesting consequences for PIT of general circuits. These results make the identity testing regime for depth-4 circuits, a very meaningful pursuit.

Three PITs in one-shot. Following the same spirit, here we solve three important (and open) PIT questions. We give the *first* deterministic polynomial-time whitebox PIT algorithm for the bounded sum-of-product-of-sum-of-univariates circuits ($\Sigma^{[k]}\Pi\Sigma\wedge$) [71, Open Prob. 2]; polynomials computed by these circuits are of the form $\Sigma_{i\in[k]}\Pi_j\,(g_{ij1}(x_1) + \cdots + g_{ijn}(x_n))$ (Theorem 1). In fact, we also design the first quasipolynomial-time blackbox PIT algorithm for the same model (Theorem 2a). To the best of our knowledge, no subexponential time algorithm was known prior to this work. A similar technique also gives a quasipolynomial-time blackbox PIT algorithm for the bounded top and bottom fanin circuits $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ (where k and δ are constants), see Theorem 2b. These circuits compute polynomials of the form $\Sigma_{i\in[k]}\Pi_j g_{ij}(\boldsymbol{x})$, where $\deg(g_{ij}) \leq \delta$. Even $\delta = 2$ was a challenging open problem [56, Open Prob. 2].

Prior works on the related models. In the last two decades, there has been a surge of results on identity testing for restricted classes of bounded depth algebraic circuits (eg. "locally" bounded independence, bounded read/occur, bounded variables). There have been numerous results on PIT for depth-3 circuits with bounded top fanin (known as $\Sigma^{[k]}\Pi\Sigma$-circuits). Divir and Shpilka [22] gave the first quasipolynomial-time deterministic whitebox algorithm for $k = O(1)$, using rank based methods, which finally lead Karnin and Shpilka [45] to design algorithm of same complexity in the blackbox setting. Kayal and Saxena [47] gave the first polynomial-time algorithm of the same. Later, a series of works in [78, 79, 80, 5] generalized the model and gave $n^{O(k)}$-time algorithm when the algebraic rank of the product polynomials are bounded.

There has also been some progress on PIT for restricted classes of depth-4 circuits. A quasipolynomial-time blackbox PIT algorithm for *multilinear* $\Sigma^{[k]}\Pi\Sigma\Pi$-circuits was designed in [43], which was further improved to a $n^{O(k^2)}$-time deterministic algorithm in [74]. A quasipolynomial blackbox PIT was given in [12, 56] when algebraic rank of the irreducible factors in each multiplication gate as well as the bottom fanin are bounded. Further interesting restrictions like sum of product of fewer variables, and more structural restrictions have been exploited, see [28, 6, 25, 61, 57]. Some progress has also been made for bounded top-fanin and bottom-fanin depth-4 circuits via incidence geometry [35, 84, 68]. In fact, very recently, [69] gave a polynomial-time blackbox PIT for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$-circuits.

Why were the problems open? As mentioned above, people have studied depth-4 PIT only with extra restrictions, mostly due to the limited applicability of the existing techniques: they were tailor-made for the specific models and do not generalize. Eg. the previous methods handle $\delta = 1$ (i.e. linear polynomials at the bottom) or $k = 2$ (via *factoring*, [71]). While $k = 2$ to 3, or $\delta = 1$ to 2 (i.e. "linear" to "quadratic") already demands a qualitatively different approach.

Whitebox $\Sigma^{[k]}\Pi\Sigma\wedge$ model generalizes the famous bounded-top-fanin-depth-3 $\Sigma^{[k]}\Pi\Sigma$ of [47]; but their Chinese Remaindering (CR) method, loses applicability and thus fails to solve even a slightly more general model. The blackbox setting involved similar "certifying path" ideas [79] which could be thought of as general CR. It comes up with an ideal I such that $f \neq 0 \bmod I$ and finally preserves it under a constant-variate linear map. The preservation gets harder (for both $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$) due to the increased non-linearity of the ideal I generators. Intuitively, larger δ, via ideal-based routes, brings us to the Gröbner basis method (which is doubly-exponential-time in n) [88]. We know that ideals even with 3-generators (analogously $k = 4$) already capture the whole ideal-membership problem [73].

The algebraic-geometric approach to $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ has been explored in [12, 35, 61, 33]. The families which satisfy a certain Sylvester–Gallai configuration (called SG-circuits) is the harder case which is conjectured to have constant transcendence degree [35, Conj. 1]. Non-SG circuits is the case where the nonzeroness-certifying-path question reduces to radical-ideal non-membership questions [30]. This is really a variety question where one could use algebraic-geometry tools to design a poly-time blackbox PIT. In fact, very recently, Guo [33] gave a s^{δ^k}-time PIT by constructing explicit variety evasive subspace families. Unfortunately, this is not the case in the ideal non-membership; this scenario makes it much harder to solve $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$. From this viewpoint, radical-ideal-membership explains well why the intuitive $\Sigma^{[k]}\Pi\Sigma$ methods do not extend to $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$.

Interestingly, Forbes [25] found a quasipolynomial-time PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ using shifted-partial derivative techniques; but it naively fails when one replaces the $\wedge$-gate by Π (the "measure" becomes too large). The "duality trick" [75] completely solves whitebox PIT for $\Sigma\wedge\Sigma\wedge$, by transforming it to a read-once oblivious ABP (ROABP); but it is inapplicable to our models with the top Π-gate (due to large waring rank and ROABP-width). A priori, our models are incomparable to ROABP, and thus, the famous PIT algorithms for ROABP [28, 27, 37] are not expected to help either.

Similarly, a naive application of the "Jacobian" + "certifying path" technique [5] fails for our models because it is difficult to come up with a *faithful* map (for constant-variate reduction). Kumar and Saraf [56] crucially used that the computed polynomial has low individual degree (such that [23] can be invoked), while in [57] they exploits the low algebraic rank of the polynomials computed below the top Π-gate. Neither of them hold, in general, for our models. Very recently, Peleg and Shpilka [69] gave a poly-time blackbox PIT for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$, via incidence geometry (eg. Edelstein-Kelly theorem involving "quadratic" polynomials), by solving [35, Conj. 1] for $k = 3, \delta = 2$. The method seems very strenuous to generalize even to "cubic" polynomials ($\delta = 3 = k$).

PIT for other models. Blackbox PIT algorithms for many restricted models are known. Egs. ROABP related models [70, 40, 3, 37, 38, 27, 8], log-variate circuits [26, 14], certain non-commutative models [31, 58]. We refer to [85, 76, 64, 77, 54, 72] for detailed surveys on PIT and related topics.

1.1 Our results: An analytic detour to three PITs

Though some attempts have been made to solve PIT for $\Sigma^{[k]}\Pi\Sigma\wedge$, no subexponential time PIT for $k \geq 3$ *even* in the whitebox settings is known, see [71, Open Prob. 2]. Our first result exactly addresses this problem and designs a polynomial-time algorithm (Algorithm 1). The technique (we call it DiDI-paradigm, Sec. 1.2) used is very analytic (& "non-ideal" based). Throughout the paper, we will work with $\mathbb{F} = \mathbb{Q}$, though all the results hold for field of large characteristic.

▶ **Theorem 1** (Whitebox $\Sigma\Pi\Sigma\wedge$ PIT). *There is a deterministic, whitebox $s^{O(k\,7^k)}$-time PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits of size s, over $\mathbb{F}[\boldsymbol{x}]$. (See Algorithm 1.)*

▶ Remark.

1. Case $k \leq 2$ can be solved by invoking [71, Thm.5.2]; but $k \geq 3$ was open.
2. Our technique *necessarily* blows up the exponent exponentially in k. In particular, it would be interesting to design a subexponential time algorithm when $k = \Theta(\log s)$.
3. It is not clear if the current technique gives PIT for $\Sigma^{[k]}\Pi\Sigma\wedge^{[2]}$ circuits, i.e. sum of *bi*variate polynomials computed and fed into the top product gate.

Next, we go to the blackbox setting and address two models of interest, namely – $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$, where k, δ are constants. The prior best algorithms were exponential-time in s. Our work builds on previous ideas for *unbounded* top fanin – (1) Jacobian [5], (2) the known blackbox PIT for $\Sigma\wedge\Sigma\wedge$ and $\Sigma\wedge\Sigma\Pi^{[\delta]}$ [37, 25] – maneuvering with an analytic approach, *via* power-series, which unexpectedly *reduces* the top Π-gate to a $\wedge$-gate.

▶ **Theorem 2** (Blackbox PIT for depth-4).

(a) *There is a deterministic $s^{O(k \log\log s)}$-time blackbox PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits of size s, over $\mathbb{F}[\boldsymbol{x}]$.*

(b) *There is a $s^{O(\delta^2\,k\,\log s)}$-time blackbox PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits of size s, over $\mathbb{F}[\boldsymbol{x}]$.*

▶ Remark.

1. Thm. 2 has a *better* dependence on k, but *worse* on s, than Thm. 1. Our results are quasipoly-time even up to $k, \delta = \mathsf{poly}(\log s)$.
2. Thm. 2a is better than Thm. 2b, because $\Sigma\wedge\Sigma\wedge$ has a faster algorithm than $\Sigma\wedge\Sigma\Pi^{[\delta]}$.
3. Even for $\Sigma^{[3]}\Pi\Sigma\wedge$ and $\Sigma^{[3]}\Pi\Sigma\Pi^{[3]}$ models, we leave the *poly*-time blackbox question open.

1.2 Proof ideas: A technical synopsis

In this section, we overview the proof of Theorems 1-2. Both the proofs are analytic, i.e. they use *logarithmic derivative*, and its power-series expansion; greatly transforming the respective models. The first proof is inductive, while the second is a *one-shot* proof. We remark that in both the cases, we essentially reduce to the well-known "wedge" models, that have unbounded top fanin, yet for which PITs are known. This reduction is unforeseeable and quite "power"ful.

The analytic tool that we use, appears in algebra (and complexity theory) through the *formal power series* ring $\mathsf{R}[[x_1, \ldots, x_n]]$ (in short $\mathsf{R}[[\boldsymbol{x}]]$), see [65, 86, 19]. The advantages of the ring $\mathsf{R}[[\boldsymbol{x}]]$ are many. They usually emerge because of the inverse: $(1 - x_1)^{-1} = \sum_{i \geq 0} x_1^i$, which does not make sense in $\mathsf{R}[x]$, but valid in $\mathsf{R}[[\boldsymbol{x}]]$. Other analytic tools used are inspired from Wronskian (aka linear dependence) [51, Thm.7] [46], jacobian (aka algebraic dependence) [12, 5, 67], and logarithmic derivative operator $\mathrm{dlog}_{z_1}(f) = (\partial_{z_1} f)/f$.

Moreover, we will be working with the division operator (eg. $\mathsf{R}(z_1)$, over a certain ring R). The divisions do not come for "free" – they require invertibility with respect to z_1 throughout (again landing us in $\mathsf{R}[[z_1]]$, see Lem. 18). We define class $\mathcal{C}/\mathcal{D} := \{f/g \mid f \in \mathcal{C}, \mathcal{D} \ni g \neq 0\}$, for circuit classes $\mathcal{C}, \mathcal{D}$, (similarly $\mathcal{C} \cdot \mathcal{D}$ denotes the class taking respective products).

The DiDI-technique [Idea of Theorem 1]. The proof of Thm. 1 is recursive and uses a novel technique that we introduce in this work, called DiDI (Di= Divide, D=Derive, I=Induct). We illustrate it in $k = 3$, which generalizes to any k.

Before going into the technicalities, we want to convey that $k = 3$ is *perhaps* the first non-trivial case-study. While $k = 1$ is the *simplest* case (follows directly using sparse-PIT hitting set [49]), $k = 2$ invokes a strong *irreducibility* property [71, Thm. 5.2]; and neither of them work for $k \geq 3$.

The case $k = 3$ asks to check whether $T_1 + T_2 + T_3 \stackrel{?}{=} 0$, where $T_i \in \Pi\Sigma\wedge$ of deg $< d$. We apply a homomorphism $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ such that $x_i \mapsto z_1 \cdot x_i + \Psi(x_i)$ where Ψ is another homomorphism. The map $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$ is a sparse-PIT map s.t. $\Psi(T_i) \neq 0$ for non-zero T_i, using [49], which ensures that the degree of z_2 is polynomially bounded (Theorem 11). Think of the variable z_1 as a degree-*counter* which also helps later to *derive* (the second "D" of DiDI). Observe that Φ is a nonzeroness preserving 1-1 map:

$$T_1 + T_2 + T_3 \neq 0 \iff \Phi(T_1) + \Phi(T_2) + \Phi(T_3) \neq 0.$$

Denote $\mathsf{R} := \mathbb{F}(z_2)[z_1]/\langle z_1^d \rangle$. We divide (first "D" of DiDI), by $\Phi(T_3)$, and derive, wrt z_1, to conclude that $T_1 + T_2 + T_3 = f$ over $\mathbb{F}[\boldsymbol{x}]$ implies

$$\partial_{z_1}\left(\frac{\Phi(T_1)}{\Phi(T_3)}\right) + \partial_{z_1}\left(\frac{\Phi(T_2)}{\Phi(T_3)}\right) = \partial_{z_1}\left(\frac{\Phi(f)}{\Phi(T_3)}\right) \quad \text{over } \mathsf{R}(\boldsymbol{x}) \; .$$

Denote $\widetilde{T_1} := \partial_{z_1}(\Phi(T_1)/\Phi(T_3))$ and $\widetilde{T_2} := \partial_{z_1}(\Phi(T_2)/\Phi(T_3))$. Moreover, $\partial_{z_1}(\Phi(f)/\Phi(T_3)) = 0$, over $\mathsf{R}(\boldsymbol{x})$, if and only if either (1) $\Phi(f)/\Phi(T_3)$ is z_1-free, in which case it is an element of $\mathbb{F}(z_2)$, this can be easily argued by substituting $z_1 = 0$ in the map Φ; or (2) $\mathrm{val}_{z_1}(\partial_{z_1}(\Phi(f)/\Phi(T_3))) \geq d$, which is a contradiction since it implies $\mathrm{val}_{z_1}(\Phi(f)) \geq d + 1$. Here, $\mathrm{val}_{z_1}(\cdot)$ denotes the valuation i.e. the maximum power of z_1 dividing it (which easily extends to fractions via $\mathrm{val}_{z_1}(p/q) := \mathrm{val}_{z_1}(p) - \mathrm{val}_{z_1}(q)$). Whenever we talk about val, think of working over $\mathbb{F}(z_2, \boldsymbol{x})(z_1)$; which is a ring notion that helps us *computationally*, and we track the degree of $\boldsymbol{z}$. This discussion summarizes a crucial fact:

$$T_1 + T_2 + T_3 \neq 0 \iff \widetilde{T_1} + \widetilde{T_2} \neq 0 \text{ over } R(\boldsymbol{x}), \quad \text{or} \quad \left.\frac{\Phi(f)}{\Phi(T_3)}\right|_{z_1=0} \in \mathbb{F}(z_2)\backslash\{0\} \; .$$

We remark that the $z_1 = 0$ substitution is a natural condition as the derivation forgets the $(\bmod\ z_1)$-part. At the core, the idea is really "primal": if a polynomial $g(x) \neq 0$, then either its derivative $g'(x) \neq 0$, or its constant-term $g(0) \neq 0$ (note: $g(0) = g \bmod x$).

Note that, the $z_1 = 0$ substitution part is easy by poly-degree restriction on z_2. If it is already $\neq 0$, we are done, otherwise we need to check $\widetilde{T_1} + \widetilde{T_2} \neq 0$. Rewrite $\widetilde{T_i}$ as $\Phi(T_i)/\Phi(T_3) \cdot \mathrm{dlog}_{z_1}(\Phi(T_i)/\Phi(T_3))$, where dlog denotes the logarithmic-derivative, i.e. $\mathrm{dlog}_{z_1}(\cdot) = \partial_{z_1}(\cdot)/(\cdot)$.

Convert top Π *to* $\wedge$*: version* 1. The map Ψ ensures that $\Phi(T_3)$ is a unit over R. A calculation shows that the action $\mathrm{dlog}(\Sigma\wedge)$ is in $\Sigma\wedge/\Sigma\wedge \in \Sigma\wedge\Sigma\wedge$, over $\mathsf{R}[\boldsymbol{x}]$ (Claim 4). This crucially uses the inverse identity:

$$\frac{1}{1 - a \cdot z_1} = 1 + a \cdot z_1 + \ldots + a^{d-1} \cdot z_1^{d-1} \quad \text{over } \mathsf{R}[\boldsymbol{x}], \tag{1}$$

for $a \in \mathsf{R}[\boldsymbol{x}]$. Since, dlog is additive over a product (Sec. 2), the action puts $\mathrm{dlog}(\Pi\Sigma\wedge/\Pi\Sigma\wedge)$ in $\sum \mathrm{dlog}(\Sigma\wedge)$, so in $\Sigma\wedge\Sigma\wedge$. Thus, both $\widetilde{T}_1$ and $\widetilde{T}_2$ are of the *bloated* form $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}(\boldsymbol{x})$.

Invertibility. The crucial point is that the $\Pi\Sigma\wedge$-circuits are still *invertible* over $R[\boldsymbol{x}]$ as: dlog newly introduces only $\Sigma\wedge\Sigma\wedge$, while the $\Pi\Sigma\wedge$-parts get multiplied by the $\Pi\Sigma\wedge$ within T_i's, which are invertible by Ψ. Thus, such $(\Pi\Sigma\wedge)|_{z_1=0} \in \mathbb{F}(z_2)\backslash\{0\}$; which will be useful later.

Bloated $k = 2$ model. Is the newly "reduced" model similar to $k = 2$ base-case? It is a more general expression $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge)+(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge)$. Let $\widetilde{T}_1+\widetilde{T}_2 =: f_1$, over $\mathsf{R}(\boldsymbol{x})$. We know that $f_1 \neq 0$ (by hypothesis). We again apply "Divide and Derive" of DiDI; here we divide with the $\widetilde{T}_i$ where val_{z_1} is *minimal.* Wlog, $\mathrm{val}_{z_1}(\widetilde{T}_2) =: v$, is the minimal valuation. Of course, $0 \le v < d$ (strict because of Ψ). Let us define $\mathsf{R}' := \mathbb{F}(z_2)[z_1]/\langle z_1^{d-v-1}\rangle$. Then, $(\widetilde{T}_1/\widetilde{T}_2) + 1 = f_1/\widetilde{T}_2$ over $\mathsf{R}'(\boldsymbol{x})$. This is well-defined as the division is being done by the minimum valuation (Lemma 18); thus after derivation, the modulus goes from z_1^d to z_1^{d-v-1} which is well-defined over $\mathsf{R}'(\boldsymbol{x})$. However, if we *derive*: $\partial_{z_1}(f_1/\widetilde{T}_2) =: f_2$ may become $= 0$ over $\mathsf{R}'(\boldsymbol{x})$. That could happen if and only if:

1. Either, $f_1/\widetilde{T}_2$ is z_1-free; in that case

$$\left.\frac{f_1}{\widetilde{T}_2}\right|_{z_1=0} = \left.\left(\frac{\widetilde{T}_1}{\widetilde{T}_2}+1\right)\right|_{z_1=0} \in \mathbb{F}(z_2)\cdot\frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge}+1.$$

   This is easy to test using $\Sigma\wedge\Sigma\wedge$ whitebox PIT (Lemma 19) (we keep track of the circuit-size respectively the degree of z_2 and ensure them polynomially bounded),
2. Or, $\mathrm{val}_{z_1}(f_2) \ge d-v-1 \implies \partial_{z_1}(f_1/\widetilde{T}_2) = z_1^{d-v-1}\cdot p$, for some $p \in \mathsf{R}'(\boldsymbol{x})$ s.t. $\mathrm{val}_{z_1}(p) \ge 0$; this further implies $p \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$ (Lemma 18). Thus $\mathrm{val}_{z_1}(f_1/\widetilde{T}_2) \ge d-v \implies f_1 = 0$, over $\mathsf{R}(\boldsymbol{x})$, a contradiction.

Thus, we check the easy condition (1). If the $z_1 = 0$ substitution outputs 0, we need to check whether other monomials of z_1 in f_2 survive. This suffices to conclude $f \neq 0$. Thankfully $f_2 = \partial_{z_1}(\widetilde{T}_1/\widetilde{T}_2)$ is now a $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ circuit over $\mathsf{R}'(\boldsymbol{x})$. This is the same analysis as above that converts top Π to $\wedge$. Except, we may not be able to remove $\Sigma\wedge\Sigma\wedge$ from the denominator; so we work with this fractional bloated model. (Note: the reciprocal may not be in the polynomial ring $\mathsf{R}'[\boldsymbol{x}]$, but only in the ring $\mathsf{R}'(\boldsymbol{x})$.)

Finally, identity testing of $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}'(\boldsymbol{x})$ is *easy*: (1) $\Sigma\wedge\Sigma\wedge$ is closed under coefficient extraction with respect to z_1 (Lemma 14), (2) whitebox identity testing is in P for both $\Pi\Sigma\wedge$ (Theorem 11) and $\Sigma\wedge\Sigma\wedge$ (convert it to an ROABP using [75] and invoke [70], see Lemma 19), (3) the degree of z_1, z_2 respectively circuit-size remain polynomially bounded.

For general induction, our bloated model is $\Sigma^{[k]}(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ [1]. More work shows that it is *closed* under DiDI-technique. This is primarily what makes our polynomial-time algorithm possible. For details, refer to Section 3.1 and Algorithm 1

Jacobian hits again [Idea of Theorem 2]. Suppose we want to test $T_1 + \ldots + T_k \stackrel{?}{=} 0$, where $T_i \in \Pi\Sigma\Pi^{[\delta]}$ (respec. $\Pi\Sigma\wedge$). We associate a famous polynomial – the Jacobian $J(T_1, \ldots, T_r)$ (see Sec. 2). It captures the algebraic independence of $T_1, \ldots, T_r$ assuming this to be a transcendence basis of the T_i's (see Fact 23). If we could find an r-variate

[1] This is a special case of $\Sigma^{[k]}\Pi\Sigma\wedge\Sigma\wedge$ circuits; which is really depth-6.

linear map Φ, that keeps $T_1, \ldots, T_r$ algebraically independent, then $\Phi(T_1), \ldots, \Phi(T_r)$ are again algebraically independent and it can be shown that for any C: $C(T_1, \ldots, T_k) = 0 \iff C(\Phi(T_1), \ldots, \Phi(T_k)) = 0$ (Fact 22). Such a map is called "faithful" [5].

The overall idea is to find an *explicit* homomorphism $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{x}, z_1, z_2]$, and then fix $\boldsymbol{x}$ by a hitting-set H' to get a *composed* map Ψ' s.t. $\mathrm{rk}_{\mathbb{F}(\boldsymbol{x})} \mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}) = \mathrm{rk}_{\mathbb{F}(\boldsymbol{z})} \Psi'(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$ [here $\mathcal{J}$ is the jacobian matrix and $\boldsymbol{T} = (T_1, \ldots, T_r)$]. Next, *extend* this map to $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{z}, \boldsymbol{y}, t]$ s.t. $x_i \mapsto (\sum_{j=1}^{k} y_j t^{ij}) + \Psi'(x_i)$, which is *faithful.* The construction of the map Ψ' is crucial. We efficiently construct it by reducing $\Psi(J_{\boldsymbol{x}_r}(\boldsymbol{T}))$ to $\Sigma\wedge\Sigma\Pi^{[\delta]}$ (respec. $\Sigma\wedge\Sigma\wedge$) circuits, which have *quasi*poly size hitting sets [25] (respec. Lemma 19).

Jacobian works. A priori, Jacobian is a difficult determinant to work with, and so is finding a faithful Φ. However, for the special models (in this paper) we are able to design Φ, mainly because of two reasons – (1) Jacobian being defined via partial derivatives, has a nice "linearizing effect" on the top Π-gates (that are only $r \le k$ many), (2) Jacobian under a homomorphism Ψ has a nice expression (think of this as a generalized dlog-expression):

$$\Psi(J_{\boldsymbol{x}_r}(\boldsymbol{T})) \;=\; \Psi(T_1 \cdots T_r) \cdot \sum_{g_1 \in L(T_1), \ldots, g_r \in L(T_r)} \frac{\Psi(J_{\boldsymbol{x}_k}(g_1, \ldots, g_r))}{\Psi(g_1 \ldots g_r)} \;. \qquad \text{(see Eqn. 6)}$$

Here, $L(T_i)$ denotes the multiset of sparse polynomials that constitutes T_i. We show: each $1/\Psi(\cdot)$ has a "small" $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit (respec. $\Sigma\wedge\Sigma\wedge$). The last point requires *invertibility.* Define, $\Psi : x_i \mapsto z_1 x_i + \Psi_1(x_i)$, where $\Psi_1(\cdot)$ is a sparse-PIT map s.t. $\Psi_1 : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$ s.t. $\Psi_1(T_i) \neq 0$. Under the Ψ, T_i is a unit over ring $\mathsf{R} := \mathbb{F}(z_2)[z_1]/\langle z_1^D \rangle$, where D is polynomially bounded. The idea behind the map is similar to that of Thm. 1. Next, we sketch why $\Psi(J_{\boldsymbol{x}_r}(\boldsymbol{T}))$ has a $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit (respec. $\Sigma\wedge\Sigma\wedge$) of size $s^{O(k)}$ over $\mathsf{R}[\boldsymbol{x}]$.

Convert top Π to $\wedge$: version 2. The critical point is to show that $1/\Psi(g_1 \cdots g_k)$, over $\mathsf{R}[\boldsymbol{x}]$, where $g_i \in \Sigma\Pi^{[\delta]}$ (respec. $\Sigma\wedge$) has $s^{O(k)}$ size $\Sigma\wedge\Sigma\Pi^{[\delta]}$ (respec. $\Sigma\wedge\Sigma\wedge$) circuit (see Lem. 10): this again follows from the inverse identity Equation 1. We keep track of the degree of $\boldsymbol{z}$ throughout, which eventually is bounded by $s^{O(k)}$. Thus, the H' can be *efficiently* constructed from the hitting set of the respective models (of quasipolynomial size), see Thm. 27 and 19. The map Φ ultimately provides a hitting set for $T_1 + \ldots + T_k$, as we reduce to a PIT of a polynomial over "few" (roughly equal to k) variables, yielding a QP-time algorithm.

It is important to note that there was no power series in [5]; this really empowers the jacobian technique as it now manifests new reduced models, for which a hitting-set is known. This technique is also inherently map-based. So, it requires a hitting-set and *fails* to give a *poly*-time whitebox PIT for the respective models. For the detailed proof, see Section 3.2.

2 Preliminaries

Before proving the results, we describe some of the assumptions and notations used throughout the paper. $\boldsymbol{x}$ denotes $(x_1, \ldots, x_n)$. $[n]$ denotes $\{1, \ldots, n\}$.

Logarithmic derivative. Over a ring R and a variable y, the logarithmic derivative $\mathrm{dlog}_y : \mathsf{R}[y] \to \mathsf{R}(y)$ is defined as $\mathrm{dlog}_y(f) := \partial_y f / f$; here ∂_y denotes the partial derivative with respect to variable y. One important property of dlog is that it is additive over a product as

$$\mathrm{dlog}_y(f \cdot g) \;=\; \frac{\partial_y(f \cdot g)}{f \cdot g} \;=\; \frac{(f \cdot \partial_y g \,+\, g \cdot \partial_y f)}{f \cdot g} \;=\; \mathrm{dlog}_y(f) + \mathrm{dlog}_y(g).$$

We refer this effect as *linearization* of product.

Circuit size. Sparsity $\mathrm{sp}(\cdot)$ refers to the number of nonzero monomials. In this paper, it is a parameter of the circuit size. In particular, $\mathrm{size}(g_1 \cdots g_s) = \sum_{i\in[s]} (\mathrm{sp}(g_i) + \deg(g_i))$, for $g_i \in \Sigma\wedge$ (respec. $\Sigma\Pi^{[\delta]}$). In whitebox settings, we also include the *bit-complexity* of the circuit (i.e. bit complexity of the constants used in the wires) in the size parameter. Some of the complexity parameters of a circuit are *depth* (number of layers), *syntactic degree* (the maximum degree polynomial computed by any node), *fanin* (maximum number of inputs to a node).

Hitting set. A set of points $\mathcal{H} \subseteq \mathbb{F}^n$ is called a *hitting-set* for a class $\mathcal{C}$ of n-variate polynomials if for any nonzero polynomial $f \in \mathcal{C}$, there exists a point in $\mathcal{H}$ where f evaluates to a nonzero value. A $T(n)$-time hitting-set would mean that the hitting-set can be generated in time $T(n)$, for input size n.

Valuation. Valuation is a map $\mathrm{val}_y : \mathsf{R}[y] \to \mathbb{Z}_{\geq 0}$, over a ring R, such that $\mathrm{val}_y(\cdot)$ is defined to be the maximum power of y dividing the element. It can be easily extended to fraction field $\mathsf{R}(y)$, by defining $\mathrm{val}_y(p/q) := \mathrm{val}_y(p) - \mathrm{val}_y(q)$; where it can be negative.

Field. We denote the underlying field as $\mathbb{F}$ and assume that it is of characteristic 0. All our results hold for other fields (eg. $\mathbb{Q}_p, \mathbb{F}_p$) of *large* characteristic (see Remarks in Section 3.1–3.2).

Jacobian. The Jacobian of a set of polynomials $\mathbf{f} = \{f_1, \ldots, f_m\}$ in $\mathbb{F}[\boldsymbol{x}]$ is defined to be the matrix $\mathcal{J}_{\boldsymbol{x}}(\mathbf{f}) := \left(\partial_{x_j}(f_i)\right)_{m\times n}$. Let $S \subseteq \boldsymbol{x} = \{x_1, \ldots, x_n\}$ and $|S| = m$. Then, polynomial $J_S(\mathbf{f})$ denotes the minor (i.e. determinant of the submatrix) of $\mathcal{J}_{\boldsymbol{x}}(\mathbf{f})$, formed by the columns corresponding to the variables in S. For its useful properties, see Appendix C.

3 Proof of the main theorems

This section proves Theorems 1-2. The proofs are self contained and we assume for the sake of simplicity that the underlying field $\mathbb{F}$ has characteristic 0. When this is not the case, we discuss the corresponding required characteristic as remarks after the respective proofs.

3.1 Proof of Theorem 1

As seen in Section 1.2, we will induct over the bloated model which naturally generalizes $\Sigma\Pi\Sigma\wedge$ circuits and works ideally under the DiDI-techniques. Formally, we define it below.

▶ **Definition 3.** *We call a circuit $\mathcal{C} \in \mathsf{Gen}(k, s)$, over $\mathsf{R}(\boldsymbol{x})$, for any ring R, with parameter k and size-s, if $\mathcal{C} \in \Sigma^{[k]}(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$. It computes $f \in \mathsf{R}(\boldsymbol{x})$, if $f = \sum_{i=1}^{k} T_i$, where*

1. *$T_i =: (U_i/V_i)\cdot(P_i/Q_i)$, for $U_i, V_i \in \Pi\Sigma\wedge$, and $P_i, Q_i \in \Sigma\wedge\Sigma\wedge$,*
2. *$\mathrm{size}(T_i) = \mathrm{size}(U_i) + \mathrm{size}(V_i) + \mathrm{size}(P_i) + \mathrm{size}(Q_i)$, and $\mathrm{size}(f) = \sum_{i\in[k]} \mathrm{size}(T_i)$.*

Eg. Size-s $\Sigma^{[k]}\Pi\Sigma\wedge$-circuit $\in \mathsf{Gen}(k, s)$. We will design a *recursive* algorithm.

Proof of Theorem 1. Begin with $T_{i,0} := T_i$ and $f_0 := f$ where $T_{i,0} \in \Pi\Sigma\wedge$; $\sum_i T_{i,0} = f_0$, and f_0 has size $\leq s$. Assume $\deg(f) < d \leq s$; we keep the parameter d separately, to help optimize the complexity later. In every recursive call we work with $\mathsf{Gen}(\cdot,\cdot)$ circuits. As the input case, define $U_{i,0} := T_{i,0}$ and $V_{i,0} := P_{i,0} := Q_{i,0} := 1$. Further define a 1-1 homomorphism $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ such that $x_i \mapsto z_1\cdot x_i + \Psi(x_i)$. Here, $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$ is a sparse-PIT map [49] s.t. $\Psi(U_{i,0}) \neq 0$, $\forall i \in [k]$ (Theorem 11). Invertibility implies that

$f_0 = 0 \iff \Phi(f_0) = 0$. Further, the degree bound of z_2 on $\Phi(T_{i,0})$ is $\mathsf{poly}(s)$. The algorithm is recursive, and first reduces the identity testing from top-fanin k to $k-1$. Note: $k = 1$ is trivial.

0-th step. Efficient reduction from k to $k-1$. By assumption, $\sum_{i=1}^{k} T_{i,0} = f_0$ and $T_{k,0} \neq 0$. Apply Φ both sides. Then divide and derive:

$$\begin{aligned}
\sum_{i\in[k]} T_{i,0} \;=\; f_0 \;\iff\; & \sum_{i\in[k]} \Phi(T_{i,0}) \;=\; \Phi(f_0) \\
\iff\; & \sum_{i\in[k-1]} \frac{\Phi(T_{i,0})}{\Phi(T_{k,0})} + 1 \;=\; \frac{\Phi(f_0)}{\Phi(T_{k,0})} \\
\implies\; & \sum_{i\in[k-1]} \partial_{z_1}\left(\frac{\Phi(T_{i,0})}{\Phi(T_{k,0})}\right) \;=\; \partial_{z_1}\left(\frac{\Phi(f_0)}{\Phi(T_{k,0})}\right) \\
\iff\; & \sum_{i=1}^{k-1} \frac{\Phi(T_{i,0})}{\Phi(T_{k,0})} \cdot \operatorname{dlog}\left(\frac{\Phi(T_{i,0})}{\Phi(T_{k,0})}\right) \;=\; \partial_{z_1}\left(\frac{\Phi(f_0)}{\Phi(T_{k,0})}\right) . \qquad (2)
\end{aligned}$$

Define the following:

- $\mathsf{R}_1 := \mathbb{F}(z_2)[z_1]/\langle z_1^d \rangle$. Note that, Eqn.(2) holds over $\mathsf{R}_1(\boldsymbol{x})$.
- $\widetilde{T}_{i,1} := \Phi(T_{i,0})/\Phi(T_{k,0}) \cdot \operatorname{dlog}(\Phi(T_{i,0})/\Phi(T_{k,0}))$, $\forall\, i \in [k-1]$.
- $f_1 := \partial_{z_1}(\Phi(f_0)/\Phi(T_{k,0}))$, over $\mathsf{R}_1(\boldsymbol{x})$.

Definability of $T_{i,1}$ and f_1. It is easy to see that these are well-defined terms. Here, we emphasize that we do not exactly compute/store $\widetilde{T}_{i,1}$ as a fraction where the degree in z_1 is $< d$; instead it is computed/stored as an element in $\mathbb{F}(z_2)(z_1, \boldsymbol{x})$, where z_1 is a formal variable. Formally, we compute $T_{i,1} \in \mathbb{F}(z_2)(z_1, \boldsymbol{x})$, such that $\widetilde{T}_{i,1} = T_{i,1}$, over $\mathsf{R}_1(\boldsymbol{x})$. We keep track of the degree of z_1 and z_2 in $T_{i,1}$. Thus, $\sum_{i\in[k-1]} T_{i,1} = f_1$, over $\mathsf{R}_1(\boldsymbol{x})$.

The "iff" condition. Equality in Eqn. (2) over $\mathsf{R}_1(\boldsymbol{x})$ is *one-sided*; however we want a $\iff$ condition to efficiently reduce the identity testing. Note that $f_1 \neq 0$ implies $\operatorname{val}_{z_1}(f_1) < d =: d_1$. By assumption, $\Phi(T_{k,0})$ is invertible over $\mathsf{R}_1(\boldsymbol{x})$. Further, $f_1 = 0$, over $\mathsf{R}_1(\boldsymbol{x})$, implies –

1. Either, $\Phi(f_0)/\Phi(T_{k,0})$ is z_1-free. This implies $\Phi(f_0)/\Phi(T_{k,0}) \in \mathbb{F}(z_2)(\boldsymbol{x})$, which further implies it is in $\mathbb{F}(z_2)$, because of the map Φ (z_1-free implies $\boldsymbol{x}$-free, by substituting $z_1 = 0$). Also, note that $f_0, T_{k,0} \neq 0$ implies $\Phi(f_0)/\Phi(T_{k,0})$ is a *nonzero* element in $\mathbb{F}(z_2)$. Thus, it suffices to check whether $\Phi(f_0)|_{z_1=0} = \Psi(f_0)$ is non-zero or not. Further, the degree of z_2 in $\Psi(f_0)$ is polynomially bounded.

2. Or, $\partial_{z_1}(\Phi(f_0)/\Phi(T_{k,0})) = z_1^{d_1} \cdot p$ where $p \in \mathbb{F}(z_2)(z_1, \boldsymbol{x})$ s.t. $\operatorname{val}_{z_1}(p) \geq 0$. By simple power series expansion, one can conclude that $p \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$ (Lemma 18). Hence, $\Phi(f_0)/\Phi(T_{k,0}) = z_1^{d_1+1} \cdot q$ where $q \in F(z_2, \boldsymbol{x})[[z_1]]$, i.e.

$$\Phi(f_0)/\Phi(T_{k,0}) \in \langle z_1^{d_1+1} \rangle_{\mathbb{F}(z_2,\boldsymbol{x})[[z_1]]} \implies \operatorname{val}_{z_1}(\Phi(f_0)) \geq d+1,$$

a contradiction.

Conversely, it is obvious that $f_0 = 0$ implies $f_1 = 0$. Thus, we have proved the following

$$\sum_{i\in[k]} T_{i,0} \neq 0 \;\text{ over } \mathbb{F}[\boldsymbol{x}] \iff \sum_{i\in[k-1]} T_{i,1} \neq 0 \;\text{ over } \mathsf{R}_1(\boldsymbol{x}), \;\text{ or }, \; 0 \neq \Phi(f_0)|_{z_1=0} \in \mathbb{F}(z_2)\,.$$

Eventually, we show that $T_{i,1} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}_1(\boldsymbol{x})$, with polynomial blowup in size (Claim 4). So, the above circuit is in $\mathsf{Gen}(k-1,\cdot)$, over $\mathsf{R}_1(\boldsymbol{x})$, which we recurse on to finally give the identity testing. The 1-th step is a bit more tricky:

Induction step. Assume that we are in the j-th step ($j \geq 1$). Our induction hypothesis assumes –

1. $\sum_{i\in[k-j]} T_{i,j} = f_j$, over $\mathsf{R}_j(\boldsymbol{x})$, where $\mathsf{R}_j := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j}\rangle$, and $T_{i,j} \neq 0$.
2. Here, $T_{i,j} =: (U_{i,j}/V_{i,j})\cdot(P_{i,j}/Q_{i,j})$, where $U_{i,j}, V_{i,j} \in \Pi\Sigma\wedge$, and $P_{i,j}, Q_{i,j} \in \Sigma\wedge\Sigma\wedge$, each in $\mathsf{R}_j[\boldsymbol{x}]$. Think of them being computed as $\mathbb{F}(z_2)(z_1, \boldsymbol{x})$, with the degrees being tracked. Wlog, assume that $\mathrm{val}_{z_1}(T_{k-j,j})$ is the minimal among all $T_{i,j}$'s.
3. $\mathrm{val}_{z_1}(T_{i,j}) \geq 0, \forall i \in [k-j]$. Moreover, $U_{i,j}|_{z_1=0} \in \mathbb{F}(z_2)\backslash\{0\}$ (similarly $V_{i,j}$).
4. $f \neq 0$, over $\mathbb{F}[\boldsymbol{x}] \iff f_j \neq 0$, over $\mathsf{R}_j(\boldsymbol{x})$, or, $\bigvee_{i=0}^{j-1}\left((f_i/T_{k-i,i})|_{z_1=0} \neq 0, \text{over}\, \mathbb{F}(z_2)(\boldsymbol{x})\right)$.

We follow the 0-th step, without applying any further homomorphism. Note that the "or condition" in the last hypothesis is similar to the $j=0$ case except that there is no Φ: this is because $\Phi(f_0)|_{z_1=0} \neq 0 \iff \Phi(f_0/T_{k,0})|_{z_1=0} \neq 0$. This condition just separates the derivative from the constant-term (as pointed in Section 1.2).

Let $\mathrm{val}_{z_1}(P_{i,j}/Q_{i,j}) =: v_{i,j}$, for $i \in [k-j]$. Note that

$$\min_i \mathrm{val}_{z_1}(T_{i,j}) = \min_i \mathrm{val}_{z_1}(P_{i,j}/Q_{i,j}) = v_{k-j,j}$$

since $\mathrm{val}_{z_1}(U_{i,j}) = \mathrm{val}_{z_1}(V_{i,j}) = 0$ (else we reorder). We remark that $0 \leq v_{i,j} < d_j$ for all i's in j-th step; upper-bound is strict, since otherwise $T_{i,j} = 0$ over $\mathsf{R}_j(x)$.

Min val computation is easy. Finding this min val is *easy*, as we can compute $\mathrm{val}_{z_1}(P_{i,j})$ and $\mathrm{val}_{z_1}(Q_{i,j})$, $\forall i \in [k-j]$. To compute val, note that $\mathrm{coef}_{z_1^e}(P_{i,j})$ and $\mathrm{coef}_{z_1^e}(Q_{i,j})$ are in $\Sigma\wedge\Sigma\wedge$ as well, over $F(z_2)[\boldsymbol{x}]$ (Lemma 14). We can keep track of z_1 degree and thus interpolate to find the minimum $e < d_j$ such that it is $\neq 0$ (implying it to be the respective val).

Efficient reduction from $k-j$ to $k-j-1$. Similar to the 0-th step, we divide and derive:

$$\begin{aligned}
\sum_{i\in[k-j]} T_{i,j} \;=\; f_j \iff & \sum_{i\in[k-j-1]} T_{i,j}/T_{k-j,j} \;+\; 1 \;=\; f_j/T_{k-j,j} \\
\implies & \sum_{i\in[k-j-1]} \partial_{z_1}(T_{i,j}/T_{k-j,j}) \;=\; \partial_{z_1}(f_j/T_{k-j,j}) \\
\iff & \sum_{i=1}^{k-j-1} T_{i,j}/T_{k-j,j}\cdot \mathrm{dlog}(T_{i,j}/T_{k-j,j}) \;=\; \partial_{z_1}(f_j/T_{k-j,j}) \qquad (3)
\end{aligned}$$

Define the following:

- $\mathsf{R}_{j+1} := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_{j+1}}\rangle$, where $d_{j+1} := d_j - v_{k-j,j} - 1$.
- $\widetilde{T}_{i,j+1} := T_{i,j}/T_{k-j,j}\cdot\mathrm{dlog}(T_{i,j}/T_{k-j,j})$, $\forall\, i \in [k-j-1]$.
- $f_{j+1} := \partial_{z_1}(f_j/T_{k-j,j})$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$.

Definability of $T_{i,j+1}$ and f_{j+1}. By the minimal valuation assumption, it follows that $\mathrm{val}(f_j) \geq v_{k-j,j}$, and thus $\widetilde{T}_{i,j+1}$ and f_{j+1} are all well-defined over $\mathsf{R}_{j+1}(\boldsymbol{x})$. Note that, Eqn. (3) holds over $\mathsf{R}_{j+1}(\boldsymbol{x})$ as $d_{j+1} < d_j$ (because, whatever identity holds true mod$z_1^{d_j}$ must hold mod$z_1^{d_{j+1}}$ as well). Hence, we must have $\sum_{i=1}^{k-j-1} \widetilde{T}_{i,j+1} = f_{j+1}$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$ [proving induction hypothesis (1)].

Similarly, we emphasize on the fact that we do not exactly compute $\widetilde{T}_{i,j+1}$ mod $z_1^{d_{j+1}}$; instead it is computed as a fraction in $\mathbb{F}(z_2)(z_1, \boldsymbol{x})$, with formal z_1. Formally, we compute/store $T_{i,j+1} \in \mathbb{F}(z_2)(z_1, \boldsymbol{x})$, such that $\widetilde{T}_{i,j+1} = T_{i,j+1}$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$. We keep track of the degree of z_1 and z_2 in $T_{i,j+1}$. Also, by definition, $\mathrm{val}_{z_1}(T_{i,j+1}) \geq 0$ (as we divide by the min val) [proving induction hypothesis (3), first part]. Of course, we have $\sum_{i\in[k-j-1]} T_{i,j+1} = f_{j+1}$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$.

The "iff" condition. The above Eqn. (3) pioneers to reduce from $k-j$-summands to $k-j-1$. But we want a $\iff$ condition to efficiently reduce the identity testing. If $f_{j+1} \neq 0$, then $\mathrm{val}_{z_1}(f_{j+1}) < d_{j+1}$. Further, $f_{j+1} = 0$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$ implies–

1. Either, $f_j/T_{k-j,j}$ is z_1-free. This implies it is in $\mathbb{F}(z_2)(\boldsymbol{x})$. Now, if indeed $f_0 \neq 0$, then the computed $T_{i,j}$ as well as f_j must be non-zero over $\mathbb{F}(\boldsymbol{z}_2)(z_1, \boldsymbol{x})$, by induction hypothesis (as they are non-zero over $\mathsf{R}_j(\boldsymbol{x})$). However,
$$\left(\frac{T_{i,j}}{T_{k-j,j}}\right)\Bigg|_{z_1=0} = \left(\frac{U_{i,j}\cdot V_{k-j,j}}{U_{k-j,j}\cdot V_{i,j}}\right)\Bigg|_{z_1=0} \cdot \left(\frac{P_{i,j}\cdot Q_{k-j,j}}{P_{k-j,j}\cdot Q_{i,j}}\right)\Bigg|_{z_1=0} \in \mathbb{F}(z_2)\cdot\left(\frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge}\right).$$
Thus,
$$\frac{f_j}{T_{k-j,j}} \in \sum \mathbb{F}(z_2)\cdot\left(\frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge}\right) \in \left(\frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge}\right).$$
Here we crucially use that $\Sigma\wedge\Sigma\wedge$ is closed under multiplication (Lemma 16). We show that the degree of z_2 (in denominator and numerator) in each $T_{i,j}/T_{k,j}$ is poly-bounded. Thus, this identity testing can be done in poly-time (Lemma 19). For, detailed time-complexity and calculations, see Claim 4 and its subsequent paragraph.
2. Or, $\partial_{z_1}(f_j/T_{k-j,j}) = z_1^{d_{j+1}}\cdot p$, where $p \in \mathbb{F}(z_2)(z_1, \boldsymbol{x})$ s.t. $\mathrm{val}_{z_1}(p) \geq 0$. By a simple power series expansion, one concludes that $p \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$ (Lemma 18). Hence, one concludes that
$$\frac{f_j}{T_{k-j,j}} \in \left\langle z_1^{d_{j+1}+1}\right\rangle_{\mathbb{F}(z_2,\boldsymbol{x})[[z_1]]} \implies \mathrm{val}_{z_1}(f_j) \geq d_j,$$
i.e. $f_j = 0$, over $\mathsf{R}_j(\boldsymbol{x})$.

Conversely, $f_j = 0$, over $\mathsf{R}_j(\boldsymbol{x})$, implies
$$\mathrm{val}_{z_1}(f_j) \geq d_j \implies \mathrm{val}_{z_1}\left(\partial_{z_1}\left(\frac{f_j}{T_{k-j,j}}\right)\right) \geq d_j - v_{k-j,j} - 1 \implies f_{j+1} = 0, \text{ over } \mathsf{R}_{j+1}(\boldsymbol{x}).$$
Thus, we have proved that $\sum_{i\in[k-j]} T_{i,j} \neq 0$ over $\mathsf{R}_j(\boldsymbol{x})$ iff
$$\sum_{i\in[k-j-1]} T_{i,j+1} \neq 0 \text{ over } \mathsf{R}_{j+1}(\boldsymbol{x}) \text{ , or , } 0 \neq \left(\frac{f_j}{T_{k-j,j}}\right)\Bigg|_{z_1=0} \in \mathbb{F}(z_2)(\boldsymbol{x}) \text{ .}$$
Therefore induction hypothesis (4) holds. All we need to show is hypothesis (2) and second part of (3). This part is involved in the size-analysis and dlog-computation, discussed below.

Invertibility of $\Pi\Sigma\wedge$-circuits. Before going into the size analysis, we want to remark that the dlog compuation plays a crucial role here. The action $\text{dlog}(\Sigma\wedge\Sigma\wedge) \in \Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$, is of poly-size (Lemma 17). What is the action on $\Pi\Sigma\wedge$? dlog distributes the product additively, so it suffices to work with $\text{dlog}(\Sigma\wedge)$; and we show that $\text{dlog}(\Sigma\wedge) \in \Sigma\wedge\Sigma\wedge$ of poly-size. Assuming these, we simplify

$$\frac{T_{i,j}}{T_{k-j,j}} = \frac{U_{i,j}\cdot V_{k-j,j}}{V_{i,j}\cdot U_{k-j,j}} \cdot \frac{P_{i,j}\cdot Q_{k-j,j}}{Q_{i,j}\cdot P_{k-j,j}},$$

and its dlog. Thus, using Eq. (3), $U_{i,(j+1)}$ grows to $U_{i,j}\cdot V_{k-j,j}$ (and similarly $V_{i,(j+1)}$). This also means: $U_{i,(j+1)}|_{z_1=0} \in \mathbb{F}(z_2)\setminus\{0\}$ (proving hypothesis (3), second part).

Size analysis. We will show that $T_{i,j+1} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$, with only polynomial blowup in size. Let $\text{size}(T_{i,j}) \le s_j$, for $i \in [k-j]$, and $j \in [k]$. Note that, by assumption, $s_0 \le s$.

▷ Claim 4 (Final size). $T_{1,k-1} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ of size $s^{O(k7^k)}$, over $\mathsf{R}_{k-1}(\boldsymbol{x})$.

Proof. Steps $j=0$ and $j>0$ are slightly different because of the Φ. However the main idea of using power-series is the same which eventually shows that $\text{dlog}(\Sigma\wedge) \in \Sigma\wedge\Sigma\wedge$.

We first deal with $j=0$. Let $A - z_1\cdot B = \Phi(g) \in \Sigma\wedge$, for some $A \in \mathbb{F}(z_2)$ and $B \in \mathsf{R}_1[\boldsymbol{x}]$. Note that $A \neq 0$ because of the map Ψ. Further, $\text{size}(B) \le O(d\cdot\text{size}(g))$, as a single monomial of the form x^e can produce $d+1$-many monomials. Over $\mathsf{R}_1(\boldsymbol{x})$,

$$\text{dlog}(\Phi(g)) = -\frac{\partial_{z_1}(B\cdot z_1)}{A(1-\frac{B}{A}\cdot z_1)} = -\frac{\partial_{z_1}(B\cdot z_1)}{A}\cdot\sum_{i=0}^{d_1-1}\left(\frac{B}{A}\right)^i\cdot z_1^i\,. \tag{4}$$

B^i has a trivial $\wedge\Sigma\wedge$-circuit of size $O(d\cdot\text{size}(g))$. Also, $\partial_{z_1}(B\cdot z_1)$ has a $\Sigma\wedge$-circuit of size at most $O(d\cdot\text{size}(g))$. Using waring identity (Lemma 15), we get that each $\partial_{z_1}(B\cdot z_1)\cdot(B/A)^i\cdot z_1^i$ has size $O(i\cdot d\cdot\text{size}(g))$, over $\mathsf{R}_1(\boldsymbol{x})$. Summing over $i \in [d_1-1]$, the overall size is at most $O(d_1^2\cdot d\cdot\text{size}(g)) = O(d^3\cdot\text{size}(g))$, as $d_0 = d_1 = d$.

For the j-th step, we emphasize that the degree could be larger than d. Assume that syntactic degree of denominator and numerator of $T_{i,j}$ (each in $\mathbb{F}[\boldsymbol{x},\boldsymbol{z}]$) are bounded by D_j (it is *not* d_j as seen above; this is to save on the trouble of mod-computation at each step). Of course, $D_0 < d \le s$.

For $j>0$, the above summation in Equation 4 is over $\mathsf{R}_j(\boldsymbol{x})$. However the degree could be D_j (possibly more than d_j) of the corresponding A and B. Thus, the overall size after the power-series expansion would be $O(D_j^2\cdot d\cdot\text{size}(g))$.

Using Lemma 17, we can show that $\text{dlog}(P_{i,j}) \in \Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ (similarly for $Q_{i,j}$), of size $O(D_j^2\cdot s_j)$. Also $\text{dlog}(U_{i,j}\cdot V_{k-j,j}) \in \sum\text{dlog}(\Sigma\wedge)$, i.e. sum of action of dlog on $\Sigma\wedge$ (since dlog linearizes product); and it can be computed by the above formulation. Thus, $\text{dlog}(T_{i,j}/T_{k-j,j})$ is a sum of 4-many $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ of size at most $O(D_j^2\,s_j)$ and 1-many $\Sigma\wedge\Sigma\wedge$ of size $O(D_j^2 d_j s_j)$ (from the above power-series computation) [Note: we summed up the $\Sigma\wedge\Sigma\wedge$-expressions from $\text{dlog}(\Sigma\wedge)$ together]. Additionally the syntactic degree of each denominator and numerator (of the $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$) is $O(D_j)$. We rewrite the 4 expressions (each of $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$) and express it as a single $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ using waring identity (Lemma 16), with the size blowup of $O(D_j^{12}\,s_j^4)$; here the syntatic degree blowsup to $O(D_j)$. Finally we add the remaining $\Sigma\wedge\Sigma\wedge$ circuit (of size $O(D_j^3 s_j)$ and degree $O(dD_j)$) to get $O(s_j^5 D_j^{16} d)$. To bound this, we need to understand the degree bound D_j.

Finally we need to multiply $T_{i,j}/T_{k-j,j} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ where each $\Sigma\wedge\Sigma\wedge$ is a product of two $\Sigma\wedge\Sigma\wedge$ expression of size s_j and syntactic degree D_j; clubbed together owing a blowup of $O(D_j \cdot s_j^2)$. Hence multiplying it with $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ expression obtained from dlog computation above gives size blowup of $s_{j+1} = s^7 \cdot D_j^{O(1)} \cdot d$.

Computing $T_{i,j}/T_{k-j,j}$ increases the syntactic degree "slowly"; which is much less than the size blowup. As mentioned before, the deg-blowup in dlog-computation is $O(dD_j)$ and in the clearing of four expressions, it is just $O(D_j)$. Thus, $D_{j+1} = O(dD_j) \implies D_j = d^{O(j)}$.

The recursion on the size is $s_{j+1} = s_j^7 \cdot d^{O(j)}$. Using $d \le s$ we deduce, $s_j = (sd)^{O(j\cdot 7^j)}$. In particular, s_{k-1}, size after $k-1$ steps is $s^{O(k\cdot 7^k)}$. This computation quantitatively establishes induction hypothesis (2). ◁

Final time complexity. The above proof actually shows that $T_{1,k-1}$ has a "bloated" circuit of size $s^{O(k\cdot 7^k)}$ over $\mathsf{R}_{k-1}(\boldsymbol{x})$; and that the degree bound on z_2 and z_1 (over $\mathbb{F}(z_2)[z_1,\boldsymbol{x}]$, keeping denominator and numerator "in place") is $D_{k-1} = d^{O(k)}$. We note that whitebox PIT for both $\Pi\Sigma\wedge$ and $\Sigma\wedge\Sigma\wedge$ is in poly-time (using Thm. 11 & Lem. 19 respectively), and the proof above is constructive: we calculate $U_{i,j+1}$ (and other terms) from $U_{i,j}$ explicitly. Thus, this part can be done in $s^{O(k7^k)}$ time.

What remains is to test the $z_1 = 0$-part of induction hypothesis (4); it could *short-circuit* the recursion much before $j = k-1$. As we mentioned before, in this case, we need to do a PIT on $\Sigma\wedge\Sigma\wedge$ only. At the j-th step, when we substitute $z_1 = 0$, the size of each $T_{i,j}$ can be at most s_j (by definition). We need to do PIT on a simpler model: $\sum^{[k-j]} \mathbb{F}(z_2)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$. We can clear out and express this as a single $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ expression; with a size blowup of $s_j^{O(k-j)} \le (sd)^{O(j(k-j)7^j)}$. Further, use the fact that $\max_{j\in[k-1]} j(k-j)7^j = (k-1)7^{k-1}$ (see Lemma 20). The degree bound on z_2 remains as before. Finally, use Lemma 19 for the base-case whitebox PIT. Thus, the final time complexity is $s^{O(k\cdot 7^k)}$.

Here we also remark that in $z_1 = 0$ substitution $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ may be undefined. However, we keep track of z_1 degree of numerator and denominator, which will be polynomially bounded as seen in the discussion above. We can easily interpolate and cancel the z_1 power to make it work.

Bit complexity. It is routine to show that the bit-complexity is really what we claim. Initially, the given circuit has bit-complexity s. The main blowup happens due to the dlog-computation which is a poly-size blowup. We also remark that while using Lemma 16 (using Lemma 15), we *may* need to go to a field extension of at most $s^{O(k)}$ (because of the $\varepsilon(i)$ and correspondingly the constants $\gamma_{\varepsilon(2),\dots,\varepsilon(k)}$, but they still are $s^{O(k)}$-bits). Also, Theorem 11 and Lemma 19 computations blowup bit-complexity polynomially. This concludes the proof. ◀

▶ Remark.

1. The above method does *not* give whitebox PIT (in poly-time) for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$, as we donot know poly-time whitebox PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$. However, the above methods do show that whitebox-PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ polynomially *reduces* to whitebox-PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$.
2. DiDI-technique can be used to give whitebox PIT for the general bloated model $\mathsf{Gen}(k,s)$.
3. The above proof works when the characteristic is $\ge d$. This is because the nonzeroness remains *preserved* after derivation wrt z_1.

3.2 Proof of Theorem 2

Here we prove Theorem 2b only. The proof technique of part (a) has analogous calculations (using bottom $\Sigma\wedge$ instead of $\Sigma\Pi^{[\delta]}$); see Appendix D. The main idea is to use the Jacobian [5]. In fact, it solves a more general model than $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$.

Transcendence basis. Polynomials $T_1, \ldots, T_m$ are called *algebraically dependent* if there exists a nonzero *annihilator* A s.t. $A(T_1, \ldots, T_m) = 0$. *Transcendence degree* is the size of the largest subset $S \subseteq \{T_1, \ldots, T_m\}$ that is algebraically independent. Then S is called a *transcendence basis.*

► **Problem 5.** *Let $\{T_i \,|\, i \in [m]\}$ be $\Pi\Sigma\Pi^{[\delta]}$ circuits of (syntactic) degree at most d and size s. Let the transcendence degree of T_i's,* $\text{trdeg}_{\mathbb{F}}(T_1, \ldots, T_m) = k \ll s$. *Further, $C(x_1, \ldots, x_m)$ be a circuit of* $(\text{size} + \text{deg}) < s'$. *Design a blackbox-PIT algorithm for $C(T_1, \ldots, T_m)$.*

Trivially, $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ is a very special case of the above setting. Let $\boldsymbol{T} := \{T_1, \ldots, T_m\}$. Let $\boldsymbol{T}_k := \{T_1, \ldots, T_k\}$ be a transcendence basis. For $T_i = \prod_j g_{ij}$, we denote the set $L(T_i) := \{g_{ij} \mid j\}$.

We want to find an explicit homomorphism $\Psi : \mathbb{F}[\boldsymbol{x}] \to \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ s.t. $\Psi(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$ is of a "nice" form. In the image we fix $\boldsymbol{x}$ suitably, to get a composed map $\Psi' : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_1, z_2]$ s.t. $\text{rk}_{\mathbb{F}(\boldsymbol{x})}\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}) = \text{rk}_{\mathbb{F}(\boldsymbol{z})}\Psi'(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$. Then, we can extend this map to $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{z}, \boldsymbol{y}, t]$ s.t. $x_i \mapsto (\sum_{j=1}^{k} y_j t^{ij}) + \Psi'(x_i)$, which is *faithful* [5, Lemma 2.7]; see Lemma 24. We show that the map Φ can be efficiently constructed using a scaling and shifting map (Ψ) which is eventually fixed by the hitting set (H' defining Ψ') of a $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit. Overall, $\Phi(f)$ is a $k + 3$-variate polynomial for which a trivial hitting set exists.

Wlog, $\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T})$ is full rank with respect to the variable set $\boldsymbol{x}_k = (x_1, \ldots, x_k)$. Thus, by assumption, $J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) \neq 0$ (for notation, see Section 2). We want to construct a Ψ s.t. $\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k))$ has an "easier" PIT. We have the following identity [5, Eqn. 3.1], from the linearity of the determinant, and the simple observation that $\partial_x(T_i) = T_i \cdot \left(\sum_j \partial_x(g_{ij})/g_{ij}\right)$, where $T_i = \prod_j g_{ij}$:

$$J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) = \sum_{g_1 \in L(T_1), \ldots, g_k \in L(T_k)} \left(\frac{T_1 \ldots T_k}{g_1 \ldots g_k}\right) \cdot J_{\boldsymbol{x}_k}(g_1, \ldots, g_k) \,. \tag{5}$$

The homomorphism Ψ. Define $\Psi : \mathbb{F}[\boldsymbol{x}] \to \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ as $x_i \mapsto z_1 \cdot x_i + \Psi_1(x_i)$, where $\Psi_1 : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$, is a *sparse-PIT* map. The importance of Ψ_1 is to ensure that $\Psi_1(g) \neq 0$, $\forall g \in \bigcup_i L(T_i)$. As $\deg(g) \leq \delta$, $\text{sp}(g) \leq \binom{n+\delta}{\delta}$, . Thus, [49] (Theorem 11) gives the upper bound:

$$\deg_{z_2}(\Psi(g)) \leq \delta \cdot \left(\binom{n+\delta}{\delta} \cdot n \cdot \log \delta\right)^2 =: D_1.$$

Denote the ring $\mathsf{R}[\boldsymbol{x}]$ where $\mathsf{R} := \mathbb{F}(z_2)[z_1]/\langle z_1^D \rangle$, and $D := k \cdot (d - 1) + 1$. Being 1-1, Ψ is clearly a non-zero preserving map. Moreover,

▷ Claim 6. $J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) = 0 \iff \Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) = 0$, over $\mathsf{R}[\boldsymbol{x}]$.

Proof. As $\deg(T_i) \leq d$, each entry of the matrix can be of degree at most $d - 1$; therefore $\deg(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) \leq k(d - 1) = D - 1$. Thus, $\deg_{z_1}(\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k))) < D$. Hence, the conclusion. ◁

Eqn. (5) implies that

$$\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) \;=\; \Psi(T_1\cdots T_k)\cdot \sum_{g_1\in L(T_1),\ldots,g_k\in L(T_k)} \frac{\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))}{\Psi(g_1\ldots g_k)}\;. \tag{6}$$

As T_i has product fanin s, the top-fanin in the sum in Eqn. (6) can be at most s^k. Then define,

$$\widetilde{F} \;:=\; \sum_{g_1\in L(T_1),\ldots,g_k\in L(T_k)} \frac{\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))}{\Psi(g_1\ldots g_k)}\;,\quad \text{over } \mathsf{R}[\boldsymbol{x}]. \tag{7}$$

Well-definability of $\widetilde{\boldsymbol{F}}$. Note that,

$$\Psi(g_i) \equiv \Psi_1(g_i) \bmod z_1 \neq 0 \implies 1/\Psi(g_1\cdots g_k) \in \mathbb{F}(z_2)[[\boldsymbol{x}, z_1]].$$

Thus, RHS is an element in $\mathbb{F}(z_2)[[\boldsymbol{x}, z_1]]$ and taking mod z_1^D it is in $\mathsf{R}[\boldsymbol{x}]$. We remark that instead of minimally reducing mod z_1^D, we will work with an $F \in \mathbb{F}(z_2)[z_1, \boldsymbol{x}]$ such that $F = \tilde{F}$ over $\mathsf{R}[\boldsymbol{x}]$. Further, we ensure that the degree of $\boldsymbol{z}$ is polynomially bounded.

▷ Claim 7. Over $\mathsf{R}[\boldsymbol{x}]$, $\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) = 0 \iff F = 0$.

Proof sketch. This follows from the invertibility of $\Psi(T_1\cdots T_k)$ in $R[\boldsymbol{x}]$. ◁

The hitting set $\boldsymbol{H'}$. By $J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) \neq 0$, and Claims 6-7, we have $F \neq 0$ over $\mathsf{R}[\boldsymbol{x}]$. We want to find $H' \subseteq \mathbb{F}^n$, s.t. $\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k))|_{\boldsymbol{x}=\boldsymbol{\alpha}} \neq 0$, for some $\boldsymbol{\alpha} \in H'$ (which will ensure the rank-preservation). Towards this, we will show (below) that F has $s^{O(\delta k)}$-size $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit over $\mathsf{R}[\boldsymbol{x}]$. Next, Theorem 27 provides the hitting set H' in time $s^{O(\delta^2 k \log s)}$.

▷ Claim 8 (Main size bound). $F \in \mathsf{R}[\boldsymbol{x}]$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size $(s3^\delta)^{O(k)}$.

The proof studies the two parts of Eqn. (7) –

1. The numerator $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))$ has $O(3^\delta 2^k k!ks)$-size $\Sigma\wedge\Sigma\Pi^{[\delta-1]}$-circuit (see Lemma 9), and
2. $1/\Psi(g_1\cdots g_k)$, for $g_i \in L(T_i)$ has $(s3^\delta)^{O(k)}$-size $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit; both over $\mathsf{R}[\boldsymbol{x}]$ (see Lemma 10).

▶ **Lemma 9** (Numerator size). $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \in \Sigma\wedge\Sigma\Pi^{[\delta-1]}$ *of size* $O(3^\delta\, 2^k k\, k!s) =: s_2$.

Proof sketch. One can show that $J_{\boldsymbol{x}_k}(g_1,\ldots,g_k) \in \Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$ of size $O(k!ks)$, where $g_i \in L(T_i)$ (Claim 25): this basically follows from the determinant expansion which has fanin $k!$ and the degree at the bottom is $\leq \delta - 1$ because of the derivative. Moreover, for a $g \in \Sigma\Pi^{[\delta-1]}$, we have $\Psi(g) \in \Sigma\Pi^{[\delta-1]}$ of size at most $3^\delta \cdot \text{size}(g)$, over $\mathsf{R}[\boldsymbol{x}]$ (Claim 26): this follows from the fact that $\boldsymbol{x}^{\boldsymbol{e}}$ (where $|\boldsymbol{e}|_0 \leq \delta$), after shift, can produce at most $\prod(e_i+1) \leq e^\delta$ many monomials (for large n). Combining these, one concludes $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \in \Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$, of size $O(3^\delta\, k!ks)$. We *convert* the Π-gate to $\wedge$ gate using waring identity (Lemma 15) which blowsup the size by a multiple of 2^{k-1}. Thus, $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \in \Sigma\wedge\Sigma\Pi^{[\delta-1]}$ of size $O(3^\delta\, 2^k k\, k!s)$. ◀

By power series expansion of expressions like $1/(1 - a\cdot z_1)$, one can conclude that $1/\Psi(g)$ has a small $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit, which would further imply the same for $1/\Psi(g_1\cdots g_k)$ (see below).

▶ **Lemma 10** (Denominator size). *Let $g_i \in L(T_i)$. Then, $1/\Psi(g_1 \cdots g_k)$ can be computed by a $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size $s_1 := (s3^\delta)^{O(k)}$, over $\mathsf{R}[\boldsymbol{x}]$.*

Proof. Let $g \in L(T_i)$ for some i. Assume, $\Psi(g) = A - z_1 \cdot B$, for some $A \in \mathbb{F}[z_2]$ and $B \in \mathsf{R}[\boldsymbol{x}]$ of degree δ, with $\text{size}(B) \le 3^\delta \cdot s$, from Claim 26. Note that, over $\mathsf{R}[\boldsymbol{x}]$,

$$\frac{1}{\Psi(g)} \;=\; \frac{1}{A(1-\frac{B}{A}\cdot z_1)} \;=\; \frac{1}{A}\cdot\sum_{i=0}^{D-1}\left(\frac{B}{A}\right)^i \cdot z_1^i\,. \tag{8}$$

As, $\text{size}(B^i)$ has a trivial $\wedge\Sigma\Pi^{[\delta]}$-circuit (over $\mathsf{R}[\boldsymbol{x}]$) of size $\le 3^\delta \cdot s + i$; summing over $i \in [D-1]$, the overall size is at most $D \cdot 3^\delta \cdot s + O(D^2)$. As $D < k \cdot d$, we conclude that $1/\Psi(g)$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$ of size $\mathsf{poly}(s \cdot k \cdot d3^\delta)$, over $\mathsf{R}[\boldsymbol{x}]$. Multiplying k-many such products directly gives an upper bound of $(s \cdot 3^\delta)^{O(k)}$, using Lemma 16 (basically, waring identity). ◀

Proof of Claim 8. Combining Lemmas 9-10, observe that $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))/\Psi(g_1\cdots g_k)$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size at most $(s_1 \cdot s_2)^2 = (s \cdot 3^\delta)^{O(k)}$, over $\mathsf{R}[\boldsymbol{x}]$, using Lemma 16. Summing up at most s^k many terms (by defn. of F), the size still remains $(s \cdot 3^\delta)^{O(k)}$. ◀

Degree bound. As, syntactic degree of T_i are bounded by d, and Ψ maintain $\deg_{\boldsymbol{x}} = \deg_{z_1}$, we must have $\deg_{z_1}(\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) = \deg_{\boldsymbol{x}}(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \le D-1$. Similarly, by assumption $\deg_{z_2}(\Psi(g)) \le D_1 := \mathsf{poly}(n^\delta)$, and thus $\deg_{z_2}(\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \le D_1 \cdot k$. Note that, Lemma 9 actually works over $\mathbb{F}[\boldsymbol{x},\boldsymbol{z}]$ and thus there is no additional degree-blow up (in $\boldsymbol{z}$). However, there is some degree blowup in Lemma 10, due to Eqn. (8).

Note that Eqn. (8) shows that over $\mathsf{R}[\boldsymbol{x}]$,

$$\frac{1}{\Psi(g)} = \left(\frac{1}{A^D}\right)\cdot\left(\sum_{i=0}^{D-1} A^{D-1-i} z_1^i \cdot B^i\right) =: \frac{p(\boldsymbol{x},\boldsymbol{z})}{q(z_2)},$$

where $q(z_2) = A^D$. We think of $p \in \mathbb{F}[\boldsymbol{x},\boldsymbol{z}]$ and $q \in \mathbb{F}[z_2]$. It follows that $\deg_{z_2}(q) \le D_1 \cdot D$. Also, $\deg_{z_1}(\Psi(g)) \le \delta$ implies $\deg_{z_1}(p) \le \deg_{z_1}((B\,z_1)^{D-1}) \le \delta \cdot (D-1)$. Since, $\deg_{z_2}(\Psi(g)) \le D_1$, by assumption, $\deg_{z_2}(p) \le \max_i \deg_{z_2}(A^{D-1-i} \cdot B^i) \le D_1 \cdot (D-1)$.

Finally, denote $1/\Psi(g_1\cdots g_k) =: P_{g_1,\ldots,g_k}/Q_{g_1,\ldots,g_k}$, over $\mathsf{R}[\boldsymbol{x}]$. This is just multiplying k-many (p/q)'s; implying a degree blowup by a multiple of k. In particular,

- $\deg_{z_1}(P_{(\cdot)}) \le \delta \cdot k \cdot (D-1)$,
- $\deg_{z_2}(P_{(\cdot)}) \le D_1 \cdot (D-1) \cdot k$, and
- $\deg_{z_2}(Q_{(\cdot)}) \le D_1 \cdot D \cdot k$.

Thus, in Eqn. (7), summing up s^k-many terms gives an expression (over $\mathsf{R}[\boldsymbol{x}]$):

$$F \;=\; \sum_{g_1\in L(T_1),\ldots,g_k\in L(T_k)} \Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))\cdot\left(\frac{P_{g_1,\ldots,g_k}}{Q_{g_1,\ldots,g_k}}\right) \;=:\; \frac{P(\boldsymbol{x},\boldsymbol{z})}{Q(z_2)}\,.$$

Verify that $Q \in \mathbb{F}[z_2]$ is of degree at most $s^k \cdot D_1 \cdot D \cdot k = s^{O(k)} \cdot \mathsf{poly}(n^\delta)$ (since $k, d < s$). A similar bound also holds for $\deg_{z_2}(P)$. The degree of z_1 also remains bounded by

$$\max_{g_i\in L(T_i), i\in[k]} \deg_{z_1}(P_{g_1,\ldots,g_k}) + \delta k \le \mathsf{poly}(s).$$

Using the degree bounds, we finally have $P \in \mathbb{F}[\boldsymbol{x},\boldsymbol{z}]$ as a $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit (over $\mathbb{F}(\boldsymbol{z})$) of size $n^{O(\delta)}\,(s3^\delta)^{O(k)} = 3^{O(\delta k)} s^{O(k+\delta)} =: s_3$.

We want to *construct* a set $H' \subseteq \mathbb{F}^n$ such that the action $P(H', \boldsymbol{z}) \neq 0$. Using [25] (Theorem 27), we conclude that it has $s^{O(\delta \log s_3)} = s^{O(\delta^2 k \log s)}$ size hitting set which is constructible in a similar time. Hence, the construction of Φ follows, making $\Phi(f)$ a $k+3$ variate polynomial. Finally, by the obvious degree bounds of $\boldsymbol{y}, \boldsymbol{z}, t$ from the definition of Φ, we get the blackbox PIT algorithm with time-complexity $s^{O(\delta^2 k \log s)}$; finishing Theorem 2b.

We could also give the final hitting set for the general problem.

Solution to Problem 5. We know that $C(T_1, \ldots, T_m) = 0 \iff E := \Phi(C(T_1, \ldots, T_m)) = 0$. Since, H' can be constructed in $s^{O(\delta^2\, k\, \log s)}$-time, it is trivial to find hitting set for $E|_{H'}$ (which is just a $k+3$-variate polynomial with the aformentioned degree bounds). The final hitting set for E can be constructed in $s'^{O(k)} \cdot s^{O(\delta^2\, k\, \log s)}$-time. ◀

▶ Remark.

1. As Jacobian Criterion (Fact 23) holds when the characteristic is $> d^{\,\mathrm{trdeg}}$, it is easy to conclude that our theorem holds for all fields of char $> d^k$.

2. The above proof gives an efficient reduction from blackbox PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits to $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits. In particular, a poly-time hitting set for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits would put PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ in P.

3. Also, DiDI-technique (of Theorem 1) directly gives a blackbox algorithm, but the complexity is *exponentially* worse (in terms of k in the exponent) for its recursive blowups.

4 Conclusion

This work introduces the powerful DiDI-technique and solves three open problems in PIT for depth-4 circuits, namely $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ (blackbox) and $\Sigma^{[k]}\Pi\Sigma\wedge$ (both whitebox and blackbox). Here are some immediate questions of interest which require rigorous investigation.

1. Can the exponent in Theorem 1 be improved to $O(k)$? Currently, it is exponential in k.
2. Can we improve Theorem 2b to $s^{O(\log \log s)}$ (like in Theorem 2a)?
3. Can we design a polynomial-time PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$?
4. Design a poly-time PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits (i.e. unbounded top-fanin)?
5. Can we solve PIT for $\Sigma^{[k]}\Pi\Sigma\wedge^{[2]}$ in *sub*exponential-time?
6. Can we design a subexponential-time PIT for rational functions of the form $\Sigma\,(1/\Sigma\wedge\Sigma)$ or $\Sigma\,(1/\Sigma\Pi)$ (for *un*bounded top-fanin)?

References

1 Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 92–105. Springer, 2005. doi:10.1007/11590156_6.

2 Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences*, 116(17):8107–8118, 2019. Preliminary version in Symposium on Theory of Computing, 2018 (STOC'18). doi:10.1073/pnas.1901272116.

3 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697, 2015. doi:10.1137/140975103.

4 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004. URL: `https://annals.math.princeton.edu/2004/160-2/p12`.

5 Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-D occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM Journal on Computing*, 45(4):1533–1562, 2016. Preliminary version in 44^{th} Symposium on Theory of Computing, 2018 (STOC'12). `doi:10.1137/130910725`.

6 Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Proceedings of the* 45^{th} *Annual ACM symposium on Theory of computing (STOC'13)*, pages 321–330, 2013. `doi:10.1145/2488608.2488649`.

7 Manindra Agrawal and V Vinay. Arithmetic Circuits: A Chasm at Depth Four. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 67–75. IEEE, 2008. URL: `https://ieeexplore.ieee.org/document/4690941`.

8 Matthew Anderson, Michael A Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *ACM Transactions on Computation Theory (TOCT)*, 10(1):1–30, 2018. Preliminary version in the IEEE 31^{st} Computational Complexity Conference (CCC'16). `doi:10.1145/3170709`.

9 Robert Andrews. Algebraic Hardness Versus Randomness in Low Characteristic. In *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *LIPIcs*, pages 37:1–37:32. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.CCC.2020.37`.

10 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998. `doi:10.1145/278298.278306`.

11 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM (JACM)*, 45(1):70–122, 1998. Preliminary version in 33^{rd} Annual Symposium on Foundations of Computer Science (FOCS'92). `doi:10.1145/273865.273901`.

12 Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Information and Computation*, 222:2–19, 2013. Preliminary version in 38^{th} International Colloquium on Automata, Languages and Programming (ICALP'11). URL: `https://www.sciencedirect.com/science/article/pii/S0890540112001435`.

13 Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the* 20^{th} *Annual ACM symposium on Theory of computing (STOC'88)*, pages 301–309, 1988. `doi:10.1145/62212.62241`.

14 Pranav Bisht and Nitin Saxena. Poly-time blackbox identity testing for sum of log-variate constant-width ROABPs. *Computational Complexity*, 2021. URL: `https://cse.iitk.ac.in/users/nitin/papers/constWidth-log-var.pdf`.

15 Enrico Carlini, Maria Virginia Catalisano, and Anthony V. Geramita. The solution to the Waring problem for monomials and the sum of coprime monomials. *Journal of Algebra*, 370:5–14, 2012. `doi:10.1016/j.jalgebra.2012.07.028`.

16 Prerona Chatterjee, Mrinal Kumar, C Ramya, Ramprasad Saptharishi, and Anamay Tengse. On the Existence of Algebraically Natural Proofs. In *IEEE* 61^{st} *Annual Symposium on Foundations of Computer Science (FOCS'20)*, 2020. URL: `https://eccc.weizmann.ac.il/report/2020/063/`.

17 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Hardness vs randomness for bounded depth arithmetic circuits. In 33^{rd} *Computational Complexity Conference (CCC'18)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. `doi:10.4230/LIPIcs.CCC.2018.13`.

18 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978. URL: `https://www.sciencedirect.com/science/article/abs/pii/0020019078900674`.

19 Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. In *Proceedings of the* 50^{th} *Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, pages 1152–1165, 2018. `doi:10.1145/3188745.3188760`.

20 Pranjal Dutta, Nitin Saxena, and Thomas Thierauf. A Largish Sum-Of-Squares Implies Circuit Hardness and Derandomization. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.23.

21 Zeev Dvir, Rafael Mendes De Oliveira, and Amir Shpilka. Testing equivalence of polynomials under shifts. In *International Colloquium on Automata, Languages, and Programming*, pages 417–428. Springer, 2014. doi:10.1007/978-3-662-43948-7_35.

22 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007. doi:10.1137/05063605X.

23 Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM Journal on Computing*, 39(4):1279–1293, 2010. Preliminary version in Proceedings of the 40^{th} Annual ACM symposium on Theory of computing (STOC'08). doi:10.1137/080735850.

24 Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. *SIAM Journal on Computing*, 62(3):109–115, 2019. Preliminary version in Proceedings of the 48^{th} Annual ACM symposium on Theory of Computing (STOC'16). URL: https://epubs.siam.org/doi/abs/10.1137/16M1097870?journalCode=smjcat.

25 Michael A Forbes. Deterministic divisibility testing via shifted partial derivatives. In *Proceedings of the 56^{th} Annual Symposium on Foundations of Computer Science (FOCS'15)*, pages 451–465. IEEE, 2015. URL: https://ieeexplore.ieee.org/document/7354409/.

26 Michael A Forbes, Sumanta Ghosh, and Nitin Saxena. Towards blackbox identity testing of log-variate circuits. In 45^{th} *International Colloquium on Automata, Languages, and Programming (ICALP'18)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.54.

27 Michael A Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46^{th} Annual ACM symposium on Theory of computing (STOC'14)*, pages 867–875, 2014. doi:10.1145/2591796.2591816.

28 Michael A Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In 54^{th} *Annual Symposium on Foundations of Computer Science (FOCS'13)*, pages 243–252, 2013. URL: https://ieeexplore.ieee.org/document/6686160/.

29 Michael A Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14:1–45, 2018. Preliminary version in Proceedings of the 49^{th} Annual ACM SIGACT Symposium on Theory of Computing (STOC'19). URL: https://theoryofcomputing.org/articles/v014a018/.

30 Abhibhav Garg and Nitin Saxena. Special-case algorithms for blackbox radical membership, Nullstellensatz and transcendence degree. In *Proceedings of the 45^{th} International Symposium on Symbolic and Algebraic Computation*, pages 186–193, 2020.

31 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In 57^{th} *Annual Symposium on Foundations of Computer Science (FOCS'16)*, pages 109–117. IEEE, 2016. URL: https://ieeexplore.ieee.org/document/7782923.

32 Joshua A Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015. Preliminary version in the IEEE $29th$ Computational Complexity Conference (CCC'14). doi:10.1007/s00037-015-0103-x.

33 Zeyu Guo. Variety Evasive Subspace Families. In *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: https://zeyuguo.bitbucket.io/papers/chow.pdf.

34 Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from Algebraic Hardness: Treading the Borders. In 60^{th} *IEEE Annual Symposium on Foundations of Computer Science (FOCS'19)*, pages 147–157. IEEE Computer Society, 2019. URL: https://ieeexplore.ieee.org/document/8948610/.

35 Ankit Gupta. Algebraic Geometric Techniques for Depth-4 PIT & Sylvester-Gallai Conjectures for Varieties. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 21, page 130, 2014. URL: https://eccc.weizmann.ac.il/report/2014/130/.

36 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. *SIAM Journal on Computing*, 45(3):1064–1079, 2016. 54^{th} Annual Symposium on Foundations of Computer Science (FOCS'13). doi:10.1137/140957123.

37 Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs. *Theory of Computing*, 13(2):1–21, 2017. Preliminary version in the 31^{st} Computational Complexity Conference (CCC'16). doi:10.4086/toc.2017.v013a002.

38 Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Computational Complexity*, 26(4):835–880, 2017. Preliminary version in the IEEE 30^{th} Computational Complexity Conference (CCC'15). doi:10.1007/s00037-016-0141-z.

39 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute. In *Proceedings of the 12^{th} annual ACM symposium on Theory of computing (STOC'80)*, pages 262–272, 1980. doi:10.1145/800141.804674.

40 Maurice Jansen, Youming Qiao, and Jayalal Sarma. Deterministic Black-Box Identity Testing π-Ordered Algebraic Branching Programs. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010*, volume 8 of *LIPIcs*, pages 296–307. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.296.

41 A Grochow Joshua, D Mulmuley Ketan, and Qiao Youming. Boundaries of VP and VNP. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 34:1–34:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.34.

42 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the Proceedings of the 35^{th} Annual ACM symposium on Theory of computing (STOC'03). doi:10.1007/s00037-004-0182-6.

43 Zohar S Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in. *SIAM Journal on Computing*, 42(6):2114–2131, 2013. Preliminary version in the Proceedings of the 42^{nd} ACM symposium on Theory of computing (STOC'10). doi:10.1137/110824516?af=R.

44 Zohar S Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In 24^{th} *Annual IEEE Conference on Computational Complexity (CCC'09)*, pages 274–285. IEEE, 2009. URL: https://ieeexplore.ieee.org/document/5231339.

45 Zohar S Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333, 2011. Preliminary version in the 23^{rd} Annual IEEE Conference on Computational Complexity (CCC'08). doi:10.1007/s00493-011-2537-3.

46 Neeraj Kayal, Pascal Koiran, Timothée Pecatte, and Chandan Saha. Lower bounds for sums of powers of low degree univariates. In *International Colloquium on Automata, Languages, and Programming (ICALP'15)*, pages 810–821. Springer, 2015. doi:10.1007/978-3-662-47672-7_66.

47 Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007. Preliminary version in the 21^{st} Computational Complexity Conference (CCC'06). doi:10.1007/s00037-007-0226-9.

48 Adam Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(1):185–206, 2006. Preliminary version in the 16^{th} Annual Conference on Learning Theory (COLT'03). URL: https://theoryofcomputing.org/articles/v002a010/.

49 Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33^{rd} Annual ACM symposium on Theory of computing (STOC'01)*, pages 216–223, 2001. doi:10.1145/380752.380801.

50 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012. URL: https://www.sciencedirect.com/science/article/pii/S0304397512003131.

51 Pascal Koiran, Natacha Portier, and Sébastien Tavenas. A Wronskian approach to the real τ-conjecture. *Journal of Symbolic Computation*, 68:195–214, 2015. URL: https://www.sciencedirect.com/science/article/pii/S0747717114001047.

52 Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *IEEE 29^{th} Conference on Computational Complexity (CCC'14)*, pages 169–180. IEEE, 2014. URL: https://ieeexplore.ieee.org/document/6875486.

53 Mrinal Kumar, C Ramya, Ramprasad Saptharishi, and Anamay Tengse. If VNP is hard, then so are equations for it. *Preprint avilable at arXiv:2012.07056*, 2020.

54 Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bulletin of EATCS*, 3(129), 2019. URL: https://mrinalkr.bitbucket.io/papers/hardness-randomness-survey.pdf.

55 Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits. In *Proceedings of the 30^{th} Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 639–646, 2019. doi:10.5555/3310435.3310475.

56 Mrinal Kumar and Shubhangi Saraf. Sums of Products of Polynomials in Few Variables: Lower Bounds and Polynomial Identity Testing. In *31^{st} Conference on Computational Complexity, CCC 2016*, volume 50 of *LIPIcs*, pages 35:1–35:29. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.35.

57 Mrinal Kumar and Shubhangi Saraf. Arithmetic Circuits with Locally Low Algebraic Rank. *Theory Comput.*, 13(1):1–33, 2017. Preliminary version in the 31^{st} Conference on Computational Complexity (CCC'16). URL: http://www.theoryofcomputing.org/articles/v013a006/.

58 Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. Non-commutative computations: lower bounds and polynomial identity testing. *Chic. J. Theor. Comput. Sci.*, 2:1–19, 2019. URL: http://cjtcs.cs.uchicago.edu/articles/2019/2/cj19-02.pdf.

59 László Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of Computation Theory (FCT'79)*, volume 79, pages 565–574, 1979. URL: http://www.math.uwaterloo.ca/~harvey/W11/1979-Lovasz-OnDeterminantsMatchingsAndRandomAlgs.pdf.

60 Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM (JACM)*, 39(4):859–868, 1992. doi:10.1145/146585.146605.

61 Partha Mukhopadhyay. Depth-4 identity testing and Noether's normalization lemma. In *International Computer Science Symposium in Russia (CSR'16)*, pages 309–323. Springer, 2016. doi:10.1007/978-3-319-34171-2_22.

62 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Comb.*, 7(1):105–113, 1987. Preliminary version in the Proceedings of the 19^{th} Annual ACM symposium on Theory of Computing (STOC'87). doi:10.1007/BF02579206.

63 Ketan D Mulmuley. Geometric complexity theory V: Equivalence between blackbox derandomization of polynomial identity testing and derandomization of Noether's normalization lemma. In *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS'12)*, pages 629–638. IEEE, 2012. arXiv:1209.5993.

64 Ketan D Mulmuley. The GCT program toward the P vs. NP problem. *Communications of the ACM*, 55(6):98–107, 2012. doi:10.1145/2184319.2184341.

65 Ivan Niven. Formal power series. *The American Mathematical Monthly*, 76(8):871–889, 1969. URL: http://www.jstor.org/stable/2317940.

66 Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

67 Anurag Pandey, Nitin Saxena, and Amit Sinhababu. Algebraic independence over positive characteristic: New criterion and applications to locally low-algebraic-rank circuits. *Computational Complexity*, 27(4):617–670, 2018. Preliminary version in the 41^{st} International Symposium on Mathematical Foundations of Computer Science (MFCS'16). doi:10.1007/s00037-018-0167-5.

68 Shir Peleg and Amir Shpilka. A generalized Sylvester-Gallai type theorem for quadratic polynomials. In 35^{th} *Computational Complexity Conference (CCC'20)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.8.

69 Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\sum^{[3]}\prod\sum\prod^{[2]}$ circuits via Edelstein-Kelly type theorem for quadratic polynomials. In 53^{rd} *Annual ACM symposium on Theory of computing (STOC'21)*, 2021. arXiv:2006.08263.

70 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. Preliminary version in the 19^{th} IEEE Annual Conference on Computational Complexity (CCC'04). doi:10.1007/s00037-005-0188-8.

71 Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. A case of depth-3 identity testing, sparse factorization and duality. *Computational Complexity*, 22(1):39–69, 2013. doi:10.1007/s00037-012-0054-4.

72 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2019. URL: https://github.com/dasarpmar/lowerbounds-survey/releases.

73 Ramprasad Saptharishi. Private communication, 2019.

74 Shubhangi Saraf and Ilya Volkovich. Black-box identity testing of depth-4 multilinear circuits. *Combinatorica*, 38(5):1205–1238, 2018. Preliminary version in the Proceedings of the 43^{rd} Annual ACM symposium on Theory of computing (STOC'11). doi:10.1007/s00493-016-3460-4.

75 Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *International Colloquium on Automata, Languages, and Programming (ICALP'08)*, pages 60–71. Springer, 2008. doi:10.1007/978-3-540-70575-8_6.

76 Nitin Saxena. Progress on Polynomial Identity Testing. *Bulletin of the EATCS*, 99:49–79, 2009. URL: https://www.cse.iitk.ac.in/users/nitin/papers/pit-survey09.pdf.

77 Nitin Saxena. Progress on polynomial identity testing-II. In *Perspectives in Computational Complexity*, pages 131–146. Springer, 2014. doi:10.1007/978-3-319-05446-9_7.

78 Nitin Saxena and Comandur Seshadhri. An almost optimal rank bound for depth-3 identities. *SIAM journal on computing*, 40(1):200–224, 2011. Preliminary version in the 24^{th} IEEE Conference on Computational Complexity (CCC'09). doi:10.1137/090770679.

79 Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM Journal on Computing*, 41(5):1285–1298, 2012. Preliminary version in the 43^{rd} Annual ACM symposium on Theory of computing (STOC'11). doi:10.1137/10848232.

80 Nitin Saxena and Comandur Seshadhri. From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *Journal of the ACM (JACM)*, 60(5):1–33, 2013. Preliminary version in the 51^{st} Annual IEEE Symposium on Foundations of Computer Science (FOCS'10). doi:10.1145/2528403.

81 Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

82 Adi Shamir. IP= PSPACE. *Journal of the ACM (JACM)*, 39(4):869–877, 1992. doi:10.1145/146585.146609.

83 Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM Journal on Computing*, 38(6):2130–2161, 2009. Preliminary version in the Proceedings of the 39^{th} Annual ACM symposium on Theory of Computing (STOC 2007). doi:10.1137/070694879.

84 Amir Shpilka. Sylvester-Gallai type theorems for quadratic polynomials. In *Proceedings of the 51^{st} Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*, pages 1203–1214, 2019. doi:10.1145/3313276.3316341.

85 Amir Shpilka and Amir Yehudayoff. *Arithmetic circuits: A survey of recent results and open questions*. Now Publishers Inc, 2010. URL: https://www.cs.tau.ac.il/~shpilka/publications/SY10.pdf.

86 Amit Kumar Sinhababu. *Power series in complexity: Algebraic Dependence, Factor Conjecture and Hitting Set for Closure of VP*. PhD thesis, PhD thesis, Indian Institute of Technology Kanpur, 2019. URL: https://www.cse.iitk.ac.in/users/nitin/theses/sinhababu-2019.pdf.

87 Leslie G Valiant. Completeness classes in algebra. In *Proceedings of the 11^{th} Annual ACM symposium on Theory of computing (STOC'79)*, pages 249–261, 1979. doi:10.1145/800135.804419.

88 Wolmer Vasconcelos. *Computational methods in commutative algebra and algebraic geometry*, volume 2. Springer Science & Business Media, 2004. URL: https://www.springer.com/gp/book/9783540213116.

89 Richard Zippel. Probabilistic Algorithms for Sparse Polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, 1979. doi:10.1007/3-540-09519-5_73.

A Basic tools from algebraic complexity

There have been a lot of work on sparse-PIT, for details see [13, 49] and references therein. Eventually, there is a poly-time hitting set, for a proof see [76, Thm. 2.1]

▶ **Theorem 11** ([49]). *Let $p(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ with individual degree at most d and sparsity at most m. Then, there exists $1 \le r \le (mn \log d)^2$, such that $p(y, y^d, \ldots, y^{d^{n-1}}) \neq 0, \bmod y^r - 1$.*

An **ABP** is a layered directed acyclic graph with $q+1$ many layers of vertices $\{V_0, \ldots, V_q\}$ and a source a and a sink b such that all the edges in the graph only go from a to V_0, V_{i-1} to V_i for any $i \in [q]$, and V_q to b. The edges have *uni*variate polynomials as their weights. The ABP is said to compute the polynomial

$$f(\boldsymbol{x}) \;=\; \sum_{p \in \text{paths}(a,b)} \prod_{e \in p} W(e)\ ,$$

where $W(e)$ is the weight of the edge e. The ABP has width-w if $|V_i| \le w$, $\forall i \in \{0, \ldots, q\}$. Formally, it computes polynomials of the form $A^T(\prod_{i\in[q]} D_i)B$, where $A, B \in \mathbb{F}^{w\times 1}[\boldsymbol{x}]$, and $D_i \in \mathbb{F}^{w\times w}[\boldsymbol{x}]$, where entries are univariate polynomials.

▶ **Definition 12** (Read-once oblivious ABP (ROABP)). *An ABP is called* a read-once oblivious ABP (ROABP) *if the edge weights are univariate polynomials in* distinct *variables across layers. Formally, there is a permutation π on the set $[q]$ such that the entries in the i-th matrix D_i are univariate polynomials over the variable $x_{\pi(i)}$, i.e., they come from the polynomial ring $\mathbb{F}[x_{\pi(i)}]$.*

A polynomial $f(x)$ is said to be computed by width-w ROABPs in *any order*, if for every permutation σ of the variables, there exists a width-w ROABP in the variable order s that computes the polynomial $f(\boldsymbol{x})$. There have been quite a few results on blackbox PIT for ROABPs [28, 27, 37] and the current best known algorithm works in quasipolynomial time.

▶ **Theorem 13** ([37]). *For n-variate, individual-degree-d polynomials computed by width-w ROABPs in any order, a hitting set of size $(ndw)^{O(\log\log w)}$ can be constructed.*

B Details for Section 3.1

Here is an important lemma which shows that coefficient of y^e of a polynomial $f(\boldsymbol{x}, y) \in \mathbb{F}[\boldsymbol{x}, y]$, computed by a $\Sigma\wedge\Sigma\wedge$ circuit, can be computed by a small $\Sigma\wedge\Sigma\wedge$ circuit.

▶ **Lemma 14** (Coefficient extraction). *Let $f(\boldsymbol{x}, y) \in \mathbb{F}[y][\boldsymbol{x}]$ be computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s and degree d. Then, $\mathrm{coef}_{y^e}(f) \in \mathbb{F}[\boldsymbol{x}]$ can be computed by a small $\Sigma\wedge\Sigma\wedge$ circuit of size $O(sd)$, over $\mathbb{F}[\boldsymbol{x}]$.*

Proof sketch. Let, $f = \sum_i \alpha_i \cdot g_i^{e_i}$. Of course, $e_i \le s$ and $\deg_y(f) \le d$. Thus, write $f = \sum_{i=0}^{d} f_i \cdot y^i$, where $f_i \in \mathbb{F}[\boldsymbol{x}]$. We can interpolate on $d+1$-many distinct points $y \in \mathbb{F}$ and conclude that f_i has a $\Sigma\wedge\Sigma\wedge$ circuit of size at most $O(sd)$. ◀

The next identity gives us a way to write a product of a few powers as a sum of powers, using simple interpolation. For a more algebraic proof, see [15, Proposition 4.3].

▶ **Lemma 15** (Waring Identity for a monomial). *Let $M = x_1^{b_1} \cdots x_k^{b_k}$, where $1 \le b_1 \le \ldots \le b_k$, and* roots of unity *$\mathcal{Z}(i) := \{z \in \mathbb{C} : z^{b_i+1} = 1\}$. Then,*

$$M = \sum_{\varepsilon(i) \in \mathcal{Z}(i) : i=2,\cdots,k} \gamma_{\varepsilon(2),\ldots,\varepsilon(k)} \cdot \left(x_1 + \varepsilon(2)x_2 + \ldots + \varepsilon(k)x_k\right)^d ,$$

where $d := \deg(M) = b_1 + \ldots + b_k$, and $\gamma_{\varepsilon(2),\ldots,\varepsilon(k)}$ are scalars ($\mathrm{rk}(M) := \prod_{i=2}^{k}(b_i+1)$ many).

▶ Remark. We actually need not work with $\mathbb{F} = \mathbb{C}$. We can go to a small extension (at most d^k), for a monomial of degree d, to make sure that $\varepsilon(i)$ exists.

The next lemma shows that $\Sigma\wedge\Sigma\wedge$ is *closed* under multiplication.

▶ **Lemma 16.** *Let $f_i(\boldsymbol{x}, y) \in \mathbb{F}[y][\boldsymbol{x}]$, of syntactic degree $\le d_i$, be computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s_i, for $i \in [k]$ (wrt $\boldsymbol{x}$). Then, $f_1 \cdots f_k$ has $\Sigma\wedge\Sigma\wedge$ circuit of size $O((d_2+1)\cdots(d_k+1) \cdot s_1 \cdots s_k)$.*

Proof. Let $f_i = \sum_j f_{ij}^{e_{ij}}$; by assumption $e_{ij} \le d_i$ (by assumption). Using Lemma 15, $f_{1j_1}^{e_{1j_1}} \cdots f_{kj_k}^{e_{kj_k}}$ has size at most $(d_2+1)\cdots(d_k+1) \cdot \left(\sum_{i\in[k]} \mathrm{size}(f_{ij_i})\right)$, for indices $j_1, \ldots, j_k$. Summing up for all $s_1 \cdots s_k$ many products (atmost) gives the upper bound. ◀

The next lemma shows that $\Sigma\wedge\Sigma\wedge$ is *closed* under differentiation.

▶ **Lemma 17** (Differentiation). *Let $f(\boldsymbol{x}, y) \in \mathbb{F}[y][\boldsymbol{x}]$ be computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s and degree d. Then, $\partial_y(f)$ can be computed by a small $\Sigma\wedge\Sigma\wedge$ circuit of size $O(sd^2)$, over $\mathbb{F}[y][\boldsymbol{x}]$.*

Proof sketch. Lemma 14 shows that each f_e has $O(sd)$ size circuit where $f = \sum_e f_e \, y^e$. Doing this for each $e \in [0, d]$ gives a blowup of $O(sd^2)$. ◀

The next lemma shows that non-negative valuation corresponds to a power-series.

▶ **Lemma 18** (Valuation). *Consider a polynomial $f \in \mathbb{F}(\boldsymbol{x}, y)$ such that $\mathrm{val}_y(f) \ge 0$. Then, $f \in \mathbb{F}(\boldsymbol{x})[[y]] \bigcap \mathbb{F}(\boldsymbol{x}, y)$.*

Proof sketch. Let $f = g/h$, where $g, h \in \mathbb{F}[\boldsymbol{x}, y]$. Now, $\mathrm{val}_y(f) \geq 0$, implies $\mathrm{val}_y(g) \geq \mathrm{val}_y(h)$. Let $\mathrm{val}_y(g) = d_1$ and $\mathrm{val}_y(h) = d_2$, where $d_1 \geq d_2 \geq 0$. Write $g = y^{d_1} \cdot \tilde{g}$ and $h = y^{d_2} \cdot \tilde{h}$. Write, $\tilde{h} = h_0 + h_1\, y + h_2\, y^2 + \ldots + h_d\, y^d$, for some d. Note that $h_0 \neq 0$. Thus,

$$\begin{aligned} f &= y^{d_1-d_2} \cdot \tilde{g}/(h_0 + h_1\, y + \ldots + h_d\, y^d) \\ &= y^{d_1-d_2} \cdot (\tilde{g}/h_0) \cdot (1 + (h_1/h_0)\, y + \ldots + (h_d/h_0)\, y^d)^{-1} \ \in \mathbb{F}(\boldsymbol{x})[[y]] \,. \end{aligned}$$

The last conclusion follows by the inverse identity in the power-series ring. ◀

Using duality trick [75] and PIT results from [70, 37], one can design efficient PIT algorithm for $\Sigma\wedge\Sigma\wedge$ circuits:

▶ **Lemma 19** (PIT for $\Sigma\wedge\Sigma\wedge$-circuits). *Let $P \in \Sigma\wedge\Sigma\wedge$ of size s. Then, there exists a* $\mathsf{poly}(s)$ *(respec. $s^{O(\log\log s)}$) time whitebox (respec. blackbox) PIT for the same.*

Proof sketch. We show that any $g(\boldsymbol{x})^e = (g_1(x_1) + \ldots + g_n(x_n))^e$, where $\deg(g_i) \leq s$ can be written as $\sum_j h_{j1}(x_1) \cdots h_{jn}(x_n)$, for some $h_{j\ell} \in \mathbb{F}[x_\ell]$ of degree at most es. Define, $G := (y + g_1) \cdots (y + g_n) - y^n$. In its e-th power, notice that the leading-coefficient is $\mathrm{coef}_{y^{e(n-1)}}(G^e) = g^e$. So, interpolate on $e(n-1)+1$ many points ($y = \beta_i \in \mathbb{F}$) to get

$$\mathrm{coef}_{y^{e(n-1)}}(G^e) \ = \ \sum_{i=1}^{e(n-1)+1} \alpha_i\, G^e(\beta_i) \,.$$

Now, expand $G^e(\beta_i) = ((\beta_i{+}g_1) \cdots (\beta_i{+}g_n) - \beta_i^n)^e$, by binomial expansion (without expanding the inner n-fold product). The top-fanin can be atmost $s \cdot (e+1) \cdot (e(n-1)+1) = O(se^2n)$. The individual degrees of the intermediate univariates can be at most es. Thus, it can be computed by an ROABP (of *any order*) of size at most $O(s^2e^3n)$.

Now, if $f = \sum_{j\in[s]} f_j^{e_j}$ is computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s, then clearly, f can also be computed by an ROABP (of any order) of size at most $O(s^6)$. So, the whitebox PIT follows from [70], while the blackbox PIT follows from Theorem 13. ◀

For the time-complexity bound, we need optimization of the following function:

▶ **Lemma 20.** *Let $k \in \mathbb{N}$, and $h(x) := x(k-x)7^x$. Then, $\max_{i\in[k-1]} h(i) = h(k-1)$.*

Proof sketch. Differentiate to get $h'(x) = (k-x)7^x - x7^x + x(k-x)(\log 7)7^x = 7^x \cdot [x^2(-\log 7) + x(k\log 7 - 2) + k]$. It vanishes at $x = \left(\frac{k}{2} - \frac{1}{\log 7}\right) + \sqrt{\left(\frac{k}{2} - \frac{1}{\log 7}\right)^2 - \frac{k}{\log 7}}$. Thus, h is maximized at the integer $x = k-1$. ◀

C Details for Section 3.2

▶ **Definition 21** (Faithful hom). *$\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{y}]$ is faithful for $\boldsymbol{T}$ if* $\mathrm{trdeg}_{\mathbb{F}}(\boldsymbol{T}) = \mathrm{trdeg}_{\mathbb{F}}(\Phi(\boldsymbol{T}))$.

The following fact about faithful maps is from [5, Thm. 2.4].

▶ **Fact 22** (Faithful is useful). *For any $C \in \mathbb{F}[y_1, \ldots, y_m]$, $C(\boldsymbol{T}) = 0 \iff C(\Phi(\boldsymbol{T})) = 0$.*

Here is an important criterion about the jacobian matrix which basically shows that it *preserves* algabraic independence.

▶ **Fact 23** (Jacobian criterion). *Let $\mathbf{f} \subset \mathbb{F}[\boldsymbol{x}]$ be a finite set of polynomials of degree at most d, and $\mathrm{trdeg}_{\mathbb{F}}(\mathbf{f}) \leq r$. If $char(\mathbb{F}) = 0$, or $char(\mathbb{F}) > d^r$, then $\mathrm{trdeg}_{\mathbb{F}}(\mathbf{f}) = \mathrm{rk}_{\mathbb{F}(\boldsymbol{x})}\mathcal{J}_{\boldsymbol{x}}(\mathbf{f})$.*

The following lemma (& the proof) is similar to [5, Lem. 2.7]. It is a recipe to "drastically" reduce variables, if trdeg is small.

▶ **Lemma 24** (Recipe for faithful maps). *Let $\boldsymbol{T} \in \mathbb{F}[\boldsymbol{x}]$ be be a finite set of polynomials of degree at most d and $\mathrm{trdeg}_{\mathbb{F}}(\boldsymbol{T}) \le r$, and char(F)=0 or $> d^r$. Let $\Psi' : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_1, z_2]$ such that $\mathrm{rk}_{\mathbb{F}(\boldsymbol{x})} \mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}) = \mathrm{rk}_{\mathbb{F}(\boldsymbol{z})} \Psi'(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$.*

Then, the map $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{z}, t, \boldsymbol{y}]$, such that $x_i \mapsto (\sum_j y_j t^{ij}) + \Psi'(x_i)$, is a faithful homomorphism for $\boldsymbol{T}$.

C.1 Technical Details for Theorem 2b

▷ Claim 25. Let $g_i \in L(T_i)$, where $T_i \in \Pi\Sigma\Pi^{[\delta]}$ of size atmost s, then $J_{\boldsymbol{x}_k}(g_1, \ldots, g_k) \in \Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$ of size $O(k!\,ks)$.

Proof sketch. Each entry of the matrix has degree at most $\delta - 1$. Trivial expansion gives $k!$ top-fanin where each product (of fanin k) has size $\sum_i$ size(g_i). As, size$(T_i) \le s$, trivially each size$(g_i) \le s$. Therefore, the total size is $k! \cdot \sum_i$ size$(g_i) = O(k!\,ks)$. ◁

▷ Claim 26. Let $g \in \Sigma\Pi^{\delta}$, then $\Psi(g) \in \Sigma\Pi^{\delta}$ of size $3^{\delta} \cdot \mathsf{size}(g)$ (for $n \gg \delta$).

Proof sketch. Each monomial $\boldsymbol{x}^{\boldsymbol{e}}$ of degree δ, can produce $\prod_i (e_i + 1) \le ((\sum_i e_i + n)/n)^n \le (\delta/n + 1)^n$-many monomials, by AM-GM inequality as $\sum_i e_i \le \delta$. As $\delta/n \to 0$, we have $(1 + \delta/n)^n \to e^{\delta}$. As $e < 3$, the upper bound follows. ◁

[25, Prop. 4.18] gave the first nontrivial PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits:

▶ **Theorem 27** ([25]). *There is a $\mathsf{poly}(n, d, \delta \log s)$-explicit hitting set of size $(nd)^{O(\delta \log s)}$ for the class of n-variate, degree-($\le d$) polynomials $f(\boldsymbol{x})$, computed by $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size s.*

D Proof sketch of Theorem 2a: Similar to Section 3.2

Similar to Theorem 2b, we generalize this theorem and prove for a much bigger class of polynomials.

▶ **Problem 28.** *Let $\{T_i \,|\, i \in [m]\}$ be $\Pi\Sigma\wedge$ circuits of (syntactic) degree at most d and size s. Let the transcendence degree of T_i's, $\mathrm{trdeg}_{\mathbb{F}}(T_1, \ldots, T_m) =: k \ll s$. Further, $C(x_1, \ldots, x_m)$ be a circuit of size + degree $< s'$. Design a blackbox-PIT algorithm for $C(T_1, \ldots, T_m)$.*

It is trivial to see that $\Sigma^{[k]}\Pi\Sigma\wedge$ is a very special case of the above settings. We will use the same idea (& notation) as in Theorem 2b, using the Jacobian technique. The main idea is to come up with Φ map, and correspondingly the hitting set H'. If $g \in L(T_i)$, then size$(g) \le O(dn)$. We also note that D_1, which is an upper bound on $\deg_{z_2} \Psi(g)$ is $\mathsf{poly}(n, d)$ (Lemma 11). The D (and hence $R[\boldsymbol{x}]$) remains as before. Claims 6-7 hold similarly. We will construct the hitting set H' by showing that F has a small $\Sigma\wedge\Sigma\wedge$ circuit over $R[\boldsymbol{x}]$.

Note that, Claim 25 remains the same for $\Sigma\wedge\Sigma\wedge$ (implying the same size blowup). However, Claim 26, the size blowup is $O(d\,\mathsf{size}(g))$, because each monomial x^e can only produce $d + 1$ many monomials. Therefore, similar to Lemma 10, one can show that $\Psi(J_{\boldsymbol{x}_k}(g_1, \ldots, g_k)) \in \Sigma\wedge\Sigma\wedge$, of size $O(2^k k! kds)$. Similarly, the size in Lemma 9 can be replaced by $s^{O(k)}$. Therefore, we get (similar to Claim 8):

▷ Claim 29. $F \in R[\boldsymbol{x}]$ has $\Sigma\wedge\Sigma\wedge$-circuit of size $s^{O(k)}$.

Next, the degree bound also remains the same (except the parameter D_1 which is now $\mathsf{poly}(nd)$). Following the same footsteps, it is not hard to see that the degree bound of z_2 on P and Q, where $F = P(\boldsymbol{x}, \boldsymbol{z})/Q(z_2)$, is $s^{O(k)}\mathsf{poly}(nd)$, while degree bound on z_1 remains $\mathsf{poly}(ksd)$. Therefore, $P \in \mathbb{F}[\boldsymbol{x}, \boldsymbol{z}]$ has $\Sigma\wedge\Sigma\wedge$ -circuit of size $s^{O(k)}$.

We want to *construct* a set $H' \subseteq \mathbb{F}^n$ such that the action $P(H', \boldsymbol{z}) \neq 0$. By Theorem 19, we conclude that it has $s^{O(k \log\log s)}$ size hitting set which is constructible in a similar time. Hence, the construction of map Φ and the theorem follows (from $\boldsymbol{z}$-degree bound).

Solution to Problem 28. We know that $C(T_1, \ldots, T_m) = 0 \iff E := \Phi(C(T_1, \ldots, T_m)) = 0$. Since, H' can be constructed in $s^{O(k \log\log s)}$ time, it is trivial to find hitting set for $E|_{H'}$ (which is just a $k+3$-variate polynomial with the aforementioned degree bounds). The final hitting set for E can be constructed in $s'^{O(k)} \cdot s^{O(k \log\log s)}$ time. ◀

E Algorithm for Theorem 1

The whitebox PIT for Theorem 1, that is discussed in Section 3.1, appears (below) as Algorithm 1.

Algorithm 1 Whitebox PIT Algorithm for $\Sigma^{[k]}\Pi\Sigma\wedge$-circuits.

```
Input  : $f = T_1 + \ldots + T_k \in \Sigma^{[k]}\Pi\Sigma\wedge$, a whitebox circuit of size $s$ over $\mathbb{F}[\boldsymbol{x}]$.
Output : 0, if $f \equiv 0$, and 1, if it is non-zero.
Let $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$, be a sparse-PIT map, using [49] (Theorem 11). Apply it on $f$ and check
  whether $\Psi(f) \overset{?}{=} 0$. If non-zero, output 1 otherwise, apply $\Phi : x_i \mapsto z_1 \cdot x_i + \Psi(x_i)$ on $f$. Check
  $\sum_{i\in[k-1]} \partial_{z_1}(\Phi(T_i)/\Phi(T_k)) \overset{?}{=} 0 \bmod z_1^{d_1}$ ($d_1 := s$) as follows:
Consider each $T_{i,1} := \partial_{z_1}(\Phi(T_i)/\Phi(T_k))$ over $R_1(\boldsymbol{x})$, where $R_1 := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_1}\rangle$. Use dlog
  computation (Claim 4), to write each $T_{i,1}$ in a "bloated" form as
  $(\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$.
for $j \leftarrow 1$ to $k-1$ do
    Reduce the top-fanin at each step using "Divide & Derive" technique. Assume that at $j$-th
      step, we have to check the identity:
      $\sum_{i\in[k-j]} T_{i,j} \overset{?}{=} 0$ over $R_j(\boldsymbol{x})$, where $R_j := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j}\rangle$ , each $T_{i,j}$ has a
      $(\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ representation and therein each $\Pi\Sigma\wedge|_{z_1=0} \in \mathbb{F}(z_2) \setminus \{0\}$.
    1. Compute $v_{k-j,j} := \min_i \mathrm{val}_{z_1}(T_{i,j})$; by reordering it is for $i = k-j$. To compute $v_{k-j,j}$,
       use coefficient extraction (Lemma 14) and $\Sigma\wedge\Sigma\wedge$ -circuit PIT (Lemma 19).
    2. "Divide" by $T_{k-j,j}$ and check whether $\left(\sum_{i\in[k-j-1]} (T_{i,j}/T_{k-j,j}) + 1\right)\Big|_{z_1=0} \overset{?}{=} 0$. Note:
       this expression is in $(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$. Use – (1) $\Pi\Sigma\wedge|_{z_1=0} \in \mathbb{F}(z_2)$, and (2) closure of
       $\Sigma\wedge\Sigma\wedge$ under multiplication. Finally, do PIT on this by Lemma 19.
    3. If it is non-zero, output 1, otherwise "Derive" wrt $z_1$ and "Induct" on
       $\left(\sum_{i\in[k-j-1]} \partial_{z_1}(T_{i,j}/T_{k-j,j})\right) \overset{?}{=} 0$, over $R_{j+1}(\boldsymbol{x})$ where $R_{j+1} := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j - v_{k-j,j} - 1}\rangle$.
    4. Again using dlog (Claim 4), show that $T_{i,j+1} := \partial_{z_1}(T_{i,j}/T_{k-j,j})$ has small
       $(\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$-circuit over $R_{j+1}(\boldsymbol{x})$. So call the algorithm on
       $\sum_{i\in[k-j-1]} T_{i,j+1} \overset{?}{=} 0$.
    $j \leftarrow j + 1$.
end
At the end, $j = k-1$. Do PIT (Lemma 19) on the single $(\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ circuit,
  over $R_{k-1}(\boldsymbol{x})$. If it is zero, output 0 otherwise output 1.
```

Words of caution: Throughout the algorithm there are intermediate expressions to be stored compactly. Think of them as "special" circuits in $\boldsymbol{x}$, but over the *function-field* $\mathbb{F}(\boldsymbol{z})$. Keep track of their degrees wrt z_1, z_2; and that of the sizes of their fractions represented in "bloated" circuit form.

Robustly Self-Ordered Graphs: Constructions and Applications to Property Testing

Oded Goldreich
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel

Avi Wigderson
School of Mathematics, Institute for Advanced Study, Princeton, USA

Abstract

A graph G is called *self-ordered* (a.k.a asymmetric) if the identity permutation is its only automorphism. Equivalently, there is a unique isomorphism from G to any graph that is isomorphic to G. We say that $G = (V, E)$ is *robustly self-ordered* if the size of the symmetric difference between E and the edge-set of the graph obtained by permuting V using any permutation $\pi : V \to V$ is proportional to the number of non-fixed-points of π. In this work, we initiate the study of the structure, construction and utility of robustly self-ordered graphs.

We show that robustly self-ordered bounded-degree graphs exist (in abundance), and that they can be constructed efficiently, in a strong sense. Specifically, given the index of a vertex in such a graph, it is possible to find all its neighbors in polynomial-time (i.e., in time that is poly-logarithmic in the size of the graph).

We provide two very different constructions, in tools and structure. The first, a direct construction, is based on proving a sufficient condition for robust self-ordering, which requires that an auxiliary graph is expanding. The second construction is iterative, boosting the property of robust self-ordering from smaller to larger graphs. Structuraly, the first construction always yields expanding graphs, while the second construction may produce graphs that have many tiny (sub-logarithmic) connected components.

We also consider graphs of unbounded degree, seeking correspondingly unbounded robustness parameters. We again demonstrate that such graphs (of linear degree) exist (in abundance), and that they can be constructed efficiently, in a strong sense. This turns out to require very different tools. Specifically, we show that the construction of such graphs reduces to the construction of non-malleable two-source extractors (with very weak parameters but with some additional natural features).

We demonstrate that robustly self-ordered bounded-degree graphs are useful towards obtaining lower bounds on the query complexity of testing graph properties both in the bounded-degree and the dense graph models. Indeed, their robustness offers efficient, local and distance preserving reductions from testing problems on ordered structures (like sequences) to the unordered (effectively unlabeled) graphs. One of the results that we obtain, via such a reduction, is a subexponential separation between the query complexities of testing and tolerant testing of graph properties in the bounded-degree graph model.

2012 ACM Subject Classification Theory of computation → Generating random combinatorial structures

Keywords and phrases Asymmetric graphs, expanders, testing graph properties, two-source extractors, non-malleable extractors, coding theory, tolerant testing, random graphs

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.12

Funding *Oded Goldreich*: Partially supported by an ISF grant number (Nr. 1146/18).
Avi Wigderson: Partially supported by NSF grant CCF-1900460.

Acknowledgements We are grateful to Eshan Chattopadhyay for discussions regarding non-malleable two-source extractors, and to Dana Ron for discussions regarding tolerant testing. We are deeply indebted to Jyun-Jie Liao for pointing out an error in the proof of [23, Lem. 9.7].

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 12; pp. 12:1–12:74
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

For a (labeled) graph $G = (V, E)$, and a bijection $\phi : V \to V'$, we denote by $\phi(G)$ the graph $G' = (V', E')$ such that $E' = \{\{\phi(u), \phi(v)\} : \{u, v\} \in E\}$, and say that G' is isomorphic to G. The set of automorphisms of the graph $G = (V, E)$, denoted $\mathtt{aut}(G)$, is the set of permutations that preserve the graph G; that is, $\pi \in \mathtt{aut}(G)$ if and only if $\pi(G) = G$. We say that a graph is asymmetric (equiv., self-ordered) if its set of automorphisms is a singleton, which consists of the trivial automorphism (i.e., the identity permutation). We actually prefer the term *self-ordered*, because we take the perspective that is offered by the following equivalent definition.

▶ **Definition 1.1** (self-ordered (a.k.a asymmetric) graphs)**.** *The graph $G = ([n], E)$ is* self-ordered *if for every graph $G' = (V', E')$ that is isomorphic to G there exists a unique bijection $\phi : V' \to [n]$ such that $\phi(G') = G$.*

In other words, given an isomorphic copy $G' = (V', E')$ of a fixed graph $G = ([n], E)$, there is a unique bijection $\phi : V' \to [n]$ that orders the vertices of G' such that the resulting graph (i.e., $\phi(G')$) is identical to G. Indeed, if $G' = G$, then this unique bijection is the identity permutation.[1]

In this work, we consider a feature, which we call *robust self-ordering*, that is a quantitative version self-ordering. Loosely speaking, a graph $G = ([n], E)$ is robustly self-ordered if, for every permutation $\pi : [n] \to [n]$, the size of the symmetric difference between G and $\pi(G)$ is proportional to the number of non-fixed-points under π; that is, $|E \triangle \{\{\pi(u), \pi(v)\} : \{u, v\} \in E\}|$ is proportional to $|\{i \in [n] : \pi(i) \neq i\}|$. (In contrast, self-ordering only means that the size of the symmetric difference is positive if the number of non-fixed-points is positive.)

▶ **Definition 1.2** (robustly self-ordered graphs)**.** *A graph $G = (V, E)$ is said to be γ-*robustly self-ordered *if for every permutation $\pi : V \to V$ it holds that*

$$\left| E \triangle \{\{\pi(u), \pi(v)\} : \{u, v\} \in E\} \right| \geq \gamma \cdot |\{i \in [n] : \pi(i) \neq i\}|, \qquad (1)$$

where $\triangle$ denotes the symmetric differece operation. An infinite family of graphs $\{G_n = ([n], E_n)\}_{n \in \mathbb{N}}$ (such that each G_n has maximum degree d) *is called* robustly self-ordered *if there exists a constant $\gamma > 0$, called the* robustness parameter*, such that for every n the graph G_n is γ-robustly self-ordered.*

Note that $|E_n \triangle \{\{\pi(u), \pi(v)\} : \{u, v\} \in E_n\}| \leq 2d \cdot |\{i \in [n] : \pi(i) \neq i\}|$ always holds (for families of maximum degree d). The term "robust" is inspired by the property testing literature (cf. [31]), where it indicates that some "parametrized violation" is reflected proportionally in some "detection parameter".

The second part of Definition 1.2 is tailored for bounded-degree graphs, which will be our focus in Section 2–6. Nevertheless, in Sections 7–10 we consider graphs of unbounded degree and unbounded robustness parameters. In this case, for a function $\rho : \mathbb{N} \to \mathbb{R}$, we say that an infinite family of graphs $\{G_n = ([n], E_n)\}_{n \in \mathbb{N}}$ is ρ-robustly self-ordered if for every n the graph G_n is $\rho(n)$-robustly self-ordered. Naturally, in this case, the graphs must have $\Omega(\rho(n) \cdot n)$ edges.[2] In Sections 7–9 we consider the case of $\rho(n) = \Omega(n)$.

[1] Naturally, we are interested in efficient algorithms that find this unique ordering, whenever it exists; such algorithms are known when the degree of the graph is bounded [29].

[2] Actually, all but at most one vertex must have degree at least $\rho(n)/2$.

1.1 Robustly self-ordered bounded-degree graphs

The first part of this paper (i.e., Section 2–6) focuses on the study of robustly self-ordered bounded-degree graphs.

1.1.1 Our main results and motivation

We show that robustly self-ordered (n-vertex) graphs of bounded-degree not only exist (for all $n \in \mathbb{N}$), but can be efficiently constructed in a strong (or local) sense. Specifically, we prove the following result.

▶ **Theorem 1.3** (constructing robustly self-ordered bounded-degree graphs). *For all sufficiently large $d \in \mathbb{N}$, there exist an infinite family of d-regular robustly self-ordered graphs $\{G_n\}_{n\in\mathbb{N}}$ and a polynomial-time algorithm that, given $n \in \mathbb{N}$ and a vertex $v \in [n]$ in the n-vertex graph G_n, finds all neighbors of v* (in G_n)*.*

We stress that the algorithm runs in time that is polynomial in the description of the vertex; that is, the algorithm runs in time that is polylogarithmic in the size of the graph. Theorem 1.3 holds both for graphs that consists of connected components of logarithmic size and for "strongly connected" graphs (i.e., expanders).

Recall that given an isomorphic copy G' of such a graph G_n, the original graph G_n (i.e., along with its unique ordering) can be found in polynomial-time [29]. Furthermore, we show that the pre-image of each vertex of G' in the graph G_n (i.e., its index in the aforementioned ordering) can be found in time that is polylogarithmic in the size of the graph (see discussion in Section 4.4, culminating in Theorem 4.7).[3]

We present two proofs of Theorem 1.3. Loosely speaking, the first proof reduces to proving that a $2d$-regular n-vertex graph representing the action of d permutations on $[n]$ is robustly self-ordered if the $n(n-1)$-vertex graph representing the action of these permutations on vertex-pairs is an expander. The graphs constructed in this proof are expanders, whereas the graphs constructed via by the second proof can be either expanders or consist of connected components of logarithmic size. More importantly, the graphs constructed in the second proof are couple with *local self-ordering and local reversed self-ordering algorithms* (see Section 4.4). The second proof proceeds in three steps, starting from the mere existence of robustly self-ordered bounded-degree ℓ-vertex graphs, which yields a construction that runs in poly(ℓ^ℓ)-time. Next, a poly(n)-time construction of n-vertex graphs is obtained by using the former graphs as small subgraphs (of $o(\log n)$-size). Lastly, strong (a.k.a local) constructability is obtained in an analogous manner. For more details, see Section 1.1.2.

We demonstrate that robustly self-ordered bounded-degree graphs are useful towards obtaining lower bounds on the query complexity of testing graph properties in the bounded-degree graph model. Specifically, we use these graphs as a key ingredient in a general methodology of transporting lower bounds regarding testing binary strings to lower bounds regarding testing graph properties in the bounded-degree graph model. In particular, using the methodology, we prove the following two results.

[3] The algorithm asserted above is said to perform *local self-ordering* of G' according to G_n. For $\phi(G') = G_n$, given a vertex v in G', this algorithm returns $\phi(v)$ in poly$(\log n)$-time. In contrast, a *local reversed self-ordering* algorithm is given a vertex $i \in [n]$ of G_n and returns $\phi^{-1}(i)$. The second algorithm is also presented in Section 4.4 (see Theorem 4.9).

1. A *subexponential separation between the complexities of testing and tolerant testing of graph properties in the bounded-degree graph model*; that is, for some constant $c > 0$, the query complexity of tolerant testing is at least $\exp(q^c)$, where q is the query complexity of standard testing.

   This result, which appears as Theorem 5.5, is obtained by transporting an analogous result that was known for testing binary strings [15].
2. A linear query complexity lower bound for testing an efficiently recognizable graph property in the bounded-degree graph model, where the lower bound holds even if the tested graph is restricted to consist of connected components of logarithmic size (see Theorem 5.2).

   As discussed in Section 5, an analogous result was known in the general case (i.e., without the restriction on the size of the connected components), and we consider it interesting that the result holds also in the special case of graphs with small connected components.

To get a feeling of why robustly self-ordered graphs are relevant to such transportation, recall that strings are ordered objects, whereas graphs properties are effectively sets of unlabeled graphs, which are unordered objects. Hence, we need to make the graphs (in the property) ordered, and furthermore make this ordering robust in the very sense that is reflected in Definition 1.2. Furthermore, local self-ordering algorithms are used for transporting lower bounds (and local reversed self-ordering algorithms are used for transporting upper bounds). We comment that the theme of reducing ordered structures to unordered structures occur often in the theory of computation and in logic, and is often coupled with analogous of query complexity.

Lastly, in Section 6, we prove that random $2d$-regular graphs are robustly self-ordered; see Theorem 6.1. This extends work in probabilistic graph theory, which proves a similar result for the weaker notion of self-ordering [5, 4].

1.1.2 Techniques

As stated above, we present two different constructions that establish Theorem 1.3: A direct construction and a three-step construction. Both constructions utilize a variant of the notion of robust self-ordering that refers to edge-colored graphs, which we review first.

1.1.2.1 The edge-coloring methodology

At several different points, we found it useful to start by demonstrating the robust self-ordering feature in a relaxed model in which edges are assigned a constant number of colors, and the symmetric difference between graphs accounts also for edges that have different colors in the two graphs (see Definition 2.1). This allows us to analyze different sets of edges separately.

For example, we actually analyze the direct construction in the edge-colored model, since this allows for identifying each of the underlying permutations with a different color. Another example, which arises in the three-step construction, occurs when we super-impose a robustly self-ordered graph with an expander graph in order to make the robustly self-ordered graph expanding (as needed for the second and third step of the aforementioned three-step construction). In this case, assigning the edges of each of the two graphs a different color, allows for easily retaining the robust self-ordering feature (of the first graph).

We obtain robustly self-ordered graphs (in the original sense) by replacing all edges that are assigned a specific color with copies of a constant-sized (asymmetric) gadget, where different (and in fact non-isomorphic) gadgets are used for different edge colors. The soundness of this transformation is proved in Theorem 2.4.

1.1.2.2 The direct construction

For any d permutations, $\pi_1, ..., \pi_d : [n] \to [n]$, we consider the *Schreier graph* (see [25, Sec. 11.1.2]) defined by the action of these permutation on $[n]$; that is, the edge-set of this graph is $\{\{v, \pi_i(v)\} : v \in [n] \,\&\, i \in [d]\}$. Loosely speaking, we prove that *this $2d$-regular n-vertex graph is robustly self-ordered if another Schreier graph is an expander.* The second Schreier graph represents the action of the same permutations on *pairs* of vertices (in $[n]$); that is, this graph consisting of the vertex-set $\{(u, v) : u, v \in [n]\}$ and the edge-set $\{\{(u, v), (\pi_i(u), \pi_i(v))\} : u, v \in [n] \,\&\, i \in [d]\}$.[4]

The argument is actually made with respect to edge-colored directed graphs (i.e., the edge-set of the first graph is $\{(v, \pi_i(v)) : v \in [n] \,\&\, i \in [d]\}$ and the directed edge $(v, \pi_i(v))$ is assigned the color i). Hence, we also present a transformation of robustly self-ordered edge-colored directed graphs to analogous undirected graphs. Specifically, we replace the directed edge (u, v) colored j by a 2-path with a designated auxiliary vertex $a_{u,v,j}$, while coloring the edge $\{u, a_{u,v,j}\}$ by $2j - 1$ and the edge $\{a_{u,v,j}, v\}$ by $2j$.

We comment that permutations satisfying the foregoing condition can be efficiently constructed; for example, any set of expanding generators for $\mathrm{SL}_2(p)$ (e.g., the one used by [28]) yield such permutations on $[n] \equiv \{(1, i) : i \in \mathrm{GF}(p)\} \cup \{(0, 1)\}$ (see Proposition 3.3).[5]

1.1.2.3 The three-step construction

Our alternative construction of robustly self-ordered (bounded-degree) n-vertex graphs proceeds in three steps.

1. First, we prove the existence of bounded-degree n-vertex graphs that are robustly self-ordered (see Theorem 4.1), while observing that this yields a $\exp(O(n \log n))$-time algorithm for constructing them.
2. Next (see Theorem 4.2), we use the latter algorithm to construct robustly self-ordered n-vertex bounded-degree graphs that consist of 2ℓ-sized connected components, where $\ell = \frac{O(\log n)}{\log \log n}$; these connected components are far from being isomorphic to one another, and are constructed using robustly self-ordered ℓ-vertex graphs as a building block. This yields an algorithm that constructs the n-vertex graph in $\mathrm{poly}(n)$-time, since $\exp(O(\ell \log \ell)) = \mathrm{poly}(n)$.
3. Lastly, we derive Theorem 1.3 (restated as Theorem 4.5) by repeating the same strategy as in Step 2, but using the construction of Theorem 4.2 for the construction of the small connected components (and setting $\ell = O(\log n)$). This yields an algorithm that finds the neighbors of a vertex in the n-vertex graph in $\mathrm{poly}(\log n)$-time, since $\mathrm{poly}(\ell) = \mathrm{poly}(\log n)$.

The foregoing description of Steps 2 and 3 yields graphs that consists of small connected components. We obtain analogous results for “strongly connected” graphs (i.e., expanders) by superimposing these graphs with expander graphs (while distinguishing the two types of edges by using colors (see the foregoing discussion)). In fact, it is essential to perform this transformation (on the result of Step 2) before taking Step 3; the transformation itself appears in the proof of Theorem 2.6.

[4] Equivalently, we consider only pairs of distinct vertices; that is, the vertex-set $\{(u, v) : u, v \in [n] \,\&\, u \neq v\}$.

[5] In this case, the primary Schreier graph represents the natural action of the group on the 1-dimensional subspaces of $\mathrm{GF}(p)^2$.

1.1.2.4 Using large collections of pairwise far apart permutations

One ingredient in the foregoing three-step construction is the use of a single ℓ-vertex robustly self-ordered (bounded-degree) graph towards obtaining a *large* collection of 2ℓ-vertex (bounded-degree) graphs such that every two graphs are far from being isomorphic to one another, where "large" means $\exp(\Omega(\ell \log \ell))$ in one case (i.e., in the proof of Theorem 4.2) and $\exp(\Omega(\ell))$ in another case (i.e., in the proof of Theorem 4.5). Essentially, this is done by constructing a large collection of permutations of $[\ell]$ that are pairwise far-apart, and letting the i^{th} graph consists of two copies of the ℓ-vertex graph that are matched according to the i^{th} permutation (see the aforementioned proofs). (Actually, we use two robustly self-ordered ℓ-vertex graphs that are far from being isomorphic (e.g., have different degree).)

A collection of $L = \exp(\Omega(\ell \log \ell))$ pairwise far-apart permutations over $[\ell]$ can be constructed in $\text{poly}(L)$-time by selecting the permutations one by one, while relying on the existence of a permutation that augments the current sequence (while preserving the distance condition, see the proof of Theorem 4.2). A collection of $L = \exp(\Omega(\ell))$ pairwise far-apart permutations over $[\ell]$ can be locally constructed such that the i^{th} permutation is constructed in $\text{poly}(\ell)$-time by using sequences of disjoint transpositions determined via a good error correcting code (see the proof of Theorem 4.5).

The foregoing discussion begs the challenge of obtaining a construction of a collection of $L = \exp(\Omega(\ell \log \ell))$ permutations over $[\ell]$ that are pairwise far-apart along with a polynomial-time algorithm that, on input $i \in [L]$, returns a description of the i^{th} permutation (i.e., the algorithm should run in $\text{poly}(\log L)$-time). We meet this challenge in [21]. Note that such a collection constitutes a an asymptotically good code over the alphabet $[\ell]$, where the permutations are the codewords (being far-apart corresponds to constant relative distance and $\log L = \Omega(\log(\ell!))$ corresponds to constant rate).

1.1.2.5 On the failure of some natural approaches

We mention that natural candidates for robustly self-ordered bounded-degree graphs fail. In particular, there exist expander graphs that are not robustly self-ordered. In fact, any Cayley graph is symmetric (i.e., has non-trivial automorphisms).[6]

In light of the above, it is interesting that expansion *can* serve as a sufficient condition for robust self-ordering (as explained in the foregoing review of the direct construction); recall, however, that this works for Schreier graphs, and expansion needs to hold for the action on vertex-pairs.

1.1.2.6 On optimization

We made no attempt to minimize the degree bound and maximize the robustness parameter. Note that we can obtain 3-regular robustly self-ordered graphs by applying degree reduction; that is, given a d-regular graph, we replace each vertex by a d-cycle and use each of these vertices to "hook" one original edge. To facilitate the analysis, we may use one color for the edges of the d-cycles and another color for the other (i.e., original) edges.[7] Hence, the issue at hand is actually one of maximizing the robustness parameter of the resulting 3-regular graphs.

[6] Specifically, multiplying the vertex labels (say, on the right) by any non-zero group element yields a non-trivial automorphism (assuming that edges are defined by multiplying with a generator on the left). Such automorphisms cannot be constructed in general for Schreier graphs, and some Schreier graphs have no automorphisms (e.g., the ones we construct here).

[7] Needless to say, we later replace all colored edges by copies of adequate constant-sized gadgets.

1.1.2.7 Caveat (tedious)

Whenever we assert a d-regular n-vertex graph, we assume that the trivial conditions hold; specifically, we assume that $n > d$ and that nd is even (or, alternatively, allow for one exceptional vertex of degree $d - 1$).

1.2 Robustly self-ordered dense graphs

In the second part of this paper (i.e., Sections 7–10) we consider graphs of unbounded degree, seeking correspondingly unbounded robustness parameters. In particular, we are interested in n-vertex graphs that are $\Omega(n)$-robustly self-ordered, which means that they must have $\Omega(n^2)$ edges.

The construction of $\Omega(n)$-robustly self-ordered graphs offers yet another alternative approach towards the construction of bounded-degree graphs that are $\Omega(1)$-robustly self-ordered. Specifically, we show that n-vertex graphs that are $\Omega(n)$-robustly self-ordered can be efficiently transformed into $O(n^2)$-vertex bounded-degree graphs that are $\Omega(1)$-robustly self-ordered; see Proposition 7.2, which is essentially proved by the "degree reduction via expanders" technique, while using a different color for the expanders' edges, and then using gadgets to replace colored edges (see Theorem 2.4).

1.2.1 Our main results

It is quite easy to show that random n-vertex graphs are $\Omega(n)$-robustly self-ordered (see Proposition 7.1); in fact, the proof is easier than the proof of the analogous result for bounded-degree graphs (Theorem 6.1). Unfortunately, constructing n-vertex graphs that are $\Omega(n)$-robustly self-ordered seems to be no easier constructing robustly self-ordered bounded-degree graphs. In particular, it seems to require completely different techniques and tools.

▶ **Theorem 1.4** (constructing $\Omega(n)$-robustly self-ordered graphs). *There exist an infinite family of dense $\Omega(n)$-robustly self-ordered graphs $\{G_n\}_{n\in\mathbb{N}}$ and a polynomial-time algorithm that, given $n \in \mathbb{N}$ and a pair of vertices $u, v \in [n]$ in the n-vertex graph G_n, determines whether or not u is adjacent to v in G_n.*

Unlike in the case of bounded-degree graphs, in general, we cannot rely on an efficient isomorphism test for finding the original ordering of G_n, when given an isomorphic copy of it. However, we can obtain dense $\Omega(n)$-robustly self-ordered graphs for which this ordering can be found efficiently (see Theorem 8.10).

Our proof of Theorem 1.4 is by a reduction to the construction of non-malleable two-source extractors, where a suitable construction of the latter was provided by Chattopadhyay, Goyal, and Li [7]. We actually present two different reductions (Theorems 8.3 and 8.7), one simpler than the other but yielding a less efficient construction when combined with the known constructions of extractors. We mention that the first reduction (Theorem 8.3) is partially reversible (see Proposition 8.5, which reverses a special case captured in Remark 8.4).

We show that $\Omega(n)$-robustly self-ordered n-vertex graphs can be used to transport lower bounds regarding testing binary strings to lower bounds regarding testing graph properties in the dense graph model. This general methodology, presented in Section 9, is analogous to the methodology for the bounded-degree graph model, which is presented in Section 5.

We mention that in a follow-up work [22], we employed this methodology in order to resolve several open problems regarding the relation between adaptive and non-adaptive testers in the dense graph model. In particular, we proved that there exist graph properties for which any non-adaptive tester must have query complexity that is almost quadratic in the query complexity of the best general (i.e., adaptive) tester, whereas it has been known for a couple of decades that the query complexity of non-adaptive testers is at most quadratic in the query complexity of adaptive testers.

The case of intermediate degree bounds

Lastly, in Section 10, we consider n-vertex graphs of degree bound $d(n)$, for every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \in [\Omega(1), n]$. Indeed, the bounded-degree case (studied in Section 2–6) and the dense graph case (studied in Sections 7–9) are special cases (which correspond to $d(n) = O(1)$ and $d(n) = n$). Using results from these two special cases, we show how to construct $\Omega(d(n))$-robustly self-ordered n-vertex graphs of maximum degree $d(n)$, for all $d : \mathbb{N} \to \mathbb{N}$.

1.2.2 Techniques

As evident from the foregoing description, we reduce the construction of $\Omega(n)$-robustly self-ordered n-vertex graphs to the construction of non-malleable two-source extractors.

Non-malleable two-source extractors were introduced in [8], as a variant on seeded (one-source) non-malleable extractors, which were introduced in [12]. Loosely speaking, we say that $\mathtt{nmE} : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^m$ is a non-malleable two-source extractor for a class of sources $\mathcal{C}$ if for every two independent sources in $\mathcal{C}$, denoted X ands Y, and for every two functions $f, g : \{0,1\}^\ell \to \{0,1\}^\ell$ that have no fixed-point it holds that $(\mathtt{nmE}(X,Y), \mathtt{nmE}(f(X), g(Y)))$ is close to $(U_m, \mathtt{nmE}(f(X), g(Y))$, where U_m denotes the uniform distribution over $\{0,1\}^m$. We show that a non-malleable two-source extractor for the class of ℓ-bit sources of min-entropy $\ell - O(1)$, with a single output bit (i.e., $m = 1$) and constant error, suffices for constructing $\Omega(n)$-robustly self-ordered n-vertex graphs. Recall that constructions with much stronger parameters (e.g., min-entropy $\ell - \ell^{\Omega(1)}$, negligible error, and $m = \ell^{\Omega(1)}$) were provided by Chattopadhyay, Goyal, and Li [7, Thm. 1]. (These constructions are quite complex. Interestingly, we are not aware of a simpler way of obtaining the weaker parameters that we need.)

Actually, we show two reductions of the construction of $\Omega(n)$-robustly self-ordered n-vertex graphs to the construction of non-malleable two-source extractors. In both cases we use extractors that operate on pairs of sources of length $\ell = \log_2 n - O(1)$ that have min-entropy $k = \ell - O(1)$, hereafter called (ℓ, k)-sources. The extractor is used to define a bipartite graph with 2^ℓ vertices on each side, and a clique is placed on the vertices of one side so that a permutation that maps vertices from one side to the other side yields a proportional symmetric difference (between the original graph and the resulting graph).

The first reduction, presented in Theorem 8.3, requires the extractor to be *quasi-orthogonal*, which means that the residual functions obtained by any two different fixings of one of the extractor's two arguments are almost unbiased and uncorrelated. Using the fact that non-malleable two-source extractors for (ℓ, k)-sources can we made quasi-orthogonal in $\exp(\ell)$-time, we obtain an explicit construction of $\Omega(n)$-robustly self-ordered n-vertex graphs (i.e., the n-vertex graph is constructed in $\mathrm{poly}(n)$-time).

The second reduction, presented in Theorem 8.7, yields a strongly explicit construction as asserted in Theorem 1.4 (i.e., the adjacency predicate of the n-vertex graph is computable in $\mathrm{poly}(\log n)$-time). This reduction uses an arbitrary non-malleable two-source extractor, and shifts the quasi-orthogonality condition to two auxiliary bipartite graphs.

Both reductions are based on the observation that if the number of non-fixed-points (of the permutation) is very large, then the non-malleability condition implies a large symmetric difference (between the original graph and the resulting graph). This holds as long as there are at least $\Omega(2^\ell)$ non-fixed-points on each of the two sides of the corresponding bipartite graph (which corresponds to the extractor). The complementary case is handled by the quasi-orthogonality condition, and this is where the two reductions differ.

The simpler case, presented in the first construction (i.e., Theorem 8.3), is that the extractor itself is quasi-orthogonal. In this case we consider the non-fixed-points on the side that has more of them. The quasi-orthogonality condition gives us a contribution of approximately $0.5 \cdot 2^\ell$ units per each non-fixed-point, whereas the upper-bound on the number of non-fixed-points on the other side implies that most of these contributions actually count in the symmetric difference (between the original graph and the resulting graph).

In the second construction (i.e., Theorem 8.7), we augment the foregoing 2^ℓ-by-2^ℓ bipartite graph, which is now determined by any non-malleable extractor, with an additional $4 \cdot 2^\ell$-vertex clique that is connected to the two original 2^ℓ-vertex sets by a bipartite graph that is merely quasi-orthogonal. The analysis is analogous to the one used in the proof of Theorem 8.3, but is slightly more complex because we are dealing with a slightly more complex graph.

Errata regarding the original posting

We retract the claims made in our initial posting [23] regarding the construction of non-malleable two-source extractors (which are quasi-orthogonal) as well as the claims about the construction of relocation-detecting codes (see Theorems 1.5 and 1.6 in the original version).[8] The source of trouble is a fundamental flaw in the proof of [23, Lem. 9.7], which may as well be wrong.

1.3 Perspective

Asymmetric graphs were famously studied by Erdos and Renyi [14], who considered the (absolute) distance of asymmetric graphs from being symmetric (i.e., the number of edges that should be removed or added to a graph to make it symmetric), calling this quantity the degree of asymmetry. They studied the extremal question of determining the largest possible degree of asymmetry of n-vertex graphs (as a function of n). We avoided the term "robust asymmetry" because it could be confused with the degree of asymmetry, which is a very different notion. In particular, the degree of asymmetry cannot exceed twice the degree of the graph (e.g., by disconnecting two vertices), whereas our focus is on robustly self-ordered graphs of bounded-degree.

We mention that Bollobas proved that, *for every constant $d \geq 3$, almost all d-regular graphs are asymmetric* [5, 4]. This result was extended to varying $d \in [3, n-4]$ by Kim, Sudakov, and Vu [26]. We also mention that their proof of [26, Thm. 3.1] implies that a random n-vertex Erdos–Renyi graph with edge probability p is $2p(1-p)n$-robustly self-ordered.

1.4 Roadmaps

This work consists of two parts. The first part (Sections 2–6) refers to bounded-degree graphs, and the second part (Sections 7–10) refers to dense graphs. These parts are practically independent of one another, except that Theorem 10.3 builds upon Section 6. Even when focusing on one of these two parts, its contents may attract attention from diverse perspectives. Each such perspective may benefit from a different roadmap.

[8] In [23] quasi-orthogonality is called niceness; we prefer the current term, which is less generic.

Efficient combinatorial constructions

As mentioned above, in the regime of bounded-degree graphs we present two different constructions that establish Theorem 1.3. Both constructions make use of the edge-colored model and the transformations presented in Section 2. The direct construction is presented in Section 3, and the three-step construction appears in Section 4. The three-step construction is augmented by local self-ordering and local reversed self-ordering algorithms (see Section 4.4).[9] In the regime of dense graphs, Sections 7 and 8 refer to the constructability of a couple of combinatorial objects; see roadmap "for the dense case" below.

Potential applications to property testing

In Section 5 we demonstrate applications of Theorem 1.3 to proving lower bounds (on the query complexity) for the bounded-degree graph testing model. Specifically, we present a methodology of transporting bounds regarding testing properties of strings to bounds regarding testing properties of bounded-degree graphs. The specific applications presented in Section 5 rely on Section 4. For the first application (Theorem 5.2) the construction presented in Section 4.2 suffices; for the second application (i.e., Theorem 5.5, which establishes a separation between testing and tolerant testing in the bounded-degree graph model), the local computation tasks studied in Section 4.4 are needed. An analogous methodology for the dense graph testing model is presented in Section 9.

Properties of random graphs

As stated above, it turns out that random $O(1)$-regular graphs are robustly self-ordered. This result is presented in Section 6, and this section can be read independently of any other section. (In addition, Section 7 presents a proof that random (dense) n-vertex graphs are $O(n)$-robustly self-ordered.)

The dense case and non-malleable two-source extractors

The regime of dense graphs is studied in Sections 7–9, where the construction of such graphs is undertaken in Section 8. In Section 7, we show that $\Omega(n)$-robustly self-ordered n-vertex graphs provide yet another way of obtaining $\Omega(1)$-robustly self-ordered bounded-degree graphs. In Section 8, we reduce the construction of $O(n)$-robustly self-ordered n-vertex graphs to the construction of non-malleable two-source extractors. As outlined in Section 1.2.2, we actually present two different reductions, where a key issue is the quasi-orthogonality condition.

Lastly, in Section 10, for every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \in [\Omega(1), n]$, we show how to construct n-vertex graphs of maximum degree $d(n)$ that are $\Omega(d(n))$-robustly self-ordered. Some of the results and techniques presented in this section are also relevant to the setting of bounded-degree graphs.

[9] For a locally constructable G_n and $G' = \phi^{-1}(G_n)$, a *local self-ordering* algorithm is given a vertex v in G', and returns $\phi(v)$. In contrast, a *local reversed self-ordering* algorithm is given a vertex $i \in [n]$ of G_n and returns $\phi^{-1}(i)$. Both algorithms run in poly$(\log n)$-time.

Part I
The Case of Bounded-Degree Graphs

As stated in Section 1.1.2, a notion of robust self-ordering of edge-colored graphs plays a pivotal role in our study of robustly self-ordered bounded-degree graphs. This notion as well as a transformation from it to the uncolored version (of Definition 1.2) is presented in Section 2.

In Section 3, we present a direct construction of $O(1)$-regular robustly self-ordered edge-colored graphs; applying the foregoing transformation, this provides our first proof of Theorem 1.3. Our second proof of Theorem 1.3 is presented in Section 4, and consists of a three-step process (as outlined in Section 1.1.2). Sections 3 and 4 can be read independently of one another, but both rely on Section 2.

In Section 5 we demonstrate the applicability of robustly self-ordered bounded-degree graphs to property testing; specifically, to proving lower bounds (on the query complexity) for the bounded-degree graph testing model. For these applications, the global notion of constructability, established in Section 4.2, suffices. This construction should be preferred over the direct construction presented in Section 3, because it yields graphs with small connected components. More importantly, the subexponential separation between the complexities of testing and tolerant testing of graph properties (i.e., Theorem 5.5) relies on the construction of Section 4 and specifically on the local computation tasks studied in Section 4.4.

Lastly, in Section 6, we prove that random $O(1)$-regular graphs are robustly self-ordered. This section may be read independently of any other section.

2 The Edge-Colored Variant

Many of our arguments are easier to make in a model of (bounded-degree) graphs in which edges are colored (by a bounded number of colors), and where one counts the number of mismatches between colored edges. Namely, an edge that appears in one (edge-colored) graph contributes to the count if it either does not appear in the other (edge-colored) graph or appears in it under a different color. Hence, we define a notion of robust self-ordering for edge-colored graphs. We shall then transform robustly self-ordered edge-colored graphs to robustly self-ordered ordinary (uncolored) graphs, while preserving the degree, the asymptotic number of vertices, and other features such as expansion and degree-regularity. Specifically, the transformation consists of replacing the colored edges by copies of different connected, asymmetric (constant-sized) gadgets such that different colors are reflected by different gadgets.

We start by providing the definition of the edge-colored model. Actually, for greater flexibility, we will consider multi-graphs; that is, graphs with possible parallel edges and self-loops. Hence, we shall consider multi-graphs $G = (V, E)$ coupled with an edge-coloring function $\chi : E \to \mathbb{N}$, where E is a multi-set containing both pairs of vertices and singletons (representing self-loops). Actually, it will be more convenient to represent self-loops as 2-element multi-sets containing two copies of the same vertex.

▶ **Definition 2.1** (robust self-ordering of edge-colored multi-graphs). *Let $G = (V, E)$ be a multi-graph with colored edges, where $\chi : E \to \mathbb{N}$ denotes this coloring, and let E_i denote the multi-set of edges colored i* (i.e., $E_i = \{e \in E : \chi(e) = i\}$). *We say that (G, χ) is γ-*robustly self-ordered *if for every permutation $\mu : V \to V$ it holds that*

$$\sum_{i\in\mathbb{N}} \Big| E_i \bigtriangleup \{\{\mu(u), \mu(v)\} : \{u, v\} \in E_i\} \Big| \geq \gamma \cdot |\{i \in V : \mu(i) \neq i\}|, \tag{2}$$

where $A \triangle B$ denotes the symmetric difference between the multi-sets A and B; that is $A \triangle B$ contains t occurrences of e if the absolute difference between the number of occurrences of e in A and B equals t.

(Definition 1.2 is obtained as a special case when the multi-graph is actually a graph and all edges are assigned the same color.)

We stress that whenever we consider "edge-colored graphs" we actually refer to edge-colored multi-graphs (i.e., we explicitly allow parallel edges and self-loops).[10] In contrast, whenever we consider (uncolored) graph, we refer to simple graphs (with no parallel edges and no self-loops).

Our transformation of robustly self-ordered edge-colored multi-graphs to robustly self-ordered ordinary graphs depends on the number of colors used by the multi-graph. In particular, γ-robustness of edge-colored multi-graph that uses c colors gets translated to $(\gamma/f(c))$-robustness of the resulting graph, where $f : \mathbb{N} \to \mathbb{N}$ is an unbounded function. Hence, we focus on coloring functions that use a constant number of colors, denoted c. That is, fixing a constant $c \in \mathbb{N}$, we shall consider multi-graphs $G = (V, E)$ coupled with an edge-coloring function $\chi : E \to [c]$.

2.1 Transformation to standard (uncolored) version

As a preliminary step for the transformation, we add self-loops to all vertices and make sure that parallel edges are assigned different colors. The self-loops make it easy to distinguish the original vertices from auxiliary vertices that are parts of gadgets introduced in the main transformation. Different colors assigned to parallel edges are essential to the mere asymmetry of the resulting graph, since we are going to replace edges of the same color by copies of the same gadget.

▶ **Construction 2.2** (preliminary step towards Construction 2.3)**.** *For a fixed $d \geq 3$, given a multi-graph $G = (V, E)$ of maximum degree d and an edge-coloring function $\chi : E \to [c]$, we define a multi-graph $G = (V, E')$ and an edge-coloring function $\chi' : E' \to [d \cdot c + 1]$ as follows.*

1. *For every pair of vertices u and v that are connected by few parallel edges, denoted $e_{u,v}^{(1)}, ..., e_{u,v}^{(d')}$, we change the color of $e_{u,v}^{(i)}$ to $\chi'(e_{u,v}^{(i)}) \leftarrow (i-1) \cdot d + \chi(e_{u,v}^{(i)})$. This includes also the case $u = v$.*
2. *We augment the multi-graph with self-loops colored $d \cdot c + 1$; that is, E' is the multi-set $E \cup \{e_v : v \in V\}$, where e_v is a self-loop added to v, and $\chi'(e_v) = dc + 1$.*

(Other edges $e \in E$ maintain their color; that is, them $\chi'(e) = \chi(e)$ holds).

(For simplicity, we re-color all parallel edges, save the first one, rather than re-coloring only parallel edges of the same color.) Note that refining the coloring may only increase the robustness parameter of a multi-graph. Clearly, G' preserves many features of G. In particular, it preserves γ-robust self-ordering, expansion, degree-regularity, and the number of vertices.

[10] We comment that a seemingly more appealing definition can be used for edge-colored (simple) graphs. Specifically, in that case (i.e., $E \subseteq \binom{V}{2}$), we can extend $\chi : E \to \mathbb{N}$ to non-edges by defining $\chi(\{u,v\}) = 0$ if $\{u,v\} \notin E$, and say that (G, χ) is γ-robustly self-ordered if for every permutation $\mu : V \to V$ it holds that

$$\left| \left\{ \{u,v\} \in \binom{V}{2} : \chi(\{\mu(u), \mu(v)\}) \neq \chi(\{u,v\}) \right\} \right| \geq \gamma \cdot |\{i \in V : \mu(i) \neq i\}|.$$

As stated above, our transformation of edge-colored multi-graphs to ordinary graphs uses gadgets, which are constant-size graphs. Specifically, when handling a multi-graph of maximum degree d with edges that are colored by c colors, we shall use c different *connected and asymmetric* graphs. Furthermore, in order to maintain d-regularity, we shall use d-regular graphs as gadgets; and in order to have better control on the number of vertices in the resulting graph, each of these gadgets will contain $k = k(d,c)$ vertices. The existence of such (d-regular) asymmetric (and connected) graphs is well-known, let alone that it is known that a random d-regular k-vertex graph is asymmetric (for any constant $d \geq 3$) [5, 4].

We stress that the different gadgets are each connected and asymmetric, and it follows that they are not isomorphic to one another. We designate in each gadget an edge $\{p,q\}$, called the designated edge, such that omitting this edge does not disconnect the gadget. The endpoint of this edge will be used to connect two vertices of the original multi-graph. Specifically, we replace each edge $\{u,v\}$ (of the original multi-graph) that is colored i by a copy of the i^{th} gadget, while omitting one its designated edge $\{p,q\}$ and connecting u to p and v to q. The construction is spelled out below.

We say that a (non-simple) multi-graph $G = (V,E)$ coupled with an edge-coloring χ is eligible if each of its vertices contains a self-loop, and parallel edges are assigned different colors. Recall that eligible comes almost for free (by applying Construction 2.2). We shall apply the following construction only to eligible edge-colored multi-graphs.

▶ **Construction 2.3** (the main transformation). *For a fixed $d \geq 3$ and c, let $k = k(d,c)$ and $G_1, ..., G_k$ be different asymmetric and connected d-regular graphs over the vertex-set $[k]$. Given a multi-graph $G = (V,E)$ of maximum degree d and an edge-coloring function $\chi : E \to [c]$, we construct a graph $G' = (V', E')$ as follows.*

> *Suppose that the multi-set E has size m. Then, for each $j \in [m]$, if the j^{th} edge of E connects vertices u and v, and is colored i, then we replace it by a copy of G_i, while omitting its designated edge and connecting one of its endpoints to u and the other to v.*
>
> *Specifically, assuming that $V = [n]$ and recalling that j is the index of the edge* (colored i) *that connects u and v, let $G_i^{u,v}$ be an isomorphic copy of G_i that uses the vertex set $\{n + (j-1) \cdot k + i : i \in [k]\}$. Let $\{p,q\}$ be the designated edge in $G_i^{u,v}$, and $\hat{G}_i^{u,v}$ be the graph that results from $G_i^{u,v}$ by omitting $\{p,q\}$. Then, we replace the edge $\{u,v\}$ by $\hat{G}_i^{u,v}$, and add the edges $\{u,p\}$ and $\{v,q\}$.*

Hence, $V' = [n + m \cdot k]$ and E' consists of the edges of all $\hat{G}_i^{u,v}$'s as well as the edges connecting the endpoint of the corresponding designated edges to the corresponding vertices u and v.

We stress that, although G may have parallel edges and self-loops, the graph G' has neither parallel edges nor self-loops. Also note that G' preserve various properties of G such as degree-regularity, number of connected components, and expansion (up to a constant factor).

Showing that the resulting graph $G' = (V', E')$ is robustly self-ordered relies on a correspondence between the colored edges of $G = (V,E)$ and the gadgets in G'. For starters, suppose that the permutation $\mu' : V' \to V'$ maps V to V (i.e., $\mu'(V) = V$), and gadgets to the corresponding gadgets; that is, if μ' maps the vertex-pair $(u,v) \in V^2$ to $(\mu'(u), \mu'(v)) \in V^2$, then μ' maps the vertices in the possible gadget that connects u and v to the vertices in the gadget that connects $\mu'(u)$ and $\mu'(v)$. In such a case, letting μ be the restriction of μ' to V, a difference of D colored edges between G and $\mu(G)$ translates to a difference of at least D edges between G' and $\mu'(G')$, due to the difference between the gadgets that replace the corresponding edges of G', whereas the number of non-fixed-point vertices in μ' is k times larger than the number of non-fixed-point vertices in μ, which is at most D/γ (by the γ-robust self-ordering of G). Hence, in this case we have

$$\frac{|G' \triangle \mu'(G')|}{|\{v \in V' : \mu'(v) \neq v\}|} = \frac{D}{k \cdot |\{v \in V : \mu(v) \neq v\}|} \geq \frac{D}{k \cdot D/\gamma}$$

which equals γ/k. However, in general, μ' needs not satisfy the foregoing condition. Nevertheless, if μ' splits some gadget or maps some gadget in a manner that is inconsistent with the vertices of V connected by it, then this gadget contributes at least one unit to the difference between G' and $\mu'(G')$, whereas the number of non-fixed-point vertices in this gadget is at most k. Lastly, if μ' maps vertices of a gadget to other vertices in the same gadget, then we get a contribution of at least one unit due to the asymmetry of the gadget. The foregoing is made rigorous in the proof of the following theorem.

▶ **Theorem 2.4** (from edge-colored robustness to standard robustness)**.** *For constant $d \geq 3$ and c, suppose that the multi-graph $G = (V, E)$ coupled with $\chi : E \to [c]$ is eligible and γ-robustly self-ordered. Then, the graph $G' = (V', E')$ resulting from Construction 2.3 is $(\gamma/3k)$-robustly self-ordered, where $k = k(d, c)$ is the number of vertices in a gadget* (as determined above).

Proof. As a warm-up, let us verify that G' is asymmetric. We first observe that the vertices of G are uniquely identified (in G'), since they are the only vertices that are incident at copies of the gadget that replaces the self-loops.[11] Hence, any automorphism of G' must map V to V. Consequently, for any i, such an automorphism μ' must map each copy of G_i to a copy of G_i, which induces a unique coloring of the edges of G. By the "colored asymmetry" of G, this implies that μ' maps each $v \in V$ to itself, and consequently each copy of G_i must be mapped (by μ') to itself. Finally, using the asymmetry of the G_i's, it follows that each vertex of each copy of G_i is mapped to itself.

We now turn to proving that G' is actually robustly self-ordered. Considering an arbitrary permutation $\mu' : V' \to V'$, we lower-bound the distance between G' and $\mu'(G')$ as a function of the number of non-fixed-points under μ' (i.e., of $v \in V'$ such that $\mu'(v') \neq v'$). We do so by considering the contribution of each non-fixed-point to the distance between G' and $\mu'(G')$. We first recall the fact that the vertices of V (resp., of gadgets) are uniquely identified in $\mu'(G')$ by virtue of the gadgets that replace self-loops (see the foregoing warm-up).

Case 1: *Vertices of some copy of G_i that are not mapped by μ' to a single copy of G_i; that is, vertices in some $G_i^{u,v}$ that are not mapped by μ' to some $G_i^{u',v'}$.*
(This includes the case of vertices w' and w'' of some $G_i^{u,v}$ such that $\mu'(w')$ is in $G_{i'}^{u',v'}$ and $\mu'(w'')$ is in $G_{i''}^{u'',v''}$, but $(i', u', v') \neq (i'', u'', v'')$. It also includes the case of a copy of G_i that is mapped by μ' to a copy of G_j for $j \neq i$, and the case that a vertex w in some $G_i^{u,v}$ that is mapped by μ' to a vertex in V.)
The set of vertices $S_i^{u,v}$ of each such copy (i.e., $G_i^{u,v}$) contribute at least one unit to the difference between G' and $\mu'(G')$, since $\mu'(S_i^{u,v})$ induces a copy of $\hat{G}_i$ in $\mu(G')$ but not in G', where here we also use the fact that the $\hat{G}_i$'s are connected (and not isomorphic (for the case of $i' = i'' \neq i$)). Note that the total contribution of all vertices of the current case equals at least the number of gadgets in which they reside. Hence, if the current case contains n_1 vertices, then their contribution to the distance between G' and $\mu'(G')$ is at least n_1/k.

[11] Note that this gadget cannot appear as part of any other gadget, since all gadgets have the same number of vertices.

Ditto for vertices that do not belong to a single copy of G_i and are mapped by μ' to a single copy of G_i. (This also includes $v \in V$ being mapped to some copy of some G_i.)

Case 2: *Vertices of some copy of G_i that are mapped by μ' to a single copy of G_i, while not preserving their indices inside G_i.*
(This refers to vertices of some $G_i^{u,v}$ that are mapped by μ to vertices of $G_i^{u',v'}$, where (u',v') may but need not equal (u,v), such that for some $j \in [k]$ the j^{th} vertex of $G_i^{u,v}$ is not mapped by μ to the j^{th} vertex of $G_i^{u',v'}$.)[12]
By the fact that G_i is asymmetric, it follows that each such copy contributes at least one unit to the difference between G' and $\mu'(G')$, and so (again) the total contribution of all these vertices is proportional to their number; that is, if the number of vertices in this case is n_2, then their contribution is at least n_2/k.

Case 3: *Vertices $v \in V$ such that $\mu'(v) \neq v$* (equiv., $\mu'(v) \in V \setminus \{v\}$).
(This is the main case, where we use the hypothesis that the edge-colored G is robustly self-ordered.
By the hypothesis that the edge-colored G is robustly self-ordered, it follows that such vertices contribute proportionally to the difference between the colored versions of the multi-graphs G and $\mu(G)$, where μ is the restriction of μ' to V. Specifically, the number of tuples $(\{u,v\},i)$ such that $\{u,v\}$ is colored i in exactly one of these multi-graph (i.e., either in G or in $\mu(G)$ but not in both) is at least $\gamma \cdot |\{v \in V : \mu(v) \neq v\}|$. Assume, without loss of generality that $\chi(\{u,v\}) = i$ but either $\{\mu^{-1}(u), \mu^{-1}(v)\} \notin E$ or $\chi(\{\mu^{-1}(u), \mu^{-1}(v)\}) = j \neq i$. Either way, it follows that some vertices that do not belong to a copy of G_i are mapped by μ' to $G_i^{u,v}$, which means that Case 1 applies for each such a tuple. Hence, if the number of vertices in the current case is n_3, then $n_1 \geq \gamma \cdot n_3$, and we get a contribution of at least $\gamma \cdot n_3/k$ via Case 1.

Case 4: *Vertices of some copy of G_i that are mapped by μ' to a different copy of G_i.*
This refers to the case that μ' maps $G_i^{u,v}$ to $G_i^{u',v'}$ such that $(u',v') \neq (u,v)$, which corresponds to mapping the gadget to a gadget connecting a different pair of vertices (but by an edge of the same color).
For u, v, u', v' and i as above, if $\mu'(u) = u'$ and $\mu'(v) = v'$, then a gadget that connects u and v in G' is mapped to a gadget that does not connects them in $\mu'(G')$ (but rather connects the vertices u' and v', whereas either $u' \neq u$ or $v' \neq v$). So we get a contribution of at least one unit to the difference between G' and $\mu'(G')$ (i.e., the gadget-edge incident at either u or v), whereas the number of vertices in this gadget is k. Hence, the contribution is proportional to the number of non-fixed-points of the current type. Otherwise (i.e., $(\mu'(u), \mu'(v)) \neq (u', v')$), we get a vertex as in Case 3, and get a proportional contribution again.

Hence, the contribution of each of these cases to the difference between G' and $\mu'(G')$ is proportional to the number of vertices involved. Specifically, if there are n_i vertices in Case i, then we get a contribution-count of at least $\gamma \cdot \sum_{i\in[4]} n_i/k$, where some of these contributions were possibly counted thrice. The claim follows. ◀

▶ **Remark 2.5** (fitting any desired number of vertices). Assuming that the hypothesis of Theorem 2.4 can be met for any sufficiently large $n \in S \subseteq \mathbb{N}$, Construction 2.3 yields robustly self-ordered n'-vertex graphs for any $n' \in \{k \cdot n : n \in S\}$. To obtain such graphs also for n' that is not a multiple of k, we may use two gadgets with a different number of vertices for replacing at least one of the sets of colored edges.

[12] Recall that $G_i^{u,v}$ and $G_i^{u',v'}$ are both copies of the k-vertex graph G_i, which is an asymmetric graph, and so the notion of the j^{th} vertex in them is well-defined. Formally, the j^{th} vertex of $G_i^{u,v}$ is $\phi^{-1}(j)$ such that ϕ is the (unique) bijection satisfying $\phi(G_i^{u,v}) = G_i$.

2.2 Application: Making the graph regular and expanding

We view the edge-colored model as an intermediate locus in a two-step methodology for constructing robustly self-ordered graphs of bounded-degree. First, one constructs edge-colored multi-graphs that are robustly self-ordered in the sense of Definition 2.1, and then converts them to ordinary robustly self-ordered graphs (in the sense of Definition 1.2), by using Construction 2.3 (while relying on Theorem 2.4).

We demonstrate the useful of this methodology by showing that it yields a simple way of making robustly self-ordered graphs be also expanding as well as regular, while maintaining a bounded degree. We just augment the original graph by super-imposing an expander (on the same vertex set), while using one color for the edges of the original graph and another color for the edges of the expander. Note that we do not have to worry about the possibility of creating parallel edges (since they are assigned different colors). The same method applies in order to make the graph regular. We combine both transformations in the following result, which we shall use in the sequel.

▶ **Theorem 2.6** (making the graph regular and expanding). *For constant $d \geq 3$ and γ, there exists an efficient algorithm that given a γ-robustly self-ordered graph $G = (V, E)$ of maximum degree d, returns a $(d + O(1))$-regular multi-graph coupled with a 2-coloring of its edges such that the edge-colored graph is γ-robustly self-ordered* (in the sense of Definition 2.1)*.*

The same idea can be applied to edge-colored multi-graphs; in this case, we use one color more than given. We could have avoided the creation of parallel edges with the same color by using more colors, but preferred to relegate this task to Construction 2.2, while recalling that it preserves both the expansion and the degree-regularity. Either way, applying Theorem 2.4 to the resulting edge-colored multi-graph, we obtain robustly self-ordered (uncolored) graphs.

Proof. For any $d'' \geq d+d'$, given a graph $G = (V, E)$ of maximum degree d that is γ-robustly self-ordered and a d'-regular expander graph $G' = (V, E')$, we construct the desired d''-regular multi-graph G'' by super-imposing the two graphs on the same vertex set, while assigning the edges of each of these graphs a different color. In addition, we add edges to make the graph regular, and color them using the same color as used for the expander.[13] Details follow.

- We superimpose G and G' (i.e., create a multi-graph $(V, E \cup E')$), while coloring the edges of G (resp., G') with color 1 (resp., color 2).

  Note that this may create parallel edges, but with different colors.
- Let $d_v \leq d + d'$ denote the degree of vertex v in the resulting multi-graph. Then, we add edges to this multi-graph so that each vertex has degree d''. These edges will also be colored 2.

  (Here, unless we are a bit careful, we may introduce parallel edges that are assigned the same color. This can be avoided by using more colors for these added edges, but in light of Construction 2.2 (which does essentially the same) there is no reason to worry about this aspect.)

(Recall that the resulting edge-colored multi-graph is denoted G''.)

[13] We assume for simplicity that $|V'|$ is even. Alternatively, assuming that G contains no isolated vertex, we first augment it with an isolated vertex and apply the transformation on the resulting graph. Yet another alternative is to consider only even d''.

The crucial observation is that, since the edges of G are given a distinct color in G'', the added edges do not harm the robust self-ordering feature of G. Hence, for any permutation $\mu : V \to V$, any vertex-pair that contributes to the symmetric difference between G and $\mu(G)$, also contributes to an inequality between colored edges of G'' and $\mu(G'')$ (by virtue of the edges colored 1). ◀

2.3 Local computability of the transformations

In this subsection, we merely point out that the transformation presented in Constructions 2.2 and 2.3 as well as the one underlying the proof of Theorem 2.6 preserve efficient local computability (e.g., one can determine the neighborhood of a vertex in the resulting multi-graph by making a polylogarithmic number of neighbor-queries to the original multi-graph). Actually, this holds provided that we augment the (local) representation of graphs, in a natural manner.

Recall that the standard representation of bounded-degree graphs is by their incidence functions. Specifically, a graph $G = ([n], E)$ of maximum degree d is represented by the incident function $g : [n] \times [d] \to [n] \cup \{0\}$ such that $g(v, i) = u \in [n]$ if u is the i^{th} neighbor of v, and $g(v, i) = 0$ if v has less than i neighbors. This does not allow us to determined the identity of the j^{th} edge in G, nor even to determine the number of edges in G, by making a polylogarithmic number of queries to g. Nevertheless, efficient local computability is preserved if we use the following local representation (presented for edge-colored multi-graphs).

▶ **Definition 2.7** (local representation). *For $d, c \in \mathbb{N}$, a* local representation *of a multi-graph $G = ([n], E)$ of maximum degree d that is coupled with a coloring $\chi : E \to [c]$ is provided by the following three functions:*

1. *An incidence function $g_1 : [n] \times [d] \to \mathbb{N} \cup \{0\}$ such that $g_1(v, i) = j \in \mathbb{N}$ if j is the index of the i^{th} edge that incident at vertex v, and $g_1(v, i) = 0$ if v has less than i incident edges.*
2. *An edge enumeration function $g_2 : \mathbb{N} \to ([n]^2 \times [c]) \cup \{0\}$ such that $g_2(j) = (u, v, \chi(e_j)$ if the j^{th} edge, denoted e_j, connects the vertices u and v, and $g_2(j) = 0$ if the multi-graph has less than j edges.*
3. *An vertex enumeration* (by degree) *function $g_3 : [d] \to ([n] \to [n]) \cup \{0\}$ such that $g_3(i, j) = v \in [n]$ if v is the i^{th} vertex of degree j in the multi-graph, and $g_3(i, j) = 0$ if the multi-graph has less than j vertices of degree i.*

Needless to say, the function g_3 is redundant in the case that we are guaranteed that the multi-graph is regular. One may augment the above representation by providing also the total number of edges, but this number can be determined by binary search.

▶ **Theorem 2.8** (the foregoing transformations preserve local computability). *The local representation of the multi-graph that result from Construction 2.2 can be computed by making a polylogarithmic number of queries to the given multi-graph. The same holds for Construction 2.3 and for the transformation underlying the proof of Theorem 2.6.*

Proof. For Construction 2.2, we mostly need to enumerate all parallel edges that connect u and v. This can be done easily by querying the incidence function on $(u, 1), ..., (u, d)$ and querying the edge enumeration function on the non-zero answers. (In addition, when adding a self-loop on vertex $v \in [n]$, we need to determine the degree of v as well as the number of edges in the multi-graph (in order to know how to index the self-loop in the incidence and edge enumeration functions, respectively). For Construction 2.3, we merely need to determine the color of the j^{th} edge and its index in the incidence list of each of its endpoints (in order to replace it by edges that lead to the gadget).

For the transformation underlying the proof of Theorem 2.6, adding edges to make the multi-graph regular requires determining the index of a vertex in the list of all vertices of the same degree (in order to properly index the added edges). Here is where we use the vertex enumeration (by degree) function. (We also need to select a fixed procedure for transforming an sorted n-long sequence $(d_1, ..., d_n) \in [d'']$ into an all-d'' sequence by making pairs of increments; that is, given $j \in [D]$ such that $D = (d''n - \sum_{i\in[n]} d_i)/2$, we should determine a pair (u_j, v_j) such that for every $i \in [n]$ it holds that $d_i + |\{j : u_j = i\}| + |\{j : v_j = i\}| = d''$.) ◀

3 The Direct Construction

We shall make use of the edge-colored variant presented in Section 2, while relying on the fact that robustly self-ordered colored multi-graphs can be efficiently transformed into robustly self-ordered (uncolored) graphs. Actually, it will be easier to present the construction as a directed edge-colored multi-graph. Hence, we first define a variant of robust self-ordering for directed edge-colored multi-graph (see Definition 3.1), then show how to construct such multi-graphs (see Section 3.2), and finally show how to transform the directed variant into an undirected one (see Section 3.1).

The construction is based on d permutations, denoted $\pi_1, ..., \pi_d : [n] \to [n]$, and consists of the directed edge-colored multi-graph that is naturally defined by them. Specifically, for every $v \in [n]$ and $i \in [d]$, this multi-graph contains a directed edge, denoted $(v, \pi_i(v))$, that goes from vertex v to vertex $\pi_i(v)$, and is colored i.

We prove that a sufficient condition for this edge-colored directed multi-graph, denoted G_1, to be robustly self-ordered is that a related multi-graph is an expander. Specifically, we refer to the multi-graph $G_2 = (V_2, E_2)$ that represents the actions of the permutation of pairs of vertices of G_1; that is, $V_2 = \{(u,v) \in [n]^2 : u \neq v\}$ and $E_2 = \{\{(u,v), (\pi_i(u), \pi_i(v))\} : (u,v) \in V_2 \,\&\, i \in [d]\}$.

The foregoing requires extending the notion of robustly self-ordered (edge-colored) multi-graphs to the directed case. The extension is straightforward and is spelled-out next, for sake of good order.

▶ **Definition 3.1** (robust self-ordering of edge-colored directed multi-graphs). *Let $G = (V, E)$ be a directed multi-graph with colored edges, where $\chi : E \to \mathbb{N}$ denotes this coloring, and let E_i denote the multi-set of edges colored i. We say that (G, χ) is γ-*robustly self-ordered *if for every permutation $\mu : V \to V$ it holds that*

$$\sum_{i\in\mathbb{N}} \left| E_i \,\triangle\, \{(\mu(u), \mu(v)) : (u,v) \in E_i\} \right| \;\geq\; \gamma \cdot |\{i \in V : \mu(i) \neq i\}|, \tag{3}$$

where $A \triangle B$ denotes the symmetric difference between the multi-sets A and B (as in Definition 2.1).

(The only difference between Definition 3.1 and Definition 2.1 is that (3) refers to the directed edges of the directed multi-graph, whereas (2) refers to the undirected edges of the undirected multi-graph.)

In Section 3.1 we present a construction of a directed edge-colored $O(1)$-regular multi-graph that is $\Omega(1)$-robustly self-ordered. We shall actually present a sufficient condition and a specific instantiation that satisfies it. In Section 3.2 we show how to transform any directed edge-colored multi-graph into an undirected one while preserving all relevant features; that is, bounded robustness, bounded degree, regularity, expansion, and local computability.

3.1 A sufficient condition for robust self-ordering of directed colored graphs

For any d permutations, $\pi_1, ..., \pi_d : [n] \to [n]$, we consider two multi-graphs.

1. The primary multi-graph (of $\pi_1, ..., \pi_d$) is a *directed* multi-graph, denoted $G_1 = ([n], E_1)$, such that $E_1 = \{(v, \pi_i(v)) : v \in [n] \,\&\, i \in [d]\}$. This directed multi-graph is coupled with an edge-coloring in which the directed edge from v to $\pi_i(v)$ is colored i.
2. The secondary multi-graph (of $\pi_1, ..., \pi_d$) is an undirected multi-graph, denoted $G_2 = (V_2, E_2)$, such that $V_2 = \{(u, v) \in [n]^2 : u \neq v\}$ and $E_2 = \{\{(u, v), (\pi_i(u), \pi_i(v))\} : (u, v) \in V_2 \,\&\, i \in [d]\}$.

We note that each of these multi-graphs is a *Schreier graph* that correspond to the action of the permutation $\pi_1, ..., \pi_d$ on the corresponding vertex sets (i.e., $[n]$ and V_2, respectively). For a wider perspective see the (paragraph at the) end of this subsection.

We now state the main result of this section, which asserts that the primary multi-graph G_1 is robustly self-ordered if the secondary multi-graph G_2 is an expander. We use the combinatorial definition of expansion: *A multi-graph $G = (V, E)$ is* γ-expanding *if, for every subset S of size at most $|V|/2$, there are at least $\gamma \cdot |S|$ vertices in $V \setminus S$ that neighbor some vertex in S.*

▶ **Theorem 3.2** (expansion of G_2 implies robust self-ordering of G_1). *For any $d \geq 2$ permutations, $\pi_1, ..., \pi_d : [n] \to [n]$, if the secondary multi-graph G_2 of $\pi_1, ..., \pi_d$ is γ-expanding, then the primary directed multi-graph G_1 of $\pi_1, ..., \pi_d$ coupled with the foregoing edge-coloring is γ-robustly self-ordered. Furthermore, G_1* (or rather the undirected multi-graph underlying G_1) *is* $\min(0.25, \gamma/3)$*-expanding.*

Proof. Let $\mu : [n] \to [n]$ be an arbitrary permutation, and let $T = \{v \in [n] : \mu(v) \neq v\}$ be its set of non-fixed-points. Then, the size of the symmetric difference between G_1 and $\mu(G_1)$ equals $2 \cdot \sum_{i \in [d]} |D_i|$ such that $v \in D_i$ if $(\mu(v), \mu(\pi_i(v)))$ is either not an edge in G_1 or is not colored i in it, whereas $(v, \pi_i(v))$ is an edge colored i in G_1. Note that if $(\mu(v), \mu(\pi_i(v)))$ is not an i-colored edge in G_1, then $\pi_i(\mu(v)) \neq \mu(\pi_i(v))$. Hence, $D_i = \{v \in [n] : \mu(\pi_i(v)) \neq \pi_i(\mu(v))\}$.

The key observation (proved next) is that *if $v \in T \setminus D_i$, then $(\pi_i(v), \pi_i(\mu(v)) \in T_2$, where $T_2 = \{(v, \mu(v)) : v \in T\}$ represents the sets of replacements performed by μ.* This fact implies that if $\sum_{i \in [d]} |D_i|$ is small in comparison to $|T|$, then the set T_2 (which is a set of vertices in G_2) does not expand much, in contradiction to the hypothesis. Details follow.

▶ **Observation 3.2.1** (key observation). *For T, D_i and T_2 as defined above, if $v \in T \setminus D_i$, then $(\pi_i(v), \pi_i(\mu(v)) \in T_2$.*

Recall that $v \in T$ implies $(v, \mu(v)) \in T_2$. Observation 3.2.1 asserts that if (in addition to $v \in T$) it holds that $v \notin D_i$, then $(\pi_i(v), \pi_i(\mu(v))$ is also in T_2. This means that the edges colored i incident at $\{(\pi_i(v), \pi_i(\mu(v))) : v \in T \setminus D_i\}$ do not contribute to the expansion of the set T_2 in G_2.

Proof. Since $v \notin D_i$ we have $\pi_i(\mu(v)) = \mu(\pi_i(v))$, and $\mu(\pi_i(v)) \neq \pi_i(v)$ follows, because otherwise $\pi_i(\mu(v)) = \pi_i(v)$, which implies $\mu(v) = v$ in contradiction to $v \in T$. However, $\mu(\pi_i(v)) \neq \pi_i(v)$ means that $\pi_i(v) \in T$, and $(\pi_i(v), \pi_i(\mu(v))) = (\pi_i(v), \mu(\pi_i(v))) \in T_2$ follows. ◁

Conclusion. Recall that Observation 3.2.1 implies that $\{(\pi_i(v), \pi_i(\mu(v))) : v \in T \setminus D_i\} \subseteq T_2$, whereas $\bigcup_{i\in[d]}\{(\pi_i(v), \pi_i(\mu(v))) : v \in T\}$ is the neighborhood of T_2 in the multi-graph G_2 (since $\{(\pi_i(v), \pi_i(\mu(v))) : i \in [d]\}$ the neighbor-set of $(v, \mu(v))$ in G_2). Using the γ-expansion of G_2 (and $|T_2| \leq n < |V_2|/2$), it follows that $\sum_{i\in[d]} |D_i| \geq \gamma \cdot |T|$. The main claim follows.

The expansion of G_1 is shown by relating sets of vertices of G_1 to the corresponding sets of pairs in G_2. Specifically, for and $S \subset [n]$ of size at most $n/2$, we consider the set $T = \{(u, v) \in V_2 : u, v \in S\}$, which has size $|S| \cdot (|S| - 1) \leq \frac{n}{2} \cdot (\frac{n}{2} - 1) < \frac{|V_2|}{2}$. Letting T' denote the set of neighbors of T in G_2, and $|S'|$ denote the set of neighbors of S in G_1, we have $|T' \setminus T| \geq \gamma \cdot |T|$, on the one hand (by expansion of G_2), and $|T' \setminus T| \leq 2 \cdot |S| \cdot |S' \setminus S| + |S' \setminus S| \cdot (|S' \setminus S| - 1)$ on the other hand. This implies $|S' \setminus S| \geq (\gamma/3) \cdot |S|$ (unless $|S| < 5$, which can be handled by using $|S' \setminus S| \geq 1$). ◀

Primary and secondary multi-graphs based on $\mathrm{SL}_2(p)$

Recall that $\mathrm{SL}_2(p)$ is the group of 2-by-2 matrices over $\mathrm{GF}(p)$ that have determinant 1. There are several different explicit constructions of constant-size expanding generating sets for $\mathrm{SL}_2(p)$, namely making the associated Cayley graph an expander (see, e.g., [28], [27, Thm. 4.4.2(i)], and [6]). We use any such generating set to define a directed (edge-colored) multi-graph G_1 on $p + 1$ vertices, and show that the associated multi-graph on pairs, G_2, is an expander.

▶ **Proposition 3.3** (expanding generators for $\mathrm{SL}_2(p)$ yield an expanding secondary multi-graph). *For any prime $p > 2$, let $V = \{(1, i)^\top : i \in \mathrm{GF}(p)\} \cup \{(0, 1)^\top\}$, and $M_1, ..., M_d \in \mathrm{SL}_2(p)$. For every $i \in [d]$, define $\pi_i : V \to V$ such that $\pi_i(u) = v$ if $v \in V$ is a non-zero multiple of $M_i u$. Then:*

1. *Each π_i is a bijection.*
2. *If the Cayley multi-graph $\mathcal{C} = \mathcal{C}(\mathrm{SL}_2(p), \{M_1, ..., M_d\}) = (\mathrm{SL}_2(p), \{\{M, M_iM\} : M \in \mathrm{SL}_2(p) \,\&\, i \in [d]\})$ is an expander, then the* (Schreier) *multi-graph G_2 with vertex-set $P = \{(v, v') : v \in V \,\&\, v' \in V \setminus \{v\}\}$ and edge-set $\{\{(v, v'), (\pi_i(v), \pi_i(v'))\} : (v, v') \in P\}$ is an expander.*

Part 1 implies that these permutations yield a primary directed edge-colored multi-graph on the vertex-set V, whereas Part 2 asserts that the corresponding secondary graph is an expander (if the corresponding Cayley graph is expanding). Note that $|V| = p + 1$ and $|P| = (p + 1)p$, whereas $|\mathrm{SL}_2(p)| = p^3 - p = (p - 1) \cdot |P|$.

Proof. Part 1 follows by observing that for every $M \in \mathrm{SL}_2(p)$ and every vector $v \in \mathrm{GF}(p)^2$ and scalar $\alpha \in \mathrm{GF}(p)$ it holds that $M\alpha v = \alpha M v$. Consequently, if for some non-zero $\alpha, \alpha' \in \mathrm{GF}(p)$ it holds that $\alpha M v = \alpha' M v'$, then $Mv = M\alpha'' v'$ for $\alpha'' = \alpha'/\alpha$, which implies $v = \alpha'' v'$ (since M is invertible). (Hence, $\pi_i(v) = \pi_i(v')$, for $v, v' \in V$, implies $v = v'$.)

Part 2 follows by observing that the vertices of G_2 correspond to equivalence classes of the vertices of $\mathcal{C}$ *that are preserved by* $\mathrm{SL}_2(p)$, where $A, B \in \mathrm{SL}_2(p)$ are equivalent if the columns of A are non-zero multiples of the corresponding columns of B. That is, we consider an equivalence relation, denoted $\equiv$, such that for $A = [A_1|A_2]$ and $B = [B_1|B_2]$ in $\mathrm{SL}_2(p)$ it holds that $A \equiv B$ if $A_i = \alpha_i B_i$ for both $i \in \{1, 2\}$, where $\alpha_1, \alpha_2 \in [p - 1]$ (and, in fact, $\alpha_2 = 1/\alpha_1$).[14] By saying that these *equivalence classes are preserved by* $\mathrm{SL}_2(p)$, we mean that, for every $A, B, M \in \mathrm{SL}_2(p)$, if $A \equiv B$, then $MA \equiv MB$. Hence, the (combinatorial)

[14] Recall that $\det(A) = 1 = \det(B)$, whereas $\det([\alpha_1 B_1|\alpha_2 B_2]) = \alpha_1\alpha_2 \cdot \det(B)$. Note that each equivalence class contains a single element of P.

expansion of G_2 follows from the expansion of $\mathcal{C}$, because the neighbors of a vertex-set $S \subseteq P$ in G_2 are the vertices of G_2 that are equivalent to T' such that T' is the set of vertices of $\texttt{CC}^{(t)}$ that neighbor (in $\texttt{CC}^{(t)}$) vertices that are equivalent to vertices in S.[15] ◀

A simple construction

Combining Theorem 3.2 with Proposition 3.3, while using a simple pair of expanding generators (which does not yield a Ramanujan graph), we get

▶ **Corollary 3.4** (a simple robustly self-ordered primary multi-graph). *For any prime $p > 2$, let $V = \{(1,i)^\top : i \in \mathrm{GF}(p)\} \cup \{(0,1)^\top\}$, and consider the matrices*

$$M_1 \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad M_2 \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{4}$$

Then, for π_1 and π_2 defined as in Proposition 3.3, the corresponding primary (directed edge-colored) *multi-graph is robustly self-ordered.*

This follows from the fact that the corresponding Cayley graph $\mathcal{C}(\mathrm{SL}_2(p), \{M_1, M_2\})$ is an expander [27, Thm. 4.4.2(i)].

Perspective

The foregoing construction using the group $\mathrm{SL}_2(p)$ is a special case of a much more general family of constructions, and the elements of the proof of Proposition 3.3 follow an established theory (explained, e.g., in [25, Sec. 11.1.2]), which we briefly describe.

Let H be any finite group, and S an expanding generating set of H (i.e., the Cayley graph $\mathcal{C}(H,S)$ is an expander). Assume that H acts on a finite set V (i.e., each $h \in H$ is associated with a permutation on V, and $h'h(v) = h'(h(v))$ for every $h, h' \in H$ and $v \in V$). Then, the primary (directed edge-colored) multi-graph G_1 on vertices V can be constructed from the permutations defined by members of S. The secondary multi-graph G_2 is naturally defined by the action of S on pairs of elements in V. Finally, the expansion of $\mathcal{C}(H,S)$ implies that every connected component of G_2 is an expander.[16] Thus, whenever this (Schreier) graph G_2 is connected (as it is in Proposition 3.3), one may conclude that G_1 is a directed edge-colored robustly self-ordered multi-graph.

3.2 From the directed variant to the undirected one

In this section we show how to transform directed (edge-colored) multi-graphs, of the type constructed in Section 3.1, into undirected ones, while preserving all relevant features (i.e., bounded robustness, bounded degree, regularity, expansion, and local computability). The transformation is extremely simple and natural: We replace the directed edge (u,v) colored j by a 2-path with a designated auxiliary vertex $a_{u,v,j}$, while coloring the edge $\{u, a_{u,v,j}\}$ by $2j-1$ and the edge $\{a_{u,v,j}, v\}$ by $2j$. Evidently, this colored 2-path encodes the direction of the original edge (as well as the original color).

[15] Specifically, let S have density at most half in P, and let T be the set of vertices of $\mathcal{C}$ that are equivalent to S. Note that $|T| = (p-1) \cdot |S|$, since each equivalence class contains a single element of P. By the foregoing, the set of neighbors of T in $\mathcal{C}$, denoted T', is a collection of equivalence classes of vertices of G_2, and $|T' \setminus T| = \Omega(|T|)$ by the expansion of $\mathcal{C}$. It follows that the set of neighbors of S in G_2, denoted S', is the set of vertices that are equivalent to T', which implies that $|S' \setminus S| = \frac{|T' \setminus T|}{p-1} = \Omega(|S|)$.

[16] Indeed, this was easy to demonstrate directly in the case of Proposition 3.3.

Note that the foregoing transformation works well provided that there are no parallel edges that are colored with the same color, a condition which is satisfied by the construction presented in Section 3.1. Furthermore, since the latter construction has no vertices of (in+out) degree less that $2d \geq 4$, there is no need to mark the original vertices by self-loops. Hence, a preliminary step akin to Construction 2.2 in unnecessary here, although it can be performed in general.

▶ **Proposition 3.5** (from directed robust self-ordering to undirected robust self-ordering)**.** *For constants $d \geq 3$ and c, let $G = (V, E)$ be a directed multi-graph in which each vertex has between three and d incident edges* (in both directions)*, and that G is coupled with an edge-coloring function $\chi : E \to [c]$ such that no parallel edges* (in same the direction) *are assigned the same color. Letting $E_i = \{e \in E : \chi(e) = i\}$ denote the set of edges colored i in G, consider the undirected multi-graph $G' = (V', E')$ such that $V' = V \cup \{a_{u,v,i} : (u, v) \in E_i\}$ and $E' = \bigcup_{j \in [2c]} E'_j$ where*

$$\begin{aligned} E'_{2i-1} &= \{\{u, a_{u,v,i}\} : (u, v) \in E_i\}, \\ E'_{2i} &= \{\{a_{u,v,i}, v\} : (u, v) \in E_i\}, \end{aligned}$$

and the edge-coloring function $\chi' : E' \to [2c]$ that assigns the edges of E'_j the color j (i.e., $\chi'(e) = j$ for every $e \in E'_j$)*. Then, if (G, χ) is γ-robustly self-ordered* (in the sense of Definition 3.1)*, then (G', χ') is $(\gamma/2)$-robustly self-ordered* (in the sense of Definition 2.1)*.*

We comment that the transformation of (G, χ) to (G', χ') preserves bounded robustness, bounded degree, regularity, expansion, and local computability (cf. Theorem 2.8).

Proof. The proof is analogous to the proof of Theorem 2.4, but it is much simpler because the gadgets used in the current transformation (i.e., the auxiliary vertices $a_{u,v,i}$) are much simpler.

Considering an arbitrary permutation $\mu' : V' \to V'$, we lower-bound the distance between G' and $\mu'(G')$ as a function of the number of non-fixed-points under μ'. We do so by considering the contribution of each non-fixed-point to the distance between G' and $\mu'(G')$. We first recall the fact that the vertices of V (resp., the auxiliary vertices) are uniquely identified in $\mu'(G')$ by virtue of the their degree, since each vertex of V has degree at least three (in G') whereas the auxiliary vertices have degree 2.

Case 1: *Auxiliary vertices of the form $a_{u,v,i}$ that are not mapped by μ' to auxiliary vertices of the form $a_{u',v',i}$; that is, $\mu'(a_{u,v,i}) \in (V \cup \bigcup_{j \neq i}\{a_{u',v',j} : (u', v') \in E\})$.*

Each such vertex $a_{u,v,i}$ contributes at least one unit to the difference between G' and $\mu'(G')$, since the two edges incident at $a_{u,v,i}$ (in G') are colored $2i - 1$ and $2i$ respectively, whereas $\mu(a_{u,v,i})$ has either more than two edges (in G') or its two edges are colored $2j - 1$ and $2j$, respectively, where for $j \neq i$. Hence, if the current case contains n_1 vertices, then their contribution to the distance between G' and $\mu'(G')$ is at least n_1.

Ditto for vertices of V that are mapped by μ' to an auxiliary vertex.

Case 2: *Vertices $v \in V$ such that $\mu'(v) \in V \setminus \{v\}$.*

By the hypothesis that the edge-colored directed G is robustly self-ordered, it follows that such vertices contribute proportionally to the difference between the colored versions of the directed multi-graphs G and $\mu(G)$, where μ is the restriction of μ' to V. Specifically, the number of tuples $((u, v), i)$ such that (u, v) is colored i in exactly one of these multi-graph (i.e., either in G or in $\mu(G)$ but not in both) is at least $\gamma \cdot |\{v \in V : \mu(v) \neq v\}|$. Assume, without loss of generality that $(u, v) \in E_i$ but either $(\mu^{-1}(u), \mu^{-1}(v)) \notin E$ or $(\mu^{-1}(u), \mu^{-1}(v)) \in E_j$ for $j \neq i$. Either way, it follows that a vertex not in $\{a_{u',v',i} :$

$(u',v') \in E_i\}$ is mapped by μ' to $a_{u,v,i}$, which means that Case 1 applies for each such a tuple. Hence, if the number of vertices in the current case is n_2, then $n_1 \geq \gamma \cdot n_2$, and we get a contribution of at least $\gamma \cdot n_2$ via Case 1.

Case 3: *Auxiliary vertices of the form $a_{u,v,i}$ that are mapped by μ' to auxiliary vertices of the form $a_{u',v',i}$ for $(u'v') \neq (u,v)$; that is, $\mu'(a_{u,v,i}) \in \{a_{u',v',i} : (u',v') \in E_i \setminus \{(u,v)\}\}$.* For u,v,u',v' and i as above, if $\mu'(u) = u'$ and $\mu'(v) = v'$, then an auxiliary vertex that connects u and v in G' is mapped to an auxiliary vertex that does not connects them in $\mu'(G')$ (but rather connects the vertices u' and v', whereas either $u' \neq u$ or $v' \neq v$). So we get a contribution of at least one unit to the difference between G' and $\mu'(G')$ (i.e., the edge incident at either u or v). Hence, the contribution is proportional to the number of non-fixed-points of the current type. Otherwise (i.e., $(\mu'(u), \mu'(v)) \neq (u',v')$), we get a vertex as in either Case 1 or Case 2, and get a proportional contribution again.

Hence, the contribution of each of these cases to the difference between G' and $\mu'(G')$ is proportional to the number of vertices involved. Specifically, if there are n_i vertices in Case i, then we get a contribution-count of at least $\gamma \cdot \sum_{i \in [3]} n_1$, where some of these contributions were possibly counted twice. The claim follows. ◀

4 The Three-Step Construction

In this section we present a different construction of bounded-degree graphs that are robustly self-ordered. It uses totally different techniques than the ones utilized in the construction presented in Section 3. Furthermore, the current construction offers the flexibility of obtaining either graphs that have small connected components (i.e., of logarithmic size) or graphs that are highly connected (i.e., are expanders). Actually, one can obtain anything in-between (i.e., n-vertex graphs that consist of $s(n)$-sized connected components that are each an expander, for any $s(n) = \Omega((\log n)/\log\log n)$). We mention that robustly self-ordered bounded-degree graphs with small connected components are used in the proof of Theorem 5.2.

As stated in Section 1.1.2, the current construction proceeds in three steps. First, in Section 4.1, we prove the existence of robustly self-ordered bounded-degree graphs, and observe that such ℓ-vertex graphs can actually be found in poly(ℓ!)-time [*sic*]. Next, setting $\ell = \Omega((\log n)/\log\log n)$, we use these graphs as part of 2ℓ-vertex connected components in an n-vertex (robustly self-ordered bounded-degree) graph that is constructed in poly(n)-time (see Section 4.2). Lastly, in Section 4.3, we repeat this strategy using the graphs constructed in Section 4.2, and obtain exponentially larger graphs that are locally constructible.

In addition, in Section 4.4, we show that the foregoing graphs can be locally self-ordered. That is, given a vertex v in any graph $G' = (V',E')$ that is isomorphic to the foregoing n-vertex graph and oracle access to the incidence function of G', we can find the vertex to which this unique isomorphism maps v in poly($\log n$)-time.

4.1 Existence

As stated above, we start with establishing the mere existence of bounded-degree graphs that are robustly self-ordered.

▶ **Theorem 4.1** (robustly self-ordered graphs exist). *For any sufficiently large constant d, there exists a family $\{G_n\}_{n\in\mathbb{N}}$ of robustly self-ordered d-regular graphs. Furthermore, these graphs are expanders.*

Actually, it turns out that random d-regular graphs are robustly self-ordered; see Theorem 6.1. Either way, given the existence of such n-vertex graphs, they can actually be found in poly($n!$)-time, by an exhaustive search. Specifically, for each of the possible $n^{dn/2}$ graphs, we check the robust self-ordering condition by checking all $n!-1$ relevant permutation. (The expansion condition can be checked similarly, by trying all $(0.5+o(1))\cdot 2^n$ relevant subsets of $[n]$.)

The proof of Theorem 4.1 utilizes a simpler probabilistic argument than the one used in the proof of Theorem 6.1. This argument (captured by Claim 4.1.1) refers to the auxiliary model of edge-colored multi-graphs (see Definition 2.1) and is combined with a transformation of this model to the original model of uncolored graphs (provided in Construction 2.3 and analyzed in Theorem 2.4). Indeed, the relative simplicity of Claim 4.1.1 is mainly due to using the edge-colored model (see digest at the end of Section 6).

Proof. To facilitate the proof, we present the construction while referring to the edge-colored model presented in Section 2. We shall then apply Theorem 2.4 and obtain a result for the original model (of uncolored simple graphs).

For $m=n/O(1)$, we shall consider $2m$-vertex multi-graphs that consists of two m-vertex cycles, using a different color for the edges of each cycle, that are connected by $d'=O(1)$ random perfect matching, which are also each assigned a different color. (Hence, we use $2+d'$ colors in total.) We shall show that (w.h.p.) a random multi-graph constructed in this way is robustly self-ordered (in the colored sense). (Note that parallel edges, if they exist, will be assigned different colors.) Specifically, we consider a generic $2m$-vertex multi-graph that is determined by d' perfect matchings of $[m]$ with $\{m+1,...,2m\}$. Denoting this sequence of perfect matchings by $\overline{M}=(M_1,...,M_{d'})$, we consider the (edge-colored) multi-graph $G_{\overline{M}}([2m],E_{\overline{M}})$ given by

$$\begin{aligned} E_{\overline{M}} &= C_1 \cup C_2 \cup \bigcup_{j\in[d']} M_j \\ &\quad \text{where } C_1=\{\{i,i+1\}:i\in[m-1]\}\ \cup\ \{\{m,1\}\} \\ &\quad \text{and } C_2=\{\{m+i,m+i+1\}:i\in[m-1]\}\ \cup\ \{\{2m,m+1\}\} \end{aligned}$$

and a coloring χ in which the edges of C_j are colored j and the edges of M_j are colored $j+2$. (That is, for $i\in\{1,2\}$, the set C_i forms a cycle of the form $((i-1)m+1,(i-1)m+2,...,(i-1)m+m,(i-1)m+1)$ and its edges are colored i.) Note that the $d'+1$ edges incident at each vertex are assigned $d'+1$ different colors.

▷ Claim 4.1.1 (w.h.p., $G_{\overline{M}}$ is robustly self-ordered). For some constant $\gamma>0$, with high probability over the choice of $\overline{M}$, the edge-colored multi-graph $G_{\overline{M}}$ is γ-robustly self-ordered. Furthermore, it is also an expander.

Proof. Consider an arbitrary permutation $\mu:[2m]\to[2m]$, and let $t=|\{i\in[2m]:\mu(i)\neq i\}|$. We shall show that, with probability $1-\exp(-\Omega(dt\log m))$ over the choice of $\overline{M}$, the difference between the colored versions of $G_{\overline{M}}$ and $\mu(G_{\overline{M}})$ is $\Omega(t)$. Towards this end, we consider two cases.

Case 1: $|\{i\in[m]:\mu(i)\notin[m]\}|>t/4$. Equivalently, $|\{i\in[2m]:\lceil\mu(i)/m\rceil\neq\lceil i/m\rceil\}|>t/2$. The vertices in the set $\{i\in[m]:\mu(i)\notin[m]\}$ are mapped from the first cycle to the second cycle, and so rather than having two incident edges that are colored 1 they have two incident edges colored 2. Hence, each such vertex contributes two units to the difference (between the colored versions of $G_{\overline{M}}$ and $\mu(G_{\overline{M}})$), and the total contribution is greater than $2\cdot(t/4)\cdot 2$, where the first factor of 2 accounts also for vertices that are mapped from C_2 to C_1.

Case 2: $|\{i\in[m]:\mu(i)\not\in[m]\}|\leq t/4$. Equivalently, $|\{i\in[2m]:\lceil\mu(i)/m\rceil\neq\lceil i/m\rceil\}|\leq t/2$. We focus on the non-fixed-points of μ that stay on their original cycle (i.e., those not considered in Case 1). Let $A\stackrel{\rm def}{=}\{i\in[m]:\mu(i)\neq i\wedge\mu(i)\in[m]\}$ and $B\stackrel{\rm def}{=}\{i\in\{m+1,....,2m\}:\mu(i)\neq i\wedge\mu(i)\in\{m+1,...,2m\}\}$. By the case hypothesis, $|A|+|B|\geq t/2$, and we may assume (without loss of generality) that $|A|\geq t/4$. As a warm-up, we first show that *each element of A contributes a non-zero number of units to the difference* (between the colored versions of $G_{\overline{M}}$ and $\mu(G_{\overline{M}})$) *with probability* $1-O(1/m)^{d'}$, over the choice of $\overline{M}$.

To see this, let $\pi_j:[m]\to\{m+1,...,2m\}$ be the mapping used in the $j^{\rm th}$ matching; that is, $M_j=\{\{i,\pi_j(i)\}:i\in[m]\}$, which means that $\pi_j(i)$ is the $j^{\rm th}$ match of i in $G_{\overline{M}}$ (i.e., the vertex matched to i by M_j). Then, we consider the event that *for some $j\in[d']$, the $j^{\rm th}$ match of $i\in[m]$ in $\mu(G_{\overline{M}})$ is different from the $j^{\rm th}$ match of i in $G_{\overline{M}}$*, and note that when this event occurs i contributes to the difference (between the colored versions of $G_{\overline{M}}$ and $\mu(G_{\overline{M}})$). Note that x is the $j^{\rm th}$ match of i in $\mu(G_{\overline{M}})$ if and only if $\mu^{-1}(x)$ is the $j^{\rm th}$ match of $\mu^{-1}(i)$ in $G_{\overline{M}}$, which holds if and only if $\mu^{-1}(x)=\pi_j(\mu^{-1}(i))$ (equiv., $x=\mu(\pi_j(\mu^{-1}(i)))$). Hence, $i\in[m]$ contributes to the difference if and only if for some j it holds that $\pi_j(i)\neq\mu(\pi_j(\mu^{-1}(i)))$, because $\pi_j(i)\neq\mu(\pi_j(\mu^{-1}(i)))$ means that the edge $\{i,\pi_j(i)\}$ is colored $j+2$ in $G_{\overline{M}}$ but is not colored $j+2$ in $\mu(G_{\overline{M}})$ (since a different edge incident at i in $\mu(G_{\overline{M}})$ is colored $j+2$). Letting $\overline{\pi}=(\pi_1,...,\pi_{d'})$, the probability of the complementary event (i.e., i does not contribute to the difference) is given by

$$\begin{aligned}\Pr_{\overline{\pi}}\left[(\forall j\in[d'])\ \ \pi_j(i)=\mu(\pi_j(\mu^{-1}(i)))\right] &= \prod_{j\in[d']}\Pr_{\pi_j}\left[\pi_j(i)=\mu(\pi_j(\mu^{-1}(i)))\right]\\ &\leq (m-1)^{-d'},\end{aligned}$$

where the inequality uses the hypothesis that $\mu(i)\neq i$ and $i,\mu(i)\in[m]$; specifically, fixing the value of $\pi_j(\mu^{-1}(i))$, leaves $\pi_j(i)$ uniformly distributed in $S\stackrel{\rm def}{=}\{m+1,...,2m\}\setminus\{\pi_j(\mu^{-1}(i))\}$, which means that $\Pr_{\pi_j}[\pi_j(i)=\mu(v)|v=\pi_j(\mu^{-1}(i))]\leq 1/|S|$ (where equality holds if $\mu(v)\in S$).

The same argument generalises to any set $I\subseteq A$ such that $I\cap\mu(I)=\emptyset$. In such a case, letting $I=\{i_1,...,i_{t'}\}$, we get

$$\begin{aligned}&\Pr_{\overline{\pi}}\left[(\forall i\in I)(\forall j\in[d'])\ \ \pi_j(i)=\mu(\pi_j(\mu^{-1}(i)))\right]\\ &=\prod_{k\in[t']}\prod_{j\in[d']}\Pr_{\pi_j}\left[\pi_j(i_k)=\mu(\pi_j(\mu^{-1}(i_k)))\,\middle|\,(\forall k'\in[k-1])\ \pi_j(i_{k'})=\mu(\pi_j(\mu^{-1}(i_{k'})))\right]\\ &\leq(m-2t'+1)^{-t'd'},\end{aligned}$$

where the inequality uses the hypothesis that $I\cap\mu(I)=\emptyset$; specifically, for each $k\in[t']$, we use the fact that $i_k\not\in\{i_1,...,i_{k-1},\mu^{-1}(i_1),...,\mu^{-1}(i_k)\}$. Hence, fixing the values of $\pi_j(i_{k'})$ for all $k'\in[k-1]$ and the values of $\pi_j(\mu^{-1}(i_{k'}))$ for all $k'\in[k]$, and denoting these values by $u_1,...,u_{k-1}$ and $v_1,...,v_k$ respectively, leaves $\pi_j(i_k)$ uniformly distributed in $S\stackrel{\rm def}{=}\{m+1,...,2m\}\setminus\{u_1,...,u_{k-1},v_1,...,v_k\}$, which means that $\Pr_{\pi_j}[\pi_j(i)=\mu(v_k)|\text{foreging fixing}]\leq 1/|S|$ (where equality holds if $\mu(v_k)\in S$).

Recalling that $|A|\geq t/4$ and $t\leq 2m$, we upper-bound the probability (over the choice of $\overline{M}$) that A contains a $t/8$-subset A' such that $(\forall i\in A')(\forall j\in[d'])\ \pi_j(i)=\mu(\pi_j(\mu^{-1}(i)))$, by taking a union bound over all possible A' and using for each such A' a subset $I\subset A'$ such that $I\cap\mu(I)=\emptyset$. (So we actually take a union bound over the I's and derive a conclusion regarding the $t/8$-subsets A'.) Observing that $|I|\geq|A'|/2\geq t/16$, we conclude that, with probability at most $\binom{t}{t/16}\cdot(m/2)^{d'\cdot t/16}=\exp(-\Omega(d't\log m))$ over the choice of $\overline{M}$, the set A contains no $t/8$-subset A' as above. This means that, with probability at most $\exp(-\Omega(d't\log m))$, less than $t/8$ of the indices $i\in A$ contribute a non-zero number of units to the difference (between the colored versions of $G_{\overline{M}}$ and $\mu(G_{\overline{M}})$).

Hence, we have shown that, for every permutations $\mu : [2m] \to [2m]$, the probability (over the choice of $\overline{M}$) that the size of the symmetric difference between the colored versions of $G_{\overline{M}}$ and $\mu(G_{\overline{M}})$ is smaller than $t/8$ is $\exp(-\Omega(d't\log m))$, where t is the number of non-fixed-points of μ. Letting $\gamma = 1/8$ and taking a union bound over all (non-trivial) permutations $\mu : [2m] \to [2m]$, we conclude that the probability, over the choice of $\overline{M}$, that $G_{\overline{M}}$ is not γ-robustly self-ordered is at most

$$\begin{aligned}\sum_{t\in[2m]} \binom{2m}{t} \cdot \exp(-\Omega(d't\log m)) &= \sum_{t\in[2m]} \exp(-\Omega((d'-O(1))\cdot t\log m)) \\ &= \exp(-\Omega((d'-O(1))\cdot \log m)),\end{aligned}$$

and the claim follows (for any sufficiently large d'), while observing that, with very high probability, these multi-graphs are expanders. ◁

Back to the non-colored version. We now convert the edge-colored multi-graphs $G = G_{\overline{M}}$ that are γ-robustly self-ordered into standard graphs G' that are robustly self-ordered in the original sense. This is done by using Construction 2.3 (while relying on Theorem 2.4). Recall that this transformation also preserves expansion. Actually, before invoking Construction 2.3, we augment the multi-graph G by adding a self-loop to each vertex, and color all these self-loops using a special color. Combining Claim 4.1.1 and Theorem 2.4, the current theorem follows. ◀

4.2 Constructions

Having established the existence of bounded-degree graphs that are robustly self-ordered, we now turn to actually construct them. We shall use the fact that the proof of existence yields a construction that runs in time that is polynomial in the number of possible graphs. Specifically, for $\ell = \frac{O(\log n)}{\log\log n}$, we shall construct ℓ-vertex graphs in $\text{poly}(\ell^\ell)$-time and use them in our construction of n-vertex graphs, while noting that $\text{poly}(\ell^\ell) = \text{poly}(n)$.

▶ **Theorem 4.2** (constructing robustly self-ordered graphs). *For any sufficiently large constant d, there exists an efficiently constructable family $\{G_n\}_{n\in\mathbb{N}}$ of robustly self-ordered graphs of maximum degree d. That is, there exists a polynomial-time algorithm that on input 1^n outputs the n-vertex graph $G_n = ([n], E_n)$. Furthermore, G_n consists of connected components of size $\frac{O(\log n)}{\log\log n} = o(\log n)$.*

Note that the connected components of G_n cannot be any smaller (than $\frac{O(\log n)}{\log\log n}$). This is the case because an asymmetric n-vertex bounded-degree graph, let alone a robustly self-ordered one, cannot have connected components of size $\frac{o(\log n)}{\log\log n}$ (because the number of t-vertex graphs of bounded-degree is $t^{O(t)}$).

Proof. The proof proceeds in two steps. We first use the existence of ℓ-vertex (d'-regular) expander graphs that are robustly self-ordered towards constructing a sequence of $m = \exp(\Omega(\ell\log\ell))$ bounded-degree 2ℓ-vertex graphs that are robustly self-ordered, expanding, and far from being isomorphic to one another. We construct this sequence of 2ℓ-vertex graphs in $\text{poly}(m)$-time, using the fact that $(\ell!)^{O(1)} = \text{poly}(m)$. In the second step, we show that the

$(m \cdot 2\ell)$-vertex graph that consists of these 2ℓ-vertex graphs (as its connected components) is robustly self-ordered. Note that this graph is constructed in time that is polynomial in its size, since its size is $\Omega(m)$, whereas it is constructed in $\mathrm{poly}(m)$-time.[17]

Given a generic n, let $\ell = \frac{O(\log n)}{\log \log n}$, which implies that $\ell^\ell = \mathrm{poly}(n)$. By Theorem 4.1, for all sufficiently large d', there exist ℓ-vertex d'-regular expander graphs that are robustly self-ordered (with respect to the robustness parameter c'). Furthermore, we can find such a graph, denoted G'_ℓ, in time $\mathrm{poly}(\ell^\ell) = \mathrm{poly}(n)$, by scanning all ℓ-vertex d'-regular graphs and checking both the expansion and the robustness (w.r.t parameter c') conditions for each of them. Actually, for $d'' = d' + 1$, we shall also find an ℓ-vertex d''-regular expander, denoted G''_ℓ, that is robustly self-ordered.

The construction of G_n. Using G'_ℓ and G''_ℓ, we construct an n-vertex robustly self-ordered graph, denoted G_n, that consists of $n/2\ell$ connected components that are pairwise far from being isomorphic to one another. This is done by picking $m = n/2\ell$ permutations, denoted $\pi_1, ..., \pi_m : [\ell] \to [\ell]$, that are pairwise far-apart and constructing 2ℓ-vertex graphs such that the i^{th} such graph consist of a copy of G'_ℓ and a copy of G''_ℓ that are connected by a matching as determined by the permutation π_i. Specifically, for $G'_\ell = ([\ell], E'_\ell)$ and $G''_\ell = ([\ell], E''_\ell)$, the i^{th} connected component is isomorphic to a graph with the vertex set $[2\ell]$ and the edge set

$$E'_\ell \ \cup\ \{\{\ell+u, \ell+v\} : \{u,v\} \in E''_\ell\} \ \cup\ \{\{v, \ell+\pi_i(v)\} : v \in [\ell]\}. \tag{5}$$

(The first two sets correspond to the copies of G'_ℓ and G''_ℓ, and the third set corresponds to the matching between these copies. Note that the vertices in $[\ell]$ have degree $d'+1$, whereas vertices in $\{\ell+1, ..., 2\ell\}$ have degree $d''+1 \neq d'+1$.)

To see that this construction can be carried out in $\mathrm{poly}(n)$-time, we need to show that the sequence of m pairwise far-apart permutations can be determined in $\mathrm{poly}(n)$-time, let alone that such a sequence exists. This is the case, because we can pick the permutation sequentially (one after the other) by scanning the symmetric group on $[\ell]$ and relying on the fact that for ($i < n$ and) any fixed sequence of permutations $\pi_1, ..., \pi_{i-1} : [\ell] \to [\ell]$ it holds that a random permutation π_i is far-apart from each of the fixed $i-1$ permutations; that is, $\mathrm{Pr}_{\pi_i}[|\{v \in [\ell] : \pi_i(v) \neq \pi_j(v)\}| = \Omega(\ell)] = 1 - o(1/n)$ for every $j \in [i-1]$.[18]

Towards proving that G_n is robustly self-ordered. We now prove that the resulting graph G_n, which consists of these m connected components, is c-robustly self-ordered, where c is a universal constant (which is independent of the generic n). For starters, let's verify that G_n is self-ordered. We first note that any automorphism of G_n must map the verifices of copies of G'_ℓ (resp., G''_ℓ) to vertices of copies of G'_ℓ (resp., G''_ℓ), since these are the only vertices of degree $d'+1$. The connectivity of these copies implies that the automorophism must map each connected component to some connected component, which determines the m connected

[17] We mention that a slightly different construction can be based on the fact that random ℓ-vertex (d'-regular) graphs are robustly self-ordered expanders (see Theorem 6.1). In this alternative construction we find a sequence of m such graphs that are pairwise far from being isomorphic to one another. As further detailed in Remark 6.2, the analysis of the alternative construction is somewhat easier than the analysis of the construction presented below, but we need the current construction for the proof of Theorem 4.5.

[18] Specifically, for some $\ell' = \Omega(\ell)$, we upper-bound $\mathrm{Pr}_\pi[|\{v \in [\ell] : \pi(v) = v)\}| \geq \ell - \ell']$, where $\pi : [\ell] \to [\ell]$ is a random permutation. We do so by observing that the number of permutations that have at least $\ell - \ell'$ fixed-points is at most $\binom{\ell}{\ell'} \cdot (\ell'!) = \frac{\ell!}{(\ell-\ell')!}$, whereas $(\ell-\ell')! = \exp(\Omega(\ell \log \ell)) = \omega(n)$ for any ℓ' such that $\ell - \ell' = \Omega(\ell)$.

components. The self-ordered feature of G'_ℓ and G''_ℓ determines a unique ordering on each copy, whereas the fact the permutations (i.e., π_i's) are different imposes that each connected component is mapped to itself (i.e., the order of the connected components is preserved). Hence, the automorphism must be trivial (and it follows that G_n is self-ordered).

An analogous argument establishes the robust self-ordering of G_n, where we use the hypothesis that G'_ℓ and G''_ℓ are expanders (rather than merely connected), the choice of the π_i's as being far-apart (rather than merely different), and the robust self-ordering of G'_ℓ and G''_ℓ (rather than their mere self-ordering) in order to establish the robust self-ordering of G_n. Considering an arbitrary permutation $\mu : [n] \to [n]$, these stronger features are used to establish a lower bound on the size of the symmetric difference between G_n and $\mu(G_n)$ as follows:

- The fact that G'_ℓ is an expander implies that if μ splits the vertices of a copy of G'_ℓ such that ℓ' vertices are mapped to copies that are different than the other $\ell - \ell' \geq \ell'$ vertices, then this contributes $\Omega(\ell')$ units to the difference between G_n and $\mu(G_n)$. Ditto for G''_ℓ, whereas mapping a copy of G'_ℓ to a copy of G''_ℓ contributes $\Omega(\ell)$ units (per the difference in the degrees).
- The robust self-ordering of G'_ℓ and G''_ℓ implies that if μ changes the index of vertices inside a component, then this yields a proportional difference between G_n and $\mu(G_n)$.
- The distance between the π_i's (along with the aforementioned robustness) implies that if μ changes the indices of the connected components, then each such change contributes $\Omega(\ell)$ units to the difference between G_n and $\mu(G_n)$.

The actual implementation of this sketch requires a careful accounting of the various contributions. As a first step in this direction we provide a more explicit description of G_n. We denote the set of vertices of the copy of G'_ℓ (resp., G''_ℓ) in the i^{th} connected component of G_n by $F_i = \{2(i-1)\ell + j : j \in [\ell]\}$ (resp., $S_i = \{2(i-1)\ell + \ell + j : j \in [\ell]\}$). Recall that F_i and S_i are connected by the edge-set

$$\{\{2(i-1)\ell + j, 2(i-1)\ell + \ell + \pi_i(j)\} : j \in [\ell]\} \tag{6}$$

whereas the subgraph of G_n induced by F_i (resp., S_i) has the edge-set $\{\{2(i-1)\ell + u, 2(i-1) + v\} : \{u,v\} \in E'_\ell\}$ (resp., $\{\{2(i-1)\ell + \ell + u, 2(i-1) + \ell + v\} : \{u,v\} \in E''_\ell\}$). In addition, let $F = \bigcup_{i\in[m]} F_i$ (resp., $S = \bigcup_{i\in[m]} S_i$).

The actual proof (that G_n is robustly self-ordered). Considering an arbitrary permutation $\mu : [n] \to [n]$, we lower-bound the distance (i.e., size of the symmetric difference) between G_n and $\mu(G_n)$ as a function of the number of non-fixed-points under μ (i.e., the number of $v \in [n]$ such that $\mu(v) \neq v$). We do so by considering the (average) contribution of every non-fixed-point to the distance between G_n and $\mu(G_n)$ (i.e., number of pairs of vertices that form an edge in one graph but not in the other). We may include the same contribution in few of the following (seven) cases, but this only means that we are double-counting the contribution by a constant factor.

Case 1: *Vertices $v \in F$ such that $\mu^{-1}(v) \in S$.* Ditto for $v \in S$ such that $\mu^{-1}(v) \in F$.
Each such vertex contributes at least one unit to the distance (between G_n and $\mu(G)$) by virtue of v having degree $d' + 1$ in G_n and strictly higher degree in $\mu(G_n)$, since vertices in F have degree $d' + 1$ (in G_n) whereas vertices in S have higher degree (in G_n).[19]

[19] Note that v neighbors u in $\mu(G_n)$ if and only if $\mu^{-1}(v)$ neighbors $\mu^{-1}(u)$ in G_n.

In light of Case 1, we may focus on vertices whose "type" is preserved by μ^{-1}. Actually, it will be more convenient to consider the set of vertices whose "type" is preserved by μ; that is, the set $\{v\in F:\mu(v)\in F\}\cup\{v\in S:\mu(v)\in S\}$. Next, for each $i\in[m]$, we define $\mu'(i)$ to be the index of the connected component that takes the plurality of $\mu(F_i)$; that is, $\mu'(i)\stackrel{\rm def}{=}j$ if $|\{v\in F_i:\mu(v)\in F_j\}|\geq|\{v\in F_i:\mu(v)\in F_k\}|$ for all $k\in[m]$ (breaking ties arbitrarily).

Case 2: *Vertices $v\in F_i$ such that $\mu(v)\in F\setminus F_{\mu'(i)}$.*

For starters, suppose that $|\{v\in F_i:\mu(v)\in F_{\mu'(i)}\}|\geq\ell/2$; that is, a majority of the vertices of F_i are mapped by μ to $F_{\mu'(i)}$. In this case, by the expansion of G'_ℓ, we get a contribution that is proportional to the size of the set $F'_i\stackrel{\rm def}{=}\{v\in F_i:\mu(v)\notin F_{\mu'(i)}\}$, because there are $\Omega(|F'_i|)$ edges betwen F'_i and the rest of F_i but there are no edges between F'_i and $F_i\setminus F'_i$ in $\mu(G_n)$. In the general case, we have to be more careful since expansion is guaranteed only for sets that have size at most $\ell/2$. In such a case we use an adequate subset of F'_i. Details follow.

Let $J\subseteq[m]\setminus\{\mu'(i)\}$ be maximal such that $\sum_{j\in J}|\{v\in F_i:\mu(v)\in F_j\}|\leq\ell/2$, and note that $F'_i\stackrel{\rm def}{=}\bigcup_{j\in J}\{v\in F_i:\mu(v)\in F_j\}$ occupies at least one third of $\{v\in F_i:\mu(v)\in F\setminus F_{\mu'(i)}\}$. Recall that the subgraph of G_n induced by F_i is an expander, and consider the edges in G_n that cross the cut between F'_i and the rest of F_i. Then, this cut has $\Omega(|F'_i|)$ edges in G_n, but there are no edges between F'_i and $F_i\setminus F'_i$ in $\mu(G_n)$, because $\mu^{-1}(F'_i)\subseteq\bigcup_{j\in J}F_j$ and $\mu^{-1}(F_i\setminus F'_i)\subseteq\bigcup_{j\in[m]\setminus J}F_j$ are not connected in G_n. Hence, the total contribution of the vertices in $\{v\in F_i:\mu(v)\in F\setminus F_{\mu'(i)}\}$ to the distance (between G_n and $\mu(G)$) is $\Omega(|F'_i|)$, which is proportional to their number (i.e., is $\Omega(|\{v\in F_i:\mu(v)\in F\setminus F_{\mu'(i)}\}|)$).

Defining $\mu''(i)$ in an analogous manner with respect to $\mu(S_i)$, we get an analogous contribution by the expander induced by S_i. Specifically, for each $i\in[m]$, we define $\mu''(i)$ to be the index of the connected component that takes the plurality of $\mu(S_i)$; that is, $\mu''(i)\stackrel{\rm def}{=}j$ if $|\{v\in S_i:\mu(v)\in S_j\}|\geq|\{v\in S_i:\mu(v)\in S_k\}|$ for all $k\in[m]$ (breaking ties arbitrarily).

Case 3: *Vertices $v\in S_i$ such that $\mu(v)\in S\setminus S_{\mu''(i)}$.*

Here we get a contribution of $\Omega(|\{v\in S_i:\mu(v)\in S\setminus S_{\mu''(i)}\}|)$, where the analysis is analogous to Case 2.

Recall that if $v\in F_i$ then it holds that $v=2(i-1)\ell+j$ for some $j\in[\ell]$, and that (in G_n) vertex v has a unique neighbor in S, which is $2(i-1)\ell+\ell+\pi_i(j)\in S_i$. It will be convinient to denote this neighbor by $\phi_i(v)$; that is, for $v\in F_i$ such that $v=2(i-1)\ell+j$, we have $\phi_i(v)=2(i-1)\ell+\ell+\pi_i(j)\in S_i$. The next two cases refer to vertices that are mapped by μ according to the plurality vote (e.g., $v\in F_i$ is mapped to $\mu(v)\in F_{\mu'(i)}$), but their match is not mapped accordingly (i.e., $\phi_i(v)\in S_i$ is not mapped to $S_{\mu'(i)}$).

Case 4: *Vertices $v\in F_i$ such that $\mu(v)\in F_{\mu'(i)}$ but $\mu(\phi_i(v))\notin S_{\mu'(i)}$.*

(Note that the condition $v\in F_i$ and $\mu(v)\in F_{\pi'(i)}$ means that vertex v is not covered in Case 2. If $\mu''(i)=\mu'(i)$, then $\mu(\phi_i(v))\notin S_{\mu'(i)}$ means that v is covered in Case 3, since $\phi_i(v)\in S_i$. Hence, the current case is of interest only when $\mu''(i)\neq\mu'(i)$. In particular, it is of interest when referring to vertices in the $i^{\rm th}$ connected component of G_n that reside in the copies of G'_ℓ and G''_ℓ and are mapped according to the plurality votes of these copies, whereas these two plurality votes are inconsistent.)

We focus on the case that a vast majority of the vertices in both F_i and S_i are mapped according to the plurality votes (i.e., $\mu'(i)$ and $\mu''(i)$), since the complementary cases are covered by Cases 2 and 3, respectively. Specifically, if either $|\{v\in F_i:\mu(v)\in[n]\setminus F_{\mu'(i)}\}|>\ell/3$ or $|\{u\in S_i:\mu(u)\in[n]\setminus S_{\mu''(i)}\}|>\ell/3$, then we get a contribution of $\Omega(\ell)$ either by Cases 1&2 or by Cases 1&3. Otherwise, it follows that

$$|\{v\in F_i:\mu(v)\in F_{\mu'(i)}\ \wedge\ \mu(\phi_i(v))\in S_{\mu''(i)}\}|\geq\ell-2\cdot\ell/3$$

which implies that, if $\mu'(i) \neq \mu''(i)$, then the i^{th} connected component of G_n contributes $\ell/3$ units to the difference (between G_n and $\mu(G_n)$), since v and $\phi_i(v)$ are connected in G_n, but $\mu(v) \in F_{\mu'(i)}$ and $\mu(\phi_i(v)) \in S_{\mu''(i)}$ reside in different connected components of $\mu(G_n)$. (That is, the contribution is due to vertices v of F_i that are mapped by μ to $F_{\mu'(i)}$, while the corresponding vertices $\phi_i(v)$ of S_i (which are connected to them in G_n) are mapped by μ to $S_{\mu''(i)} \subset S \setminus S_{\mu'(i)}$, whereas $F_{\mu'(i)}$ and $S_{\mu''(i)}$ are not connected in G_n, assuming $\mu'(i) \neq \mu''(i)$.)

To conclude: The contribution of the vertices of Case 4 (to the difference between G_n and $\mu(G_n)$) is proportional to the number of these vertices (where this contribution might have been counted already in Cases 1, 2 and 3).

Case 5: *Vertices $v \in F_i$ such that $\mu(v) \notin F_{\mu''(i)}$ but $\mu(\phi_i(v)) \in S_{\mu''(i)}$.*

(Equiv., vertices $v \in S_i$ such that $\mu(v) \in S_{\mu''(i)}$ but $\mu(\phi_i^{-1}(v)) \notin F_{\mu''(i)}$.)

Analogously to Case 4, the contribution of these vertices is proportional to their number. (Analogously, this augments Case 2 only in case $\mu''(i) \neq \mu'(i)$.)

In light of Cases 2–5, we may focus on indices $i \in [m]$ such that $\mu'(i) = \mu''(i)$ and on vertices in i^{th} connected component that are mapped by μ to the $\mu'(i)^{\text{th}}$ connected component (and the same "type" per Case 1). The following case refers to such vertices that do not maintain their position in this connected component.

Case 6: *Vertices $v = 2(i-1)\ell + j \in F_i$ such that $\mu(v) \in F_{\mu'(i)} \setminus \{2(\mu'(i)-1)\ell + j\}$.*

Ditto for $v = 2(i-1)\ell + \ell + j \in S_i$ such that $\mu(v) \in S_{\mu''(i)} \setminus \{2(\mu''(i)-1)\ell + \ell + j\}$.

(This case refers to vertices in F_i that are mapped to $F_{\mu'(i)}$ but do not maintain their index in the relevant copy of G'_ℓ; indeed, $v = 2(i-1)\ell + j$ is the j^{th} vertex of F_i, but it is mapped by μ to the k^{th} vertex of $F_{\mu'(i)}$ (i.e., $\mu(v) = 2(\mu'(i)-1)\ell + k$) such that $k \neq j$.)

Fixing i, let $C \overset{\text{def}}{=} \{v = 2(i-1)\ell + j \in F_i : \mu(v) \in F_{\mu'(i)} \setminus \{2(\mu'(i)-1)\ell + j\}\}$ denote the set of vertices considered in this case, and $D = \{v \in F_i : \mu(v) \notin F_{\mu'(i)}\}$ denote the set of vertices that we are going to discount for. As a warm-up, consider first the case that $D = \emptyset$. In this case, by the robust self-ordering of G'_ℓ, the contribution of the vertices in C to the difference between G_n and $\mu(G_n)$ is $\Omega(|C|)$.

In the general case (i.e., where D may not be empty), we get a contribution of $\Omega(|C|) - d' \cdot |D|$, where the second term compensates for the fact that the vertices of D were moved outside of this copy of G'_ℓ and replaced by different vertices that may have different incidences. Letting c' be the constant hidden in the Ω-notation, we get a contribution of at least $c' \cdot |D| - d' \cdot |D|$, which is at least $c' \cdot |C|/2$ if $|D| \leq c' \cdot |C|/2d'$. On the other hand, if $|D| > c' \cdot |C|/2d'$, then we get a contribution of $\Omega(|D|) = \Omega(|C|)$ by Cases 1–2. Hence, in both sub-cases we have a contribution of $\Omega(|C|)$ to the difference between G_n and $\mu(G_n)$.

The same analysis applies to $\{v = 2(i-1)\ell + \ell + j \in S_i : \mu(v) \in S_{\mu''(i)} \setminus \{2(\mu''(i)-1)\ell + \ell + j\}\}$, where we use the robust self-ordering of G''_ℓ and Cases 1&3.

Lastly, we consider vertices that do not fall into any of the prior cases. Such vertices maintain their type, are mapped with the plurality vote of their connected component, which is consistent among its two parts (i.e., μ' and μ''), and maintain their position in that component. Hence, the hypothesis that they are not fixed-points of μ can only be attributed to the fact that these vertices are mapped to a connected component with a different index.

Case 7: *Vertices $v \in F_i$ such that both $\mu(v) \in F_{\mu'(i)} \setminus F_i$ and $\mu(\phi_i(v)) \in S_{\mu''(i)} \setminus S_i$ hold.*

(We may assume that $\mu'(i) \neq i$ and $\mu''(i) \neq i$, since otherwise this set is empty. We may also assume that $\mu'(i) = \mu''(i)$, since the complementary case was covered by Cases 4 and 5. Hence, we focus on pairs of vertices that are matched in the i^{th} connected component of G_n and are mapped by μ to the k^{th} component of G_n such that $k \neq i$.)

For every $i \neq k$, let $\Delta_{i,k} = \{j \in [\ell] : \pi_i(j) \neq \pi_k(j)\}$ be the sets on which π_i and π_k differ. (Note that if for every $v = 2(i-1)\ell + j \in F_i$ it holds that $\mu(v) = 2(k-1)\ell + j$ and $\mu(\phi_i(v)) = 2(k-1)\ell + \pi_i(j)$ (equiv., $\mu(2(i-1)\ell + \ell + \pi_i(j)) = 2(k-1)\ell + \pi_i(j)$), then we get a contribution of $|\Delta_{i,k}|$ to the difference between G_n and $\mu(G_n)$.)
Fixing i, let $D = D_1 \cup D_2$ such that

$$\begin{aligned} D_1 &= \{v \in F_i : \mu(v) \notin F_{\mu'(i)} \ \vee \ \mu(v+\ell) \notin S_{\mu''(i)}\} \\ D_2 &= \left\{ v{=}2(i-1)\ell + j \in F_i : \begin{array}{l} \mu(v) \in F_{\mu'(i)} \setminus \{2(\mu'(i)-1)\ell + j\} \\ \vee \ \mu(\phi_i(v)) \in S_{\mu''(i)} \setminus \{2(\mu''(i)-1)\ell + \ell + \pi_i(j)\} \end{array} \right\} \end{aligned}$$

(Recall that $\phi_i(2(i-1)\ell + j) = 2(i-1)\ell + \ell + \pi_i(j)$. The set D_1 accounts for the vertices covered in Cases 2&3, whereas D_2 accounts for the vertices covered in (the two sub-cases of) Case 6.)
As a warm-up, consider first the case that $D = \emptyset$. In this case, assuming $\mu'(i) = \mu''(i) \neq i$, we get a contribution of $|\Delta_{i,\mu'(i)}| = \Omega(\ell)$ (to the difference between G_n and $\mu(G_n)$). This contribution is due to the difference in the edges that match $F_{\mu'(i)}$ and $S_{\mu'(i)}$ in G_n and the edges that match F_i and S_i in G_n, where $|\Delta_{i,\mu'(i)}| = \Omega(\ell)$ is due to the fact that the permutations (i.e., π_k's) are far-apart. The hypothesis $D_1 = \emptyset$ means that all vertices of F_i (resp., of S_i) are mapped to $F_{\mu'(i)}$ (resp., to $S_{\mu''(i)} = S_{\mu'(i)}$), whereas $D_2 = \emptyset$ means that these vertices preserves their order within the two parts of the connected component.
The general case (i.e., where D may not be empty) requires a bit more care. Suppose that the π_k's are γ-apart; that is, $|\Delta_{k',k}| > \gamma \cdot \ell$ for every $k' \neq k$. We focus on the case that a vast majority of the vertices in both F_i and S_i are mapped according to the plurality votes (i.e., $\mu'(i)$ and $\mu''(i)$), since the complementary cases are covered by Cases 2 and 3, respectively. Specifically, if $|D_1| > \gamma\ell/3$, then we get a contribution of $\Omega(\ell)$ by either Case 2 or Case 3. Likewise, if $|D_2| > \gamma\ell/3$, then we get a contribution of $\Omega(\ell)$ by Case 6. So, assuming $\mu'(i) \neq i$, we are left with the case that

$$|\{v{=}2(i-1)\ell + j \in F_i \setminus D : j \in \Delta_{i,\mu'(i)}\}| \geq \gamma\ell - 2\gamma\ell/3.$$

In this case, assuming $\mu'(i) = \mu''(i)$, we get a contribution of at least $\gamma\ell/3$ to the difference between G_n and $\mu(G_n)$. This contribution is due to the difference in the edges that match $F_{\mu'(i)}$ and $S_{\mu'(i)}$ in G_n and the edges that match F_i and S_i in G_n, where edges that have an endpoint (or its ϕ_i-mate) in D were discarded. Specifically, letting $k = \mu'(i) = \mu''(i) \neq i$, the pair $(v, w) = (2(i-1)\ell + j, 2(i-1)\ell + \ell + \pi_i(j)) \in F_i \times S_i$ contributes to the difference if $j \in \Delta_{i,k}$ and both $\mu(v) = 2(k-1)\ell + j \in F_k$ and $\mu(w) = 2(k-1)\ell + \ell + \pi_i(j) \in S_k$ hold (i.e., $v \notin D_1$ and $v, \phi_i^{-1}(w) \notin D_2$).[20] Indeed, in this case $\{v, w\}$ is an edge in G_n but $\{v, w\}$ is not an edge in $\mu^{-1}(G_n)$. (Hence, if the number of vertices of this case is $\Omega(|\{u \in [n] : \mu(u) \neq u\}|)$, then the difference between G_n and $\mu^{-1}(G_n)$ is $\Omega(|\{u \in [n] : \mu(u) \neq u\}|)$, and the same holds with respect to the difference between $\mu(G_n)$ and G_n.)

Combining all these cases, we get a total contribution that is proportional to $|\{v \in [n] : \mu(v) \neq v\}|$, where we might have counted the same contribution in several different cases. Since the number of cases is a constant, the theorem follows. ◀

Digest: Using large collections of pairwise far apart permutations

The construction presented in the proof of Theorem 4.2 utilizes a collection of $(\ell!)^{\Omega(1)}$ permutations over $[\ell]$ that are pairwise far-apart (i.e., every two permutations differ on $\Omega(\ell)$ inputs). Such a collection is constructed in $\widetilde{O}(\ell!)$-time by an iterative exhaustive

[20] Recall that $\phi_i^{-1}(w) = \phi^{-1}((2(i-1)\ell + \ell + \pi_i(j))) = 2(i-1)\ell + j = v$.

search, where the permutations are selected iteratively such that in each iteration we find a permutation that is far from permutations that were included in previous iterations. We mention that in Section 4.3 we shall use a collection of $\exp(\Omega(\ell))$ such permutations that is locally computable (i.e., given the index of a permutation we find its explicit description in polynomial time). We also mention that, in follow-up work [21], we provided a locally computable collection of $(\ell!)^{\Omega(1)}$ that are pairwise far-apart.

Digest: Combining two robustly self-ordered graphs

One ingredient in the proof of Theorem 4.2 is forming connected components that consist of two robustly self-ordered graphs that have different vertex degrees and are connected by a bounded-degree bipartite graph. Implicit in the proof is the fact that such the resulting graph is robustly self-ordered graph.

▷ Claim 4.3 (combining two $\Omega(1)$-robustly self-ordered graphs). For $i \in \{1,2\}$ and constant $\gamma > 0$, let $G_i = (V_i, E_i)$ be an γ-robustly self-ordered graph, and consider a graph $G = (V_1 \cup V_2, E_1 \cup E_2 \cup E)$ of maximum degree d such that E contain edges with a single vertex in each V_i; that is, G consists of G_1 and G_2 and an arbitrary bipartite graph that connects them. If the maximun degree in G of each vertex in V_1 is strictly smaller than the minimum degree of each vertex in V_2, then G is $\gamma/(2d+3)$-robustly self-ordered.

Proof Sketch. For an arbitrary permutation $\mu : V \to V$, let T denote the set of its non-fixed-points, and consider the following two cases.

Case 1: More than $t = \gamma' \cdot |T|$ vertices are mapped by μ from G_1 to G_2, where $\gamma' = \gamma/(2d+3)$. In this case, we get a contribution of at least one unit per each such vertex, due to the difference in the degrees between V_1 and V_2.

Case 2: at most t vertices are mapped by μ from G_1 to G_2.
In this case, letting T_i denote the set of non-fixed vertices in G_i that are mapped by μ to G_i, we get a contribution of at least $\sum_{i=1,2}(\gamma \cdot |T_i| - d \cdot t)$ units, where the negative term is due to possible change in the incidence with vertices in $T \setminus T_i$. Hence, the total contribution in this case is at least $\gamma \cdot (|T| - 2t) - 2d \cdot t = \gamma' \cdot |T|$.

The claim follows. ◁

Regaining regularity and expansion

While Theorem 4.2 achieves our main objective, it useful towards some applications (see, e.g., the proof of Theorem 4.5) to obtain this objective with graphs that are both regular and expanding. This is achieved by applying Theorem 2.6. Hence, we have.

▶ **Theorem 4.4** (Theorem 4.2, revised). *For any sufficiently large constant d, there exists an efficiently constructable family $\{G_n\}_{n\in\mathbb{N}}$ of robustly self-ordered d-regular expander graphs. That is, there exists a polynomial-time algorithm that on input 1^n outputs the n-vertex graph G_n.*

4.3 Strong (i.e., local) constructions

While Theorem 4.4 provides an efficient construction of robustly self-ordered d-regular expander graphs, we seek a stronger notion of constructability. Specifically, rather than requiring that the graph be constructed in time that is polynomial in its size, we require that the neighbors of any given vertex can be found in time that is polynomial in the vertex's name (i.e., time that is polylogarithmic in the size of the graph). We call such graphs locally constructable (and comment that the term "strongly explicit" is often used in the literature).

▶ **Theorem 4.5** (locally constructing robustly self-ordered graphs)**.** *For any sufficiently large constant d, there exists a locally constructable family $\{G_n = ([n], E_n)\}_{n\in\mathbb{N}}$ of robustly self-ordered d-regular graphs. That is, there exists a polynomial-time algorithm that on input n and $v \in [n]$ outputs the list of neighbours of vertex v in G_n. Furthermore, the graphs are either expanders or consist of connected components of logarithmic size.*

(Indeed, this establishes Theorem 1.3.) We comment that using the result of [21], we can also get connected components of sub-logarithmic size, as in Theorem 4.2.[21]

Proof. We employ the idea that underlies the proof of Theorem 4.2, while starting with an *efficiently constructable family* of robustly self-ordered graphs (as provided by Theorem 4.4) rather than with the mere existence of a family of such graphs (equiv., with ℓ-vertex graphs that can be constructed in poly($\ell!$)-time). We use a slightly larger setting of ℓ, which allows us to use a collection of $\exp(\Omega(\ell))$ pairwise-far-apart permutations (rather than a collection of $\exp(\Omega(\ell \log \ell))$ such permutations). Lastly, we apply the same transformation as in the proof of Theorem 4.4 (so to regain regularity and expansion). Details follow.

Given a generic n, let $\ell = O(\log n)$, which implies that $\exp(\ell) = \text{poly}(n)$. By Theorem 4.4, for all sufficiently large d', we can construct ℓ-vertex d'-regular expander graphs that are robustly self-ordered (with respect to the robustness parameter c) in poly(ℓ)-time. Again, we shall use two such graphs: a d'-regular graph, denoted $G'_\ell = ([\ell], E'_\ell)$, and a d''-regular graph, denoted $G''_\ell = ([\ell], E''_\ell)$, where $d'' = d' + 1$.

Using G'_ℓ and G''_ℓ, we construct an n-vertex robustly self-ordered graph, denoted G_n, that consists of $n/2\ell$ connected components that are pairwise far from being isomorphic to one another. This is done by picking $m = n/2\ell$ permutations, denoted $\pi_1, ..., \pi_m : [\ell] \to [\ell]$, that are pairwise far-apart, and constructing 2ℓ-vertex graphs such that the i^{th} such graph consist of a copy of G'_ℓ and a copy of G''_ℓ that are connected by a matching as determined by the permutation π_i. (as detailed in (7)).

Using the fact that $m < 2^\ell$ (rather that $m = \exp(\Theta(\ell \log \ell))$), we can construct each of these permutations in poly(ℓ)-time by using sequences of disjoint traspositions determined via a good error correcting code. Specifically, for $k = \log_2 m < \log_2 n$, we use an error correcting code $C : \{0,1\}^k \to \{0,1\}^\ell$ of constant rate (i.e., $\ell = O(k)$) and linear distance (i.e., the codewords are $\Omega(\ell)$ bits apart from each other), and let $\pi_i(2j-1) = 2j-1+C(i)_j$ and $\pi_i(2j) = 2j - C(i)_j$, where $i \in [m] = [2^k] \equiv \{0,1\}^k$ and $j \in [\ell/2]$. (That is, the i^{th} permutation switches the pair $(2j-1, 2j) \in [\ell]^2$ if and only if the j^{th} bit in the i^{th} codeword is 1, where $C(i)$ is considered the i^{th} codeword.)

Like in the proof of Theorem 4.2, the i^{th} connected component of G_n is isomorphic to a graph with the vertex set $[2\ell]$ and the edge set

$$E'_\ell \;\cup\; \{\{\ell+u, \ell+v\} : \{u,v\} \in E''_\ell\} \;\cup\; \{\{v, \ell+\pi_i(v)\} : v\in[\ell]\}. \tag{7}$$

The key observation is that, for every $i \in [m]$ and $j \in [\ell]$, the neighborhood of the j^{th} (resp., $(\ell+j)^{\text{th}}$) vertex in the i^{th} connected component of the n-vertex graph G_n is determined by G'_ℓ and $\pi_i(j)$ (resp., by G''_ℓ and $\pi_i^{-1}(j)$), which means that it can be found in poly(ℓ)-time. This implies local constructability, since $\ell = O(\log n)$.

[21] Specifically, the result of [21] provides a construction of a collection of $L = \exp(\Omega(\ell \log \ell))$ permutations over $[\ell]$ that are pairwise far-apart along with a polynomial-time algorithm that, on input $i \in [L]$, returns a description of the i^{th} permutation (i.e., the algorithm should run in poly($\log L$)-time). Using this algorithm, we can afford to set $\ell = \frac{O(\log n)}{\log \log n}$ as in Theorem 4.2.

The fact that G_n is robustly self-ordered was already established in the proof of Theorem 4.2, which is oblivious of the permutations used as long as any pair of permutations disagrees on $\Omega(\ell)$ points. Lastly, we may obtain regularity and expansion by applying Theorem 2.6. ◀

4.4 Local self-ordering

Recall that by Definition 1.1 a graph $G = ([n], E)$ is called self-ordered if for every graph $G' = (V', E')$ that is isomorphic to G there exists a unique bijection $\phi : V' \to [n]$ such that $\phi(G') = G$. One reason for our preferring the term "self-ordered" over the classical term "asymmetric" is that we envision being given such an isomorphic copy $G' = (V', E')$ and asked to find its unique isomorphism to G, which may be viewed as ordering the vertices of G' according to (their name in) G. The task of finding this unique isomorphism will be called *self-ordering G' according to G* or *self-ordering G'* (when G is clear from the context).

Evidently, the task of self-ordering a given graph G' according to a self-ordered graph G that can be efficiently constructed reduces to testing isomorphism. When the graphs have bounded-degree the latter task can be performed in polynomial-time [29]. These are general facts that do apply also to the robustly self-ordered graph G_n constructed in the proof of Theorem 4.5. However, in light of the fact that the graph G_n is *locally* constructable, we can hope for more. Specifically, it is natural to ask if we can perform self-ordering of a graph G' that is isomorphic to G_n in a *local* manner; that is, given a vertex in G' (and oracle access to the incidence function of G'), can we find the corresponding vertex in G_n in poly$(\log n)$-time? Let us define this notion formally.

▶ **Definition 4.6** (locally self-ordering a self-ordered graph). *We say that a self-ordered graph $G = ([n], E)$ is* locally self-ordered *if there exists a polynomial-time algorithm that, given a vertex v in any graph $G' = (V', E')$ that is isomorphic to G and oracle access to the incidence function of G', finds $\phi(v) \in [n]$ for the unique bijection $\phi : V' \to [n]$ such that $\phi(G') = G$ (i.e., the unique isomorphism of G' to G).*

Indeed, the isomorphism ϕ orders the vertices of G' in accordance with the original (or target) graph G. We stress that the foregoing algorithm works in time that is polynomial in the description of a vertex (i.e., poly$(\log n)$)-time), which is polylogarithmic in the size of the graph (i.e., n). We show that such algorithms exist for the graphs constructed in the proof of Theorem 4.5.

▶ **Theorem 4.7** (locally self-ordering the graphs of Theorem 4.5). *For any sufficiently large constant d, there exists a locally constructable family $\{G_n = ([n], E_n)\}_{n\in\mathbb{N}}$ of robustly self-ordered d-regular graphs that are locally self-ordered. Furthermore, the graphs are either expanders or consist of connected components of logarithmic size.*

As in Theorem 4.5, we can obtain connected components of sub-logarithmic size by using [21].

Proof. We first consider the version that yields n-vertex graphs that consist of connected components of logarithmic size. The basic idea is that it we can afford reconstructing the connected component in which the input vertex reside, and this allows us both to determine the index of the vertex in this connected component as well as the index of the component in the graph. Specifically, on input a vertex v in a graph G' that is isomorphic to G_n, we proceed as follows.

1. Using queries to the incidence function of G', we explore and retrieve the entire 2ℓ-vertex connected component in which v resides, where $\ell = \log_2 n$.
   Recall that this connected component consists of (copies of) two ℓ-vertex regular graphs, denoted G'_ℓ and G''_ℓ, that are connected by a matching. Furthermore, these graphs have different degrees and are each (robustly) self-ordered.
2. Relying on the different degrees, we identify the foregoing partition of this 2ℓ-vertex component into two ℓ-vertex (self-ordered) graphs, denoted A_v and B_v, where A_v (resp., B_v) is isomorphic to G'_ℓ (resp., G''_ℓ).
3. Relying on the self-ordering of G'_ℓ (resp., G''_ℓ), we order the vertices of A_v (resp., G''_v). This is done by constructing G'_ℓ (resp., G''_ℓ), and using an isomorphism tester. The order of the vertices in A_v and B_v also determines the permutation that defines the matching between the two graphs.
4. Relying on the correspondence between the permutations used in the construction and codewords of a good error-correcting code, we decode the relevant codeword (i.e., this is decoding without error). This yields the index of the permutation in the collection, which equals the index of the connected component.

Note that this refers to the basic construction that was presented in the proof of Theorem 4.5, before it was transformed to a regular graph and to an expander. Recall that both transformations are performed by augmenting the graph with auxiliary edges that are assigned a different color than the original edges, and that edges with different colors are later replaced by copies of different (constant-size) gadgets. These transformations do not hinder the local self-ordering procedure described above, since it may identify the original graph (and ignore the gadgets that replace other edges). The claim follows. ◀

Local reversed self-ordering

While *local self-ordering* a (self-ordered) graph seems *the natural local version* of self-ordering the graph, an alternative notion called *local reversed self-ordering* will be defined and studied next (and used in Section 5). Both notions refer to a self-ordered graph, denoted $G = ([n], E)$, and to an isomorphic copy of it, denoted $G' = (V', E')$; that is, $G = \phi(G')$ for a (unique) bijection $\phi : V' \to [n]$. While local self-ordering is the task of finding the index of a given vertex of G' according to G (i.e., given $v \in V'$, find $\phi(v) \in [n]$), local reversed self-ordering is the task of finding the vertex of G' that has a given index in G (i.e., given $i \in [n]$, find $\phi^{-1}(i) \in V'$). In both cases, the graph G is locally constructible and we are given oracle access to the incidence function of G'. In addition, in the reversed task, we assume that the algorithm is given an arbitrary vertex in G', since otherwise there is no hope to hit any element of V'.[22]

▶ **Definition 4.8** (locally reversed self-ordering). *We say that a self-ordered graph $G = ([n], E)$ is* locally reversed self-ordered *if there exists a polynomial-time algorithm that, given $i \in [n]$ and oracle access to the incidence function of a graph $G' = (V', E')$ that is isomorphic to G and an arbitrary vertex $s \in V'$, finds $\phi^{-1}(i) \in V'$ for the unique bijection $\phi : V' \to [n]$ such that $\phi(G') = G$* (i.e., the unique isomorphism of G' to G)*.*

We stress that the foregoing algorithm works in time that is polynomial in the description of a vertex (i.e., $\text{poly}(\log n)$)-time), which is polylogarithmic in the size of the graph (i.e., n). We show that such algorithms exist for variants of the graphs constructed in the proof of Theorem 4.5. In fact, we show a more general result that refers to any graph that is locally self-ordered and for which short paths can be locally found between any given pair of vertices.

[22] Needless to say, this is not needed in case $V' = [n]$, which is the case that is used in Section 5.

▶ **Theorem 4.9** (sufficient conditions for locally reversed self-ordering of graphs)**.** *Suppose that $\{G_n = ([n], E_n)\}_{n\in\mathbb{N}}$ is a family of bounded degree graphs that is locally self-ordered. Further suppose that given $v, u \in [n]$, one can find in polynomial-time a path from u to v in G_n. Then, $\{G_n = ([n], E_n)\}_{n\in\mathbb{N}}$ is locally reversed self-ordered.*

We mention that a family of robustly self-ordered graphs that is locally self-ordered can be transformed into one that also supports locally finding short paths. This is done by superimposing the graphs of this family with graphs that supports locally finding short paths, while using different colors for the edges of the two graphs and later replacing these colored edges by gadgets (as done in Section 2.1). We also mention that applying degree reduction to the hyper-cube (i.e., replacing the original vertices with simple cycles) yields a graph that supports locally finding short paths.[23]

Proof. On input $i \in [n]$ and $s \in V'$, and oracle access to the incidence function of a graph $G' = (V', E')$ that is isomorphic to G_n, we proceeds as follows.

1. Using the local self-ordering algorithm, we find $i_0 = \phi(s)$, where $\phi : V' \to [n]$ is the unique bijection satisfying $\phi(G') = G$.
2. Using the path-finding algorithm for G, we find a $\text{poly}(\log n)$-long path from i_0 to i in G. Let ℓ denote the length of the path, and denote its intermediate vertices by $i_1, ..., i_{\ell-1}$; that is, the full path is $i_0, i_1, ..., i_{\ell-1}, i_\ell = i$.
3. For $j = 1, ..., \ell$, we find $v_j \overset{\text{def}}{=} \phi^{-1}(i_j)$ as follows. First, using queries to the incidence function of G', we find all neighbors (in G') of v_{j-1}, where $v_0 \overset{\text{def}}{=} s$ (and, indeed, $v_0 = \phi^{-1}(i_0)$). Next, using the local self-ordering algorithm, we find the indices of all these vertices in G; that is, for every vertex w that neighbors v_{j-1}, we find $\phi(w)$. Last, we set v_j to be the neighbor that has index i_j in G; that is, v_j satisfies $\phi(v_j) = i_j$.

Hence, v_ℓ is the desired vertex; that is, v_ℓ satisfies $\phi(v_\ell) = i_\ell = i$.

Assuming that the local self-ordering algorithm has query complexity $q(n)$, that the paths found in G have length at most $\ell(n)$, and that d is the degree bound, the query complexity of our reversed self-ordering algorithm is $(1 + \ell(n) \cdot d) \cdot (q(n) + 1)$, where we count both our direct queries to the incidence function of G and the queries performed by the local self-ordering algorithm. Similar considerations apply to its time complexity. ◀

▶ **Corollary 4.10** (a version of Theorem 4.7 supporting local reversed self-ordering)**.** *For any sufficiently large constant d, there exists a locally constructable family $\{G_n = ([n], E_n)\}_{n\in\mathbb{N}}$ of robustly self-ordered graphs of maximum degree d that are both locally self-ordered and locally reversed self-ordered.*

The corollary follows by combining Theorem 4.7 with Theorem 4.9, while using the augmentation outlined following the statement of Theorem 4.9. We mention that Corollary 4.10 will be used in Section 5.

[23] For any $\ell \in \mathbb{N}$, the resulting graph consists of the vertex-set $\{\langle x, i\rangle : x \in \{0,1\}^\ell \,\&\, i \in [\ell]\}$ and edges that connect $\langle x, i\rangle$ to $\langle x \oplus 0^{i-1}10^{\ell-i}, i\rangle$ and to $\langle x, i+1\rangle$, where $\ell + 1$ stands for 1. For simplicity of exposition, we also add self-loops on all vertices. Then, given $\langle x, i\rangle$ and $\langle y, j\rangle$, we can combine the 2ℓ-path that goes from $\langle x, i\rangle$ to $\langle y, i\rangle$ with the $|j - i|$-path that goes from $\langle y, i\rangle$ to $\langle y, j\rangle$, where the odd steps on the first path move from $\langle z, k\rangle$ to $\langle z \oplus 0^{i-1}10^{\ell-i}, k\rangle$ (or stay in place) and the even steps (on this path) move from $\langle z, k\rangle$ to $\langle z, k+1\rangle$.

5 Application to Testing Bounded-Degree Graph Properties

Our interest in efficiently constructable bounded-degree graphs that are robustly self-ordered was triggered by an application to property testing. Specifically, we observed that such constructions can be used for proving a linear lower bound on the query complexity of testing an *efficiently recognizable* graph property in the *bounded-degree graph model*.

It is well known that 3-Colorability has such a lower bound [3], but this set is NP-complete. On the other hand, linear lower bounds on the query complexity of testing efficiently recognizable properties of *functions* (equiv., sequences) are well known (see [18, Sec. 10.2.3]). So the idea was to transport the latter lower bounds from the domain of functions to the domain of bounded-degree graphs, and this is where efficient constructions of robustly self-ordered bounded-degree graphs come into play. (We mention that an alternative way of obtaining the desired lower bound was outlined in [17, Sec. 1], see details below.)

More generally, the foregoing transportation demonstrates a general methodology of transporting lower bounds that refer to testing binary strings to lower bounds regarding testing graph properties in the bounded-degree graph model. The point is that strings are ordered objects, whereas graphs properties are effectively sets of unlabeled graphs, which are unordered objects. Hence, we need to make the graphs (in the property) ordered, and furthermore make this ordering robust in the very sense that is reflected in Definition 1.2. Essentially, we provide a reduction of testing a property of strings to testing a (related) property of graphs.

We apply this methodology to obtain a subexponential separation between the complexities of testing and tolerant testing of graph properties in the bounded-degree graph model. This result is obtained by transporting an analogous result that was known for testing binary strings [15]. In addition to using a reduction from tolerantly testing a property of strings to tolerantly testing a property of graphs, this trasportation also uses a reduction in the opposite direction, which relies on the local computation features asserted in Corollary 4.10.

Organization of this section

We start with a brief review of the bounded-degree graph model for testing graph properties. Next, we prove the aforementioned linear lower bound on the query complexity of testing an efficiently recognizable property, and later we abstract the reduction that underlies this proof. Observing that this reduction applies also to tolerant testing, and presenting a reduction in the opposite direction, we derive the aforementioned separation between testing and tolerant testing.

Background

Property testing refers to algorithms of sublinear query complexity for *approximate decision*; that is, given oracle access to an object, these algorithms (called testers) distinguish objects that have a predetermined property from objects that are far from the property. Different models of property testing arise from different query access and different distance measures.

In the last couple of decades, the area of property testing has attracted significant attention (see, e.g., [16]). Much of this attention was devoted to testing graph properties in a variety of models including the dense graph model [18], and the bounded-degree graph model [20] (surveyed in [16, Chap. 8] and [16, Chap. 9], resp.). In this section, we refer to the bounded-degree graph model, in which graphs are represented by their incidence function and distances are measured as the ratio of the number of differing incidences to the maximal number of edges.

Specifically, for a degree bound $d \in \mathbb{N}$, we represent a graph $G = ([n], E)$ of maximum degree d by the incidence function $g : [n] \times [d] \to [n] \cup \{0\}$ such that $g(v, i)$ indicates the i^{th} neighbor of v (where $g(v, i) = 0$ indicates that v has less than i neighbors). The distance between the graphs $G = ([n], E)$ and $G' = ([n], E')$ is defined as the size of the symmetric difference between E and E' over $dn/2$.

A tester for a property Π is given oracle access to the tested object, where here oracle access to a graph means oracle access to its incidence function. In addition, such a tester is given a size parameter n (i.e., the number of vertices in the graph), and a proximity parameter, denoted $\epsilon > 0$. Tolerant testers, introduced in [30] (and briefly surveyed in [16, Sec. 12.1]), are given an additional parameter, $\eta < \epsilon$, which is called the tolerance parameter.

▶ **Definition 5.1** (testing and tolerant testing graph properties in the bounded-degree graph model). *For a fixed* degree bound d*, a* tester *for a graph property* Π *is a probabilistic oracle machine that, on input parameters* n *and* ϵ*, and oracle access to an* n*-vertex graph* $G = ([n], E)$ *of maximum degree* d*, outputs a binary verdict that satisfies the following two conditions.*

1. *If* $G \in \Pi$*, then the tester accepts with probability at least* $2/3$*.*
2. *If* G *is* ϵ*-far from* Π*, then the tester accepts with probability at most* $1/3$*, where* G *is* ϵ-far *from* Π *if for every* n*-vertex graph* $G' = ([n], E') \in \Pi$ *of maximum degree* d *it holds that the size of the symmetric difference between* E *and* E' *has cardinality that is greater than* $\epsilon \cdot dn/2$*.*

A tolerant tester *is also given a* tolerance parameter η*, and is required to accept with probability at least* $2/3$ *any graph that is* η*-close to* Π (i.e., not η-far from Π).[24]

We stress that a graph property is defined as a property that is preserved under isomorphism; that is, if $G = ([n], E)$ is in the graph property Π, then all its isomorphic copies are in the property (i.e., $\pi(G) \in \Pi$ for every permutation $\pi : [n] \to [n]$). The fact that we deal with graph properties (rather than with properties of functions) is the source of the difficulty (of transporting results from the domain of functions to the domain of graphs) and the reason that robust self-ordering is relevant.[25]

The query complexity of a tester for Π is a function (of the parameters d, n and ϵ) that represents the number of queries made by the tester on the worst-case n-vertex graph of maximum degree d, when given the proximity parameter ϵ. Fixing d, we typically ignore its effect on the complexity (equiv., treat d as a hidden constant). Also, when stating that the query complexity is $\Omega(q(n))$, we mean that this bound holds for all sufficiently small $\epsilon > 0$; that is, there exists a constant $\epsilon_0 > 0$ such that distinguishing between n-vertex graphs in Π and n-vertex graphs that are ϵ_0-far from Π requires $\Omega(q(n))$ queries.

Our first result

With the foregoing preliminaries in place, we state the first result of this section, which is proved using Theorem 4.2.

▶ **Theorem 5.2** (linear query complexity lower bound for testing an efficiently recognizable graph property in the bounded-degree graph model). *For any sufficiently large constant* d*, there exists an efficiently recognizable graph property* Π *such that testing* Π *in the bounded-degree graph model* (with degree bound d) *has query complexity* $\Omega(n)$*. Furthermore, each* n*-vertex graph in* Π *consists of connected components of size* $o(\log n)$*.*

[24] Of course, a tolerant tester is also required to reject with probability at least $2/3$ any graph that is ϵ-far from Π.

[25] As noted in Section 1.1.1, this is a special case of the general phenomenon pivoted at the difference between ordered and unordered structures, which arises in many contexts (in complexity and logic).

The main part of the theorem was known before: As observed in [17, Sec. 1], *there exists graph properties that are recognizable in polynomial-time and yet are extremely hard to test in the bounded-degree graph model.* This follows from the fact that the local reduction from testing 3LIN (mod 2) to testing 3-Colorability used by Bogdanov, Obata, and Trevisan [3] is invertible in polynomial-time (which is a common feature of reductions used in the context of NP-completeness proofs).[26] Indeed, their reduction actually demonstrates that the set of (3-colorable) graphs that are obtained by applying this reduction to satisfiable 3LIN (mod 2) instances is hard to test (i.e., requires linear query complexity in the bounded-degree graph model).[27] We note that the resulting property contains only connected graphs, which means that Theorem 5.2 has some added value: The fact that it applies to graphs with tiny connected components is interesting, since testing properties of such graphs may seem easy (or at least not extremely hard) at first thought.

Proof. Our starting point is a property Φ of (binary) strings (equiv., Boolean functions) that is recognizable in polynomial-time but has a linear query complexity lower bound (see, e.g., [19, Sec. 7]). This refers to a model in which one makes queries to bits of the tested string, and the distance between strings is the (relative) Hamming distance. Such lower bounds were transported to the *dense graph model* in [18, 10.2.3] (see also [19]), but – to the best of own knowledge – no such transportation were performed before in the context of the bounded-degree graph model. Using robustly self-ordered graphs of bounded degree, we present such a transportation.

▷ Construction 5.2.1 (from properties of strings to properties of bounded-degree graphs). Suppose that $\{G_n = ([n], E_n)\}_{n\in\mathbb{N}}$ is a family of robustly self-ordered graphs of maximum degree $d-2$.

- For every $n \in \mathbb{N}$ and $s \in \{0,1\}^n$, we define the graph $G'_s = ([3n], E'_s)$ such that

$$E'_s = E_n \cup \{\{i, n+i\}, \{i, 2n+i\} : i \in [n]\} \cup \{\{n+i, 2n+i\} : i \in [n] \wedge s_i = 1\} \tag{8}$$

  That is, G'_s consists of a copy of G_n augmented by $2n$ vertices such that vertex $i \in [n]$ forms a triangle with $n+i$ and $2n+i$ is $s_i = 1$, and forms a wedge with $n+i$ and $2n+i$ otherwise.
- For a set of strings Φ, we define $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$ as the set of all graphs that are isomorphic to some graph G'_s such that $s \in \Phi$; that is,

$$\Pi_n = \{\pi(G'_s) : s \in (\Phi \cap \{0,1\}^n) \wedge \pi \in \mathrm{Sym}_{3n}\} \tag{9}$$

  where Sym_{3n} denote the set of all permutations over $[3n]$.

We may assume, without loss of generality, that G_n has no isolated vertices. Hence, given a graph of the form $\pi(G'_s)$, the vertices of G_n are easily identifiable as having degree at least three (since vertices outside G_n have degree at most two). The foregoing construction yields a local reduction of Φ to Π, where locality means that each query to G'_s can be answered by making a constant number of queries to s, and the (standard) validity of the reduction is based on the fact that G_n is asymmetric.[28]

[26] Of course, 3LIN (i.e., the satisfiability of linear equations (with three variables each) over GF(2)) is easily solvable in polynomial-time. Nevertheless, Bogdanov et al. [3] use a reduction of 3LIN to 3-Colorability (via 3SAT) that originates in the theory of NP-completeness in order to reduce between the testing problems.

[27] Like almost all reductions of this type, the analysis of the reduction actually refers to the promise problem induced by the image of the reduction (i.e., the image of both the yes- and no-instances).

[28] Standard validity means that $s \in \Phi$ if and only if $G'_s \in \Pi$. Evidently, $s \in \Phi$ is mapped to $G'_s \in \Pi$; the asymmetry of G_n is used to show that $s \notin \Phi$ is mapped to $G'_s \notin \Pi$, since G'_s can not be isomorphic to any graph G'_w such that $w \neq s$. This, by itself, does not mean that if s is far from Φ then G'_s is far from Π.

In order to be useful towards proving lower bounds on the query complexity of testing Π, we need to show that the foregoing reduction is "distance preserving" (i.e., strings that are far from Φ are transformed into graphs that are far from Π). The hypothesis that G_n is robustly self-ordered is pivotal to showing that if the string s is far from Φ, then the graph G'_s is far from Π.

▷ Claim 5.2.2 (preserving distances). If $s \in \{0,1\}^n$ is ϵ-far from Φ, then the $3n$-vertex graph G'_s (as defined in Construction 5.2.1) is $\Omega(\epsilon)$-far from Π.

Proof. We prove the contrapositive. Suppose that G'_s is δ-close to Π. Then, for some $r \in \Phi$ and a permutation $\pi : [3n] \to [3n]$, it holds that G'_s is δ-close to $\pi(G'_r)$. (The possible use of a non-trivial permutation arises from the fact that Π is closed under isomorphism.) If $\pi(i) = i$ for every $i \in [n]$, then s must be $(3d\delta/2)$-close to r, where d is the degree bound (of the model), since $s_i = 1$ (resp., $r_i = 1$) if and only if i forms a triangle with $n+i$ and $2n+i$ in G'_s (resp., in $\pi(G'_r) = G'_r$).[29] Unfortunately, the foregoing condition (i.e., $\pi(i) = i$ for every $i \in [n]$) need not hold in general.

In general, the hypothesis that $\pi(G'_r)$ is δ-close to G'_s implies that π maps at most $3\delta dn/2$ vertices of $[n]$ to $\{n+1, ..., 3n\}$. This is the case since each vertex of $[n]$ has degree at least three in G'_r, whereas the other vertices have degree at most two in G'_s (or in any other graph $G'_{s'}$). Hence, if $t = |\{i \in [n] : \pi(i) \in \{n+1, ..., 3n\}|$, then $\pi(G'_r)$ and G'_s differ on at least t edges, whereas the hypothesis is that the difference is at most $\delta \cdot 3dn/2$.

Turning to the vertices $i \in [n]$ that π maps to $[n] \setminus \{i\}$, we upper-bound their number by $O(\delta d^2 n)$, since the difference between $\pi(G'_r)$ and G'_s is at most $\delta \cdot 3dn/2$, whereas the hypothesis that G_n is c-robustly self-ordered implies that the difference between $\pi(G'_r)$ and G'_s (or any other graph G'_w) is at least

$$\Delta = c \cdot |\{i \in [n] : \pi(i) \neq i\}| - d \cdot |\{i \in [n] : \pi(i) \notin [n]\}|.$$

(Compare Case 6 in the proof of Theorem 4.2.)[30]

Letting $I = \{i \in [n] : \pi(i) = i\}|$, observe that $D \overset{\text{def}}{=} |\{i \in I : r_i \neq s_i\}| \leq 3\delta dn/2$, since $r_i \neq s_i$ implies that, for every $i \in I$, the subgraph induced by $\{i, n+i, 2n+1\}$ is different in $\pi(G'_r)$ and G'_s (i.e., it is a triangle in one graph and contains two edges in the other), whereas by the hypothesis $\pi(G'_r)$ and G'_s differ on at most $\delta \cdot 3dn/2$ edges. Recalling that $|I| = n - O(\delta d^2 n)$, it follows that $|\{i \in [n] : r_i \neq s_i\}| \leq (n - |I|) + D = O(\delta d^2 n)$. Recalling that d is a constant, we infer that s is $O(\delta)$-close to $r \in \Phi$, and the claims follows. ◁

Conclusion. Starting with Theorem 4.2 (i.e., an efficient construction of robustly self-ordered graphs of bounded degree), using Construction 5.2.1, and applying Claim 5.2.2, the theorem follows. Specifically, we need to verify the following facts.

[29] Hence, G'_s is δ-close to G'_r implies that $|\{i \in [n] : s_i \neq r_i\}| \leq \delta \cdot 3dn/2$, which means that s is $\frac{3\delta dn/2}{n}$-close to r.

[30] Hence, $\Delta \leq \delta \cdot 3dn/2$ implies that

$$\begin{aligned} |\{i \in [n] : \pi(i) \neq i\}| &= \frac{\Delta + d \cdot |\{i \in [n] : \pi(i) \notin [n]\}|}{c} \\ &\leq \frac{3\delta dn/2 + d \cdot 3\delta dn/2}{c} \end{aligned}$$

which is $O(\delta d^2 n)$.

- The set Π is polynomial-time recognizable.

  Given an $3n$-vertex graph G', an adequate algorithm first tries to identify and order the vertices of the corresponding graph G_n, which means that it finds $s \in \{0,1\}^n$ such that G' is isomorphic to G'_s (or determines that no such s exists). (Note that once the vertices of G_n are identified, their unique ordering, whenever it exists, can be found in polynomial time by running an isomorphism tester on the subgraph induced by them (while relying on the fact that the degree of the graph is bounded [29]).) Having found s, the algorithm accepts if and only if $s \in \Phi_n$, where Φ is polynomial-time recognizable by our starting hypothesis.
- Testing Π requires linear query complexity.

  This is shown by reducing testing Φ to testing Π, while recalling that testing Φ requires linear query complexity. Given (proximity parameter ϵ and) oracle access to a string $s \in \{0,1\}^n$, we invoke the tester for Π (with proximity parameter $\Omega(\epsilon)$) while emulating oracle access to G'_s in a straightforward manner (i.e., each query to G'_s is answered by making at most one query to s). Recall that $s \in \Phi$ implies $G'_s \in \Pi$, whereas by Claim 5.2.2 if s is ϵ-far from Φ then G'_s is $\Omega(\epsilon)$-far from Π.

This completes the proof, since the n-vertex graphs of Theorem 4.2 have connected components of size $o(\log n)$. ◀

Digest: Reducing testing properties of strings to testing graph properties

We wish to highlight the fact that the proof of Theorem 5.2 is based on a general reduction of testing any property Φ of strings to testing a corresponding (bounded-degree) graph property Π. This reduction is described in Construction 5.2.1 and its validity is proved in Claim 5.2.2. Recall that, for any n, the graph property Π consists of $3n$-vertex graphs (of bounded-degree) that encode the different n-bit long strings in Φ. This reduction is local and preserves distances:

Locality: Each string $s \in \{0,1\}^n$ is encoded by a graph G'_s such that each query to G'_s can be answered by making at most one query to s.

Preserving distances: If $s \in \Phi$ then $G'_s \in \Pi$, whereas if s is ϵ-far from Φ then G'_s is $\Omega(\epsilon)$-far from Π.

Recall that G'_s consists of a fixed robustly self-ordered n-vertex graph G_n augmented by (n two-vertex) gadgets that encode s. Let us spell out the effect of this reduction.

▶ **Corollary 5.3** (implicit in the proof of Theorem 5.2). *For Φ and Π as in Construction 5.2.1, let Q_Φ and Q_Π denote the query complexities of testing Φ and Π, respectively. Then, $Q_\Phi(n,\epsilon) \leq Q_\Pi(3n,\Omega(\epsilon))$. Likewise, letting Q'_Φ* (resp., Q'_Π) *denote the query complexity of tolerantly testing Φ* (resp., Π)*, it holds that $Q'_\Phi(n,\eta,\epsilon) \leq Q'_\Pi(3n,\eta/3,\Omega(\epsilon))$.*

The tolerant testing part requires an additional justification. Specifically, we observe that strings s that are η-close to Φ yield graphs G'_s that are $\eta/3$-close to Π. This is the case because, if the n-bit long strings s and r differ on k bits, then the $3n$-vertex graphs G'_s and G'_r differ on k vertex pairs. In preparation to proving the separation between the complexities of testing and tolerant testing, we show a reduction in the opposite direction. This reduction holds provided that the robustly self-ordered graphs used in the definition of Π are locally reversed self-ordered (see Definition 4.8).

▶ **Proposition 5.4** (reducing testing Π to testing Φ). *Suppose that the graphs used in Construction 5.2.1 are locally self-ordered and locally reversed self-ordered, and let Φ, Π and Q_Φ, Q_Π be as in Corollary 5.3. Then, $Q_\Pi(3n, \epsilon) \leq \mathrm{poly}(\log n) \cdot (Q_\Phi(n, 2\epsilon) + O(1/\epsilon))$. Furthermore, one-sided error probability is preserved.*[31]

Recall that the hypothesis can be met by using Corollary 4.10.

Proof. Given oracle access to a graph $G' = ([3n], E')$, we first test that G' is isomorphic to G'_s, for some $s \in \{0,1\}^n$, and then invoke the tester for Φ while providing it with oracle access to s. Specifically, when the latter tester queries the bit i, we use the local reversed self-order algorithm in order to locate the i^{th} vertex of G_n in G', and then determine the bit s_i accordingly. Details follow.

Let V denote the set of vertices of the graph $G' = ([3n], E')$ that have degree greater than 2 and neighbor two vertices that have degree at most 2 and neighbor each other if they have degree 2. Evidently, the vertices of V are easy to identify by querying G' for their neighbors and their neighbors' neighbors. Furthermore, $|V| \leq n$, since each vertex in V has two neighbors that are not connected to any other vertex in V, and equality holds in case $G' \in \Pi$. We try to find a ("pivot") vertex $p \in V$ by picking an arbitrary vertex in G' and checking it and its neighbors. If none of these is in V, then we reject. Otherwise, we continue; we shall be using p as an auxiliary input in all (future) invocations of the local reversed self-ordering algorithm, denoted A.

Using the foregoing algorithm A and the pivot $p \in V$, we define $A'(i) = A(p, i)$ if $A(p, i) \in V$ and invoking the local self-ordering algorithm on input $A(p, i)$ yields i. Otherwise $A'(i)$ is undefined. Hence, evaluating A' amounts to evaluating A as well as evaluating the local self-ordering algorithm. Letting $I' \subseteq [n]$ denote the set of "indices" (i.e., vertices of G_n) on which A' is defined, we note that A' is a bijection from I' to $V' \stackrel{\text{def}}{=} \{A'(i) : i \in I'\}$, and that $I' = [n]$ if $G' \in \Pi$. Hence, our first test is testing whether $I' = [n]$, which is done by selecting at random $O(1/\epsilon)$ elements of $[n]$, and rejecting if A' is undefined on any of them. Otherwise, we proceed, while assuming that $|I'| \geq (1 - 0.1\epsilon) \cdot n$.

Next, we test whether the subgraph of G_n induced by I' is isomorphic to the subgraph of G' induced by V', where the isomorphism is provided by A' (which maps I' to V'). This can be done by sampling $O(1/\epsilon)$ vertices of G_n and comparing their neighbors to the neighbors of the corresponding vertices in G', which are found by A'. Specifically, for every sampled vertex $i \in [n]$, we determine its set of neighbors S_i in G_n, obtain both $A'(i)$ and $A'(S_i) = \{A'(j) : j \in S_i\}$, which are supposedly the corresponding vertices in G', and check whether $A'(S_i)$ is the set of neighbors of $A'(i)$ in G'. We reject if A' is undefined on any of these vertices (i.e., on sampled vertices or their neighbors in G_n). Needless to say, we also reject if any of the foregoing neighborhood checks fails.

Assuming that we did not reject so far, we may assume that G' is $\epsilon/2$-close to being isomorphic to some G'_s, where the isomorphism is consistent with the inverse of A'. At this point, we invoke the tester for Φ, denoted T, in order to test whether $s \in \Phi$. This is done by providing T with oracle access to s as follows. When T makes a query $i \in [n]$, we determine $A'(i)$, and use our query access to G' in order to determine the two neighbors of $A'(i)$ that have degree at most 2. If this fails, we reject. Otherwise, we answer 1 if and only if these two neighbors are connected in G'.

[31] A tester is said to have one-sided error probability if it always accepts objects that have the property.

To summarize, we employ three tests to G': An *initial test* of the size I' (which also includes finding a pivot $p \in V$), an *isomorphism test* between the subgraph of G' induced by I' and the subgraph of G_n induced by V', and an emulation of the testing of Φ. (In all tests, if we encounter an index in $[n] \setminus I'$, we suspend the execution and reject.) For simplicity and without loss of generality, we may assume that T is correct with high (constant) probability.

Note that if $G' \in \Pi$, then it holds that $G' = \pi(G'_s)$ for some $s \in \Phi$ and some permutation $\pi \in \mathrm{Sym}_{3n}$. In this case, it holds that $|I'| = n$ and we always find a pivot $p \in V$. Furthermore, A' equals the restriction of π to $[n]$, the isomorphism test always succeeds, and the emulation of oracle access to s is perfect. Hence, we accept with high probability (or always, if T has one-sided error probability).

On the other hand, suppose that G' is ϵ-far from Π. If either $|I'| < (1 - 0.1\epsilon) \cdot n$ or the subgraph of G' induced by V' is 0.1ϵ-far from $A'(G_{I'})$, where $G_{I'}$ denotes the subgraph of G_n induced by I', then we reject with high probability due to one of the first two tests. Otherwise, letting π be an arbitrary bijection of $[3n]$ to $[3n]$ that extends A', it follows that for some $s \in \{0,1\}^n$ the graph G' is 0.2ϵ-close to $\pi(G'_s)$, since we may obtain $\pi(G'_s)$ from G' by modifying the neighborhood of $0.1n$ vertices in I' as well as of the vertices in $[n] \setminus I'$. Furthermore, for every $i \in [n]$ on which A' is defined, it holds that $s_i = 1$ if and only if the two neighbors of $A'(i)$ that have degree at most 2 are connected. By the hypothesis regarding G', the string s must be 2.4ϵ-far from Φ, and $A'(i) = \pi(i)$ whenever A' is defined on $i \in [n]$. It follows that either the emulation of T was abruptly terminated (leading to rejection) or the answers provided to T are according to s. Hence, we reject with high probability. ◀

Separating tolerant testing from testing

Using Corollary 5.3 and Proposition 5.4, we transport the separation of tolerant testing from testing, which has been established in [15], from the domain of testing strings to the domain of testing graph properties in the bounded-degree graph model.

▶ **Theorem 5.5** (in the bounded-degree graph model, tolerant testing is harder than testing)**.** *For any sufficiently large constant d and any constant $c \in (0,1)$, there exists a graph property Π such that testing Π in the bounded-degree graph model* (with degree bound d) *has query complexity $O(\mathrm{poly}(\log n)/\epsilon)$, but tolerantly testing Π has query complexity $\Omega(n^{\Omega(1-c)})$, provided that the tolerance parameter is not smaller than n^{-c}. Furthermore, Π is efficiently recognizable.*

Proof. A small variant on the proof of [15, Thm. 1.3] yields an efficiently recognizable set of strings Φ that is testable in $O(1/\epsilon)$ queries but tolerantly testing it requires $\Omega(n^{\Omega(1-c)})$ queries.[32] Using Construction 5.2.1 with graphs that are locally self-ordered and locally reversed self-ordered (as provided by Corollary 4.10), we obtain the desired graph property Π. By Corollary 5.3 tolerantly testing Π requires $\Omega(n^{\Omega(1)})$ queries, whereas by Proposition 5.4 (non-tolerant) testing Π has query complexity $\mathrm{poly}(\log n) \cdot O(1/\epsilon)$. The claim follows. ◀

[32] Basically, the construction of [15] consists of repeating some m-bit long string $\mathrm{poly}(m)$ times and augmenting it with a PCP of Proximity (PCPP) [2, 11] of membership in some polynomial-time recognizable set that is hard to test. Essentially, the PCPP helps the tester, but it may be totally useless (when corrupted) in the tolerant testing setting. While [15] lets the PCPP occupy an $o(1/\log\log n)$ fraction of the final n-bit string, we let it occupy just a n^{-c} fraction (and use $m = n^{\Omega(1-c)}$). This requires using a different PCPP than the one used in [15]; e.g., using a strong PCPP with linear detection probability [10, Def. 2.2] will do, and such a PCPP is available [10, Thm. 3.3].

Digest: Tightly reducing testing properties of strings to testing graph properties

In continuation to (the main part of) Corollary 5.3, we highlight the fact that Construction 5.2.1 not only reduces testing the string property Φ to testing the graph property Π, but rather does so in a rather tight manner. Specifically, for Φ, Π and Q_Φ, Q_Π as in Corollary 5.3, it holds that $Q_\Phi(n, \epsilon)$ and $Q_\Pi(\Theta(n), \Theta(\epsilon))$ agree up to a poly$(\log n)$ factor. In other words, *for any property of strings* Φ*, there exists a property of bounded-degree graphs* Π *such that the* (*query and time*) *complexity of testing* Φ *is reflected in the* (*query and time*) *complexity of testing* Π, where our notion of reflection allows for a polylogarithmic slackness. Recall that the transformation of strings in Φ to graphs in Π is (strongly/locally) efficient.

6 Random Regular Graphs are Robustly Self-Ordered

While Theorem 4.1 only asserts the existence of robustly self-ordered d-regular graphs, we next show that almost all d-regular graphs are robustly self-ordered. This extends work in probabilistic graph theory, which proves a similar result for the weaker notion of self-ordered (a.k.a asymmetric) graphs [5, 4].

▶ **Theorem 6.1** (random d-regular graphs are robustly self-ordered). *For any sufficiently large constant d, a random $2d$-regular n-vertex graph is robustly self-ordered with probability $1 - o(1)$.*

Recall that, with very high probability, these graphs are expanders. We mention that the proof of Theorem 4.1 actually established that n-vertex graphs drawn from a weird distribution (which has min-entropy $\Omega(n)$) are robustly self-ordered with probability $1 - o(1)$. However, this is established by using the edge-coloring variant, and requires employing the transformation presented in Section 2.1. In contrast, the following proof works directly with the original (uncolored) variant, and is completely self-contained.

Proof. The proof is quite similar to the proof Claim 4.1.1, but it faces complications that were avoided in the prior proof by using edge-colors and implicitly directed edges. Specifically, for candidate permutations $\pi_1, ..., \pi_d : [n] \to [n]$ (to be used in the construction) and all (non-trivial) permutations $\mu : [n] \to [n]$, the proof of Claim 4.1.1 considered events of the form $(\forall j \in [d])\ \pi_j(i) = \mu(\pi_j(\mu^{-1}(i)))$, whereas here we shall consider events of the form $\{\pi_j^b(i) : j \in [d] \,\&\, b \in \{\pm 1\}\} = \{\mu(\pi_j^b(\mu^{-1}(i))) : j \in [d] \,\&\, b \in \{\pm 1\}\}$. These multi-set equalities will be reduced to equalities among sequences by considering all possible ordering of these multi-sets. This amounts to taking a union bound over all possible ordering and results in a more complicated analysis (due to the π_j^{-1}'s) and much more cumbersome notation.

To facilitate the proof, we use the standard methodology (cf. [13, Apdx. 2]) of first proving the result in the *random permutation model*, then transporting it to the *configuration model* (by using a general result of [24]), and finally conditioning on the event that the generated graph is simple (which occurs with positive constant probability). Indeed, both models generate multi-graphs that are not necessarily simple graphs (i.e., these multi-graphs may have self-loops and parallel edges). We also use the fact that the simple graphs that are generated by the configuration model (for degree d') are uniformly distributed among all d'-regular graphs.

Recall that in the random permutation model a $2d$-regular n-vertex multi-graph is generated by selecting uniformly and independently d permutations $\pi_1, ..., \pi_d : [n] \to [n]$. The multi-graph, denoted $G_{(\pi_1,...,\pi_d)}$, consists of the edge multi-set $\bigcup_{j \in [d]} \{\{i, \pi_j(i)\} : i \in [n]\}$, where the $2j^{\text{th}}$ (resp., $(2j-1)^{\text{st}}$) neighbor of vertex i is $\pi_j(i)$ (resp., $\pi_j^{-1}(i)$). Note that this multi-graph may have self-loops (due to $\pi_j(i) = i$), which contributed two units to the degree of a vertex, as well as parallel edges (due to $\pi_j(i) = \pi_k(i)$ for $j \neq k$ and $\pi_j(i) = \pi_k^{-1}(i)$ for any j, k). We denote the j^{th} neighbor of vertex i by $g_j(i)$; that is, $g_j(i) = \pi_{j/2}(i)$ if j is even, and $g_j(i) = \pi^{-1}_{(j+1)/2}(i)$ otherwise.

Consider an arbitrary permutation $\mu : [n] \to [n]$, and let $T = \{i \in [n] : \mu(i) \neq i\}$ be its set of non-fixed-point. We shall show that, with probability $1 - \exp(-\Omega(d \cdot |T| \cdot \log n))$ over the choice of $\overline{\pi} = (\pi_1, ..., \pi_d)$, the size of the symmetric difference between $G_{\overline{\pi}}$ and $\mu(G_{\overline{\pi}})$ is $\Omega(|T|)$. Note that this difference is (half) the sum over $i \in [n]$ of the size of the symmetric difference between the multi-set of neighbors of vertex i in $G_{\overline{\pi}}$ and the multi-set of neighbors of vertex i in $\mu(G_{\overline{\pi}})$. We refer to the latter difference by the phrase *the contribution of vertex i to the difference between $G_{\overline{\pi}}$ and $\mu(G_{\overline{\pi}})$*.

As a warm-up, we first show that each element of T contributes a non-zero number of units to the difference (between $G_{\overline{\pi}}$ and $\mu(G_{\overline{\pi}})$) with probability $1 - O(\text{poly}(d)/n)^{d/3}$ over the choice of $\overline{\pi}$. Consider the event that *for some $j, k \in [2d]$, the j^{th} neighbor of $i \in [n]$ in $\mu(G_{\overline{\pi}})$ is different from the k^{th} neighbor of i in $G_{\overline{\pi}}$*. Note that x is the j^{th} neighbor of i in $\mu(G_{\overline{\pi}})$ if and only if $\mu^{-1}(x)$ is the k^{th} neighbor of $\mu^{-1}(i)$ in $G_{\overline{\pi}}$, which holds if and only if $\mu^{-1}(x) = g_k(\mu^{-1}(i))$ (equiv., $x = \mu(g_k(\mu^{-1}(i)))$). Recalling that $i \in T$ contributes to the difference (between $G_{\overline{\pi}}$ and $\mu(G_{\overline{\pi}})$) if the multi-sets of its neighbors in $G_{\overline{\pi}}$ and $\mu(G_{\overline{\pi}})$ differ, it follows that $i \in T$ contributes to the difference if and only if *for every permutation $\sigma : [2d] \to [2d]$ there exists $j \in [2d]$ such that $g_j(i) \neq \mu(g_{\sigma(j)}(\mu^{-1}(i)))$*. Thus, the probability of the complementary event (i.e., i does not contribute to the difference) is given by

$$\begin{aligned} &\Pr_{\overline{\pi}}\left[\exists \sigma \in \text{Sym}_{2d}\ (\forall j \in [2d])\ \ g_j(i) = \mu(g_{\sigma(j)}(\mu^{-1}(i)))\right] \\ &\quad = \ \ (2d)! \cdot \max_{\sigma \in \text{Sym}_{2d}} \left\{\Pr_{\overline{\pi}}\left[(\forall j \in [2d])\ \ g_j(i) = \mu(g_{\sigma(j)}(\mu^{-1}(i)))\right]\right\}. \end{aligned} \tag{10}$$

Fixing σ that maximizes the probability, and denoting it σ_i, consider any $J_i \subseteq [d]$ such that for the j's in J_i the multi-sets $\{j, \lceil \sigma_i(2j)/2 \rceil\}$'s are disjoint (i.e., $\{j, \lceil \sigma_i(2j)/2 \rceil\} \cap \{k, \lceil \sigma_i(2k)/2 \rceil\} = \emptyset$ for any $j \neq k \in J_i$). Note that we may select J_i such that $|J_i| \geq d/3$, since taking j to J_i only rules out taking (to J_i) any k such that $\lceil \sigma_i(2k)/2 \rceil = v \overset{\text{def}}{=} \lceil \sigma_i(2j)/2 \rceil$ (equiv., k such that $\sigma_i(2k) \in \{2v-1, 2v\}$). Using this proerty of J_i, we prove –

▷ Claim 6.1.1 (warm-up). [33] (10) is upper-bounded by $(2d)^{2d} \cdot (2/n)^{|J_i|}$.

Proof. We upper-bound (10) by

$$\begin{aligned} &(2d)! \cdot \max_{\sigma} \left\{\Pr_{\overline{\pi}}\left[(\forall j \in J_i)\ \ g_{2j}(i) = \mu(g_{\sigma(2j)}(\mu^{-1}(i)))\right]\right\} \\ &\quad = \ \ (2d)! \cdot \prod_{j \in J_i} \Pr_{\pi_j, \pi_{\lceil \sigma_i(2j)/2 \rceil}}\left[g_{2j}(i) = \mu(g_{\sigma_i(2j)}(\mu^{-1}(i)))\right] \end{aligned} \tag{11}$$

where the equality uses the disjointness of the multi-sets $\{j, \lceil \sigma_i(2j)/2 \rceil\}$ for the j's in J_i. Next, we upper-bound (11) by

$$(2d)! \cdot \prod_{j \in J_i} \Pr_{\pi_j, \pi_{\lceil \sigma_i(2j)/2 \rceil}}\left[\pi_j(i) = \mu(\pi_{\lceil \sigma_i(2j)/2 \rceil}^{(-1)^{\sigma_i(2j) \bmod 2}}(\mu^{-1}(i)))\right] \ < \ (2d)^{2d} \cdot (2/n)^{|J_i|}, \tag{12}$$

where $\Pr_{\pi_j, \pi_j}[\cdot]$ stands for $\Pr_{\pi_j}[\cdot]$ and π^1 stands for π, while the inequality is justified by considering the following three cases (w.r.t each $j \in J_i$).

[33] One may obtain a better bound of $O(d/n)^{2d}$ by analyzing (10) directly, by considering all the $2d$ events and accounting for their small dependency. On the other hand, we can obtain higher robustness parameter by considering smaller sets J_i's (say of size $d/4$), which suffice for counting vertices that contribute (say) $d/4$ units to the difference between $G_{\overline{\pi}}$ and $\mu(G_{\overline{\pi}})$.

1. If $k \overset{\text{def}}{=} \lceil \sigma_i(2j)/2 \rceil \neq j$, then, letting $b = (-1)^{\sigma_i(2j) \bmod 2}$, the corresponding factor in the l.h.s of (12) is

$$\Pr_{\pi_j,\pi_k}\left[\pi_j(i) = \mu(\pi_k^b(\mu^{-1}(i)))\right]$$

which equals $1/n$ by fixing π_k, letting $v = \mu(\pi_k^b(\mu^{-1}(i)))$, and using $\Pr_{\pi_j}[\pi_j(i)=v] = 1/n$.

2. If $\sigma_i(2j) = 2j$, then the corresponding factor in the l.h.s of (12) is

$$\Pr_{\pi_j}\left[\pi_j(i) = \mu(\pi_j(\mu^{-1}(i)))\right]$$

which is at most $1/(n-1)$ since $\mu(i) \neq i$; specifically, fixing the value of $\pi_j(\mu^{-1}(i))$, and denoting this value by v, leaves $\pi_j(i)$ uniformly distributed in $[n] \setminus \{v\}$, which means that $\Pr_{\pi_j}[\pi_j(i)=\mu(v)|v = \pi_j(\mu^{-1}(i))] \leq 1/(n-1)$ (where equality holds if $\mu(v) \neq v$).

3. If $\sigma_i(2j) = 2j-1$, then the corresponding factor in the l.h.s of (12) is

$$\Pr_{\pi_j}\left[\pi_j(i) = \mu(\pi_j^{-1}(\mu^{-1}(i)))\right]$$

which is less than $2/n$. In this case, we consider two sub-cases depending on whether or not $\pi_j(i) = \mu^{-1}(i)$, while noting that the first case occurs with probability $1/n$ whereas $\Pr_{\pi_j}[\pi_j(i) = \mu(\pi_j^{-1}(\mu^{-1}(i)))|\pi_j(i) \neq \mu^{-1}(i)] \leq 1/(n-1)$.

Hence, each of the factors in the l.h.s of (12) is upper-bounded by $2/n$, and the claim follows. ◁

The general case. The same argument generalizes to a set $I \subseteq T$ such that $I \cap \mu(I) = \emptyset$. In such a case we get

$$\begin{aligned}
&\Pr_{\overline{\pi}}\left[(\forall i \in I)\, (\exists \sigma_i \in \mathrm{Sym}_{2d})\, (\forall j \in [2d])\;\; g_j(i) = \mu(g_{\sigma_i(j)}(\mu^{-1}(i)))\right] \\
&\quad = \;\; (2d)!^{|I|} \cdot \max_{\sigma_1,\dots,\sigma_n}\left\{\Pr_{\overline{\pi}}\left[(\forall i \in I)\, (\forall j \in [2d])\;\; g_j(i) = \mu(g_{\sigma_i(j)}(\mu^{-1}(i)))\right]\right\} \qquad (13)
\end{aligned}$$

▷ Claim 6.1.2 (actual analysis). (13) is upper-bounded by

$$(2d)^{2d\cdot|I|} \cdot (2/(n-2(|I|-1)))^{|I|\cdot d/3}. \qquad (14)$$

Proof. For every $i \in I = \{i_1, ..., i_m\}$, we fixed a set J_i of size at least $d/3$ such that the multi-sets $\{j, \lceil \sigma_i(2j)/2 \rceil\}$'s are disjoint, and upper-bound (13) by

$$\begin{aligned}
&(2d)!^m \cdot \prod_{k\in[m]} \prod_{j\in J_{i_k}} \Pr_{\pi_1,\dots,\pi_{2d}}\left[g_{2j}(i_k) = \mu(g_{\sigma_{i_k}(2j)}(\mu^{-1}(i_k)))\,|E_{j,k}(\pi_1,....,\pi_{2d})\right] \\
&\quad = \;\; (2d)!^m \cdot \prod_{k\in[m]} \prod_{j\in J_{i_k}} \Pr_{\pi_1,\dots,\pi_{2d}}\left[\pi_j(i_k) = \mu(\pi_{\sigma'_{i_k}(2j)}^{\sigma''_{i_k}(2j)}(\mu^{-1}(i_k)))\,|E_{j,k}(\pi_1,....,\pi_{2d})\right] \qquad (15)
\end{aligned}$$

where $\sigma'_i(2j) \overset{\text{def}}{=} \lceil \sigma_i(2j)/2 \rceil$, and $\sigma''_i(2j) \overset{\text{def}}{=} (-1)^{\sigma_i(2j) \bmod 2}$, whereas $E_{j,k}(\pi_1, ..., \pi_{2d})$ is an event that depends only on the value of π_j and $\pi_{\sigma'_{i_k}(2j)}^{\sigma''_{i_k}(2j)}$ on the points $i_1, ..., i_{k-1}$ and $\mu^{-1}(i_1), ..., \mu^{-1}(i_{k-1})$, respectively. Specifically, $E_{j,k}(\pi_1, ..., \pi_{2d})$ is the event

$$(\forall k' \in [k-1])\;\; g_{2j}(i_{k'}) = \mu(g_{\sigma_{i_{k'}}(2j)}(\mu^{-1}(i_{k'})))$$

which can be written as

$$(\forall k' \in [k-1])\;\; \pi_j(i_{k'}) = \mu(\pi_{\sigma'_{i_{k'}}(2j)}^{\sigma''_{i_{k'}}(2j)}(\mu^{-1}(i_{k'}))).$$

Now, when analyzing the foregoing conditional probability in (15), we consider two cases. If $j \neq \sigma'_{i_k}(2j)$, then we fix the value of each of these two permutations (i.e., π_j and $\pi_{\sigma'_{i_k}(2j)}$) on the corresponding $k-1$ points that occur in the condition $E_{j,k}$, and the value of these permutations on the k^{th} points (i.e., i_k and $\mu^{-1}(i_k)$) is restricted accordingly (i.e., to the remaining $n-(k-1)$ values). Otherwise (i.e., $j = \sigma'_{i_k}(2j)$), we fix the value of π_j on these $2(k-1)$ points. Hence, the argument in the warm-up analysis applies with n replaces by either $n-(k-1)$ or $n-2(k-1)$. It follows that (15) is upper-bounded by

$$(2d)!^m \cdot \prod_{k\in[m]} (2/(2-2(m-1)))^{|J_{i_k}|}.$$

Using $|J_{i_k}| \geq d/3$ for every $k \in [m]$, the claim follows. $\triangleleft$

Recall that (14) refers to a fixed set $I \subseteq T$ such that $I \cap \mu(I) = \emptyset$, and that it constitutes an upper bound on the probability (over the choice of $\overline{\pi}$) that, for each $i \in I$ there exists a permutation $\sigma_i : [2d] \to [2d]$ such that $g_j(i) = \mu(g_{\sigma_i(j)}(\mu^{-1}(i)))$ holds for all $j \in [2d]$. This upper bound (i.e., $(2d)^{2d\cdot|I|} \cdot (2/(n-2(|I|-1)))^{|I|\cdot d/3}$) simplifies to $(2d)^{2d\cdot|I|} \cdot (6/n)^{|I|\cdot d/3}$, provided that $|I| \leq n/3$.

Recalling that $t \overset{\text{def}}{=} |T| \in [n]$, we shall upper-bound the probability (over the choice of $\overline{\pi}$) that T contains a $\lceil t/2 \rceil$-subset T' such that for each $i \in T'$ there exists a permutation $\sigma_i : [2d] \to [2d]$ such that $g_j(i) = \mu(g_{\sigma_i(j)}(\mu^{-1}(i)))$ holds for all $j \in [2d]$. We do so by taking a union bound over all $\lceil t/6 \rceil$-subsets I such that $I \cap \mu(I) = \emptyset$ and for each $i \in I$ there exists a permutation $\sigma_i : [2d] \to [2d]$ such that $g_j(i) = \mu(g_{\sigma_i(j)}(\mu^{-1}(i)))$ holds for all $j \in [2d]$. (Note that such a $\lceil t/6 \rceil$-subset I exists in each $\lceil t/2 \rceil$-subset T', and that $\lceil t/6 \rceil < n/3$.) Using the aforementioned simplified form of (14), we conclude that, with probability at most

$$\binom{t}{\lceil t/6 \rceil} \cdot (2d)^{2d\cdot\lceil t/6 \rceil} \cdot (6/n)^{\lceil t/6 \rceil \cdot d/3} \;<\; 2^t \cdot (6 \cdot (2d)^6/n)^{\lceil t/6 \rceil \cdot d/3} \;=\; \exp(-\Omega(dt \log n))$$

over the choice of $\overline{\pi}$, the set T contains no $\lceil t/6 \rceil$-subset I as above. This means that, with probability at most $\exp(-\Omega(dt \log n))$, less than $t/2$ of the indices $i \in T$ contribute a non-zero number of units to the difference (between $G_{\overline{\pi}}$ and $\mu(G_{\overline{\pi}})$).

Letting $c' = 1/2$ and considering all (non-trivial) permutations $\mu : [n] \to [n]$, we conclude that the probability, over the choice of $\overline{\pi}$, that $G_{\overline{\pi}}$ is not c'-robustly self-ordered is at most

$$\begin{aligned}\sum_{t\in[n]} \binom{n}{t} \cdot \exp(-\Omega(dt \log n)) \;&=\; \sum_{t\in[n]} \exp(-\Omega((d-O(1)) \cdot t \log n)) \\ &=\; \exp(-\Omega((d-O(1)) \cdot \log n)),\end{aligned}$$

and the claim follows for the permutation model (and for any sufficiently large d).

As stated upfront, using the general result of [24, Thm. 1.3], we infer that a uniformly distributed $2d$-regular n-vertex multi-graph fails to be c'-robustly self-ordered with probability $o(1)$. Lastly, recalling that such a $2d$-regular multi-graph is actually a simple graph with probability $\exp(-((2d)^2-1)/4)$, the theorem follows. $\blacktriangleleft$

Digest

The proof of Theorem 6.1 is quite similar to the proof Claim 4.1.1, but it faces two complications that were avoided in the prior proof (by using edge-colors and implicitly directed edges). Most importantly, the current proof has to handle equality between multi-sets instead of equality between sequences. This is done by considering all possible ordering of these multi-sets, which amounts to taking a union bound over all possible ordering and results in more complicated analysis and notation. (Specifically, see the introduction of σ_i's and J_i's and the three cases analyzed in the warm-up.) In addition, since edges are defined by permutations over the vertex-set rather than by perfect matching, we have to consider both the forward and backward direction of each permutation, which results in further complicating the analysis and the notation. (Specifically, see the introduction of σ_i''s and σ_i'''s and the three cases analyzed in the warm-up.)

An alternative proof of Theorem 4.2

We mention that combining an extension of Theorem 6.1 with some of the ideas underlying the proof of Theorem 4.2 yields an alternative proof of Theorem 4.2 (i.e., an alternative construction of robustly self-ordered bounded-degree graphs).

▶ Remark 6.2 (an alternative construction of d-regular robustly self-ordered graphs). On input 1^n, we set $\ell = \frac{O(\log n)}{\log\log n}$, and proceeds in three steps.

1. Extending the proof of Theorem 6.1, we show that for all sufficiently large constant d, for any set $\mathcal{G}$ of $t = t(\ell) < n = \ell^{\Omega(\ell)}$ ($2d$-regular) ℓ-vertex graphs, with probability $1 - o(1)$, a random $2d$-regular ℓ-vertex graph is both robustly self-ordered and far from being isomorphic to any graph in $\mathcal{G}$. Note that, with probability $1 - o(1)$, such a graph is also expanding.

   Here two ℓ-vertex graphs are said to be far apart if they disagree on $\Omega(\ell)$ vertex-pairs.

   The proof of Theorem 6.1 is extended by considering, for a random graph, the event that it is either not robustly self-ordered or is not far from an isomorphic copy of one of the t (fixed) graphs. The later event *(i.e., being close to isomorphic to one of these graphs)* occurs with probability $o(t/n)$.
2. Relying on Step 1, we find a sequence of n/ℓ robustly self-ordered $2d$-regular ℓ-vertex graphs that are expanding and pairwise far from being isomorphic to one another.

   This is done by iteratively finding robustly self-ordered $2d$-regular ℓ-vertex expanding graphs that are far from being isomorphic to all prior ones, where scanning all possible graphs and checking the condition can be done in time $n \cdot \ell^{d\ell/2} \cdot (\ell!) = \mathrm{poly}(n)$.
3. Using the sequence of n/ℓ graphs found in Step 2, we consider the n-vertex graph that consists of these ℓ-vertex graphs as its connected components, and use parts of the proof of Theorem 4.2 to show that this graph is robustly self-ordered. Specifically, we only need to consider cases that are analogous to Cases 2, 6 and 7. The treatment of the analogous cases is slightly simpler than in the proof of Theorem 4.2, since the graphs are somewhat simpler.

Note that the resulting graphs are not locally constructable.

Part II
The Case of Dense Graphs

Recall that when considering graphs of unbounded degree, we ask whether we can obtain unbounded robustness parameters. In particular, we are interested in n-vertex graphs that are $\Omega(n)$-robustly self-ordered, which means that they must have $\Omega(n^2)$ edges.

In Section 7 we prove the existence of $\Omega(n)$-robustly self-ordered n-vertex graphs, and show that they imply $\Omega(1)$-robustly self-ordered bounded-degree $O(n^2)$-vertex graphs. In Section 8, we reduce the construction of the former (dense) n-vertex graphs to the construction of non-malleable two-source extractors (with very mild parameters). We actually show two reductions: The first reduction (presented in Section 8.1) requires the extractors to have an additional natural feature, called quasi-orthogonality, and yields a construction of such n-vertex graphs that runs in $\text{poly}(n)$-time. The second reduction (presented in Section 8.2) does not make this requirement, and yields an algorithm that computes the adjacency predicate of such n-vertex graphs in $\text{poly}(\log n)$-time.

In Section 9 we demonstrate the applicability of $\Omega(n)$-robustly self-ordered n-vertex graphs to property testing; specifically, to proving lower bounds (on the query complexity) for the dense graph testing model. Lastly, in Section 10, we consider the construction of $\Omega(d(n))$-robustly self-ordered n-vertex graphs of maximum degree $d(n)$, for every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \in [\Omega(1), n]$.

7 Existence and Transformation to Bounded-Degree Graphs

It seems easier to prove that random n-vertex graphs are $\Omega(n)$-robustly self-ordered (see Proposition 7.1) than to prove that random bounded-degree graphs are $\Omega(1)$-robustly self-ordered (or even just prove that such bounded-degree graphs exist). In contrast, it seems harder to construct $\Omega(n)$-robustly self-ordered n-vertex graphs than to construct $\Omega(1)$-robustly self-ordered bounded-degree graphs. In particular, we show that $\Omega(n)$-robustly self-ordered n-vertex graphs can be easily transformed into $O(n^2)$-vertex bounded-degree graphs that are $\Omega(1)$-robustly self-ordered (see Proposition 7.2). We stress that the construction of robustly self-ordered bounded-degree graphs that is obtained by combining the foregoing transformation with Theorem 1.4 is entirely different from the constructions presented in the first part of the paper.

Random graphs are robustly self-ordered

We first show that, with very high probability, a random n-vertex graph $G_n = ([n], E_n)$, where E_n is a uniformly distributed subset of $\binom{[n]}{2}$, is $\Omega(n)$-robustly self-ordered.

▶ **Proposition 7.1** (robustness analysis of a random graph). *A random n-vertex graph $G_n = ([n], E_n)$ is $\Omega(n)$-robustly self-ordered with probability $1 - \exp(-\Omega(n))$.*

As stated above, the following proof is significantly easier than the proof provided for the bounded-degree analogue (i.e., Theorem 6.1).

Proof. For each (non-trivial) permutation $\mu : [n] \to [n]$, letting $T \overset{\text{def}}{=} \{i \in [n] : \mu(i) \neq i\}$ denote its (non-empty) set of non-fixed-points, we show that, with probability $1 - \exp(-\Omega(n \cdot |T|))$, the size of the symmetric different between a random n-vertex graph $G_n = ([n], E_n)$ and $\mu(G_n)$ is $\Omega(n \cdot |T|)$.

For every $u, v \in [n]$ such that $u < v$, let $\chi_{u,v} = \chi^{\mu}_{u,v}(G_n)$ represent the event that *the pair* $(\mu(u), \mu(v))$ *contributes to the symmetric difference between* G_n *and* $\mu(G_n)$; that is, $\chi_{u,v} = 1$ if exactly one of the edges $\{\mu(u), \mu(v)\}$ and $\{u, v\}$ is in G_n, since $\{u, v\}$ is an edge of G_n if and only if $\{\mu(u), \mu(v)\}$ is an edge of $\mu(G_n)$. We shall prove that

$$\mathrm{Pr}_{G_n}\left[\sum_{u<v\in[n]} \chi^{\mu}_{u,v}(G_n) \;<\; \frac{n\cdot|T|}{20}\right] \;=\; \exp(-\Omega(n\cdot|T|)). \tag{16}$$

We prove (16) by using a $\lceil |T|/3 \rceil$-subset $I \subseteq T$ such that $I \cap \mu(I) = \emptyset$. Let $T' = T \setminus (I \cup \mu^{-1}(I))$, which implies $T' \cap I = \emptyset$ and $\mu(T') \cap I = \emptyset$. Let $J = ([n] \setminus T) \cup T'$, and note that $|J| = n - |T| + (|T| - 2 \cdot \lceil |T|/3 \rceil) \geq n - (2|T|/3) - 2 \geq (n/3) - 2$. Observe that, for every $(u, v) \in J \times I$, it holds that $u \neq v$ and $\Pr[\chi_{u,v} = 1] = 1/2$, where the equality is due to $\{u, v\} \neq \{\mu(u), \mu(v)\}$, which holds since $(u, v) \in J \times I$ but $\mu(u), \mu(v) \in [n] \setminus I$. Furthermore, the events the correspond to the pairs in $J \times I$ are independent, because the sets $\{\{u, v\} : (u, v) \in J \times I\}$ and $\{\{\mu(u), \mu(v)\} : (u, v) \in J \times I\}$ are disjoint; that is, $(u, v) \in J \times I$ implies $(\mu(u), \mu(v)) \in ([n] \setminus I) \times ([n] \setminus I)$. Hence (using $n \leq 3(|J| + 2)$ and $|T| \leq 3|I|$ (as well as $3(|J| + 2) \cdot 3|I| < 9.9 \cdot |J| \cdot |I|$)), the l.h.s. of (16) is upper-bounded by

$$\begin{aligned}\mathrm{Pr}_{G_n}\left[\sum_{(u,v)\in J\times I} \chi^{\mu}_{u,v}(G_n) \;<\; \frac{3(|J|+2)\cdot 3|I|}{20}\right] &\leq \mathrm{Pr}_{G_n}\left[\sum_{(u,v)\in J\times I} \chi^{\mu}_{u,v}(G_n) \;<\; \frac{0.99\cdot|J|\cdot|I|}{2}\right] \\ &= \exp(-\Omega(|J|\cdot|I|))\end{aligned}$$

which is $\exp(-\Omega(n \cdot |T|))$. Having established (16), the claim follows by a union bound (over all non-trivial permutations $\mu : [n] \to [n]$); specifically, denoting the set of non-trivial permutations by P_n, we upper-bound the probability that G_n is not $\frac{n}{20}$-robust by

$$\begin{aligned}&\sum_{\mu\in P_n} \mathrm{Pr}_{G_n}\left[\mu \text{ violates the condition in (16)}\right] \\ &\qquad \leq \quad \sum_{t\in[n]} \binom{n}{t} \cdot (t!) \cdot \exp(-\Omega(n\cdot t)) \\ &\qquad < \quad n \cdot \max_{t\in[n]}\{n^t \cdot \exp(-\Omega(n\cdot t))\} \\ &\qquad = \quad \exp(-\Omega(n))\end{aligned}$$

where t represents the size of the set of non-fixed-points (w.r.t μ). ◀

Obtaining bounded-degree robustly self-ordered graphs

We next show how to transform $\Omega(n)$-robustly self-ordered n-vertex graphs to $O(n^2)$-vertex bounded-degree graphs that are $\Omega(1)$-robustly self-ordered. Essentially, we show that the standard "degree reduction via expanders" technique works (when using a different color for the expanders' edges, and then using gadgets to replace colored edges). Specifically, we replace each vertex in $G_n = ([n], E_n)$ by an $(n-1)$-vertex expander graph and connect each of these vertices to at most one vertex in a different expander, while coloring the edges of the expanders with 1, and coloring the other edges by 2. Actually, the vertex v is replaced by the vertex-set $C_v = \{\langle v, u\rangle : u \in [n] \setminus \{v\}\}$ and in addition to the edges of the expander, colored 1, we connect each vertex $\langle v, u\rangle \in C_v$ to the vertex $\langle u, v\rangle \in C_u$ and color this edge 2 if

$\{u,v\} \in E_n$ and 0 otherwise.[34] This yields an $n \cdot (n-1)$-vertex $O(1)$-regular graph, denoted G'_n, coupled with an edge-coloring, denoted χ', which uses three colors. Using the hypothesis that G_n is $\Omega(n)$-robustly self-ordered, we prove that (G'_n, χ') is $\Omega(1)$-robustly self-ordered (in the colored sense).

▶ **Proposition 7.2** (robustness analysis of the degree reduction). *If G_n is $\Omega(n)$-robustly self-ordered, then (G'_n, χ') is $\Omega(1)$-robustly self-ordered* (in the colored sense of Definition 2.1).

Using Theorem 2.4 (after adding self-loops), we obtain a $O(1)$-regular $O(n^2)$-vertex graph that is $\Omega(1)$-robustly self-ordered (in the standard sense).

Proof. Denoting the vertex-set of G'_n by $V = \bigcup_{v \in [n]} C_v$, we consider an arbitrary (non-trivial) permutation $\mu' : V \to V$, and the corresponding set of non-fixed-points T'. Intuitively, if μ' maps vertices of C_v to several C_w's, then we get a proportional contribution to the difference between G'_n and $\mu'(G'_n)$ by the (1-colored) edges of the expander. Otherwise, μ' induces a permutation μ over the vertices of G_n, and we get a corresponding contribution via the (2-colored) edges of G_n. Lastly, non-identity mapping inside the individual C_v's are charged using the (0-colored and 2-colored) edges that connect different C_v's. Details follow.

For a permutation $\mu' : V \to V$ as above, let $\mu : [n] \to [n]$ be a permutation that maximizes the (average over $v \in [n]$ of the) number of vertices in C_v that are mapped by μ' to vertices in $C_{\mu(v)}$; that is, for every permutation $\nu : [n] \to [n]$, it holds that

$$\left|\left\{\langle v,u\rangle \in V : \mu'(\langle v,u\rangle) \in C_{\mu(v)}\right\}\right| \;\geq\; \left|\left\{\langle v,u\rangle \in V : \mu'(\langle v,u\rangle) \in C_{\nu(v)}\right\}\right|. \tag{17}$$

We consider the following three cases.

Case 1: $\sum_{v\in[n]} |B_v| = \Omega(|T'|)$, where $B_v \overset{\text{def}}{=} \{\langle v,u\rangle \in C_v : \mu'(\langle v,u\rangle) \notin C_{\mu(v)}\}$.

(This refers to the case that many vertices are mapped by μ' to an expander that is different from the one designated by μ, which represents the best possible mapping of whole expanders.)

Letting $C_{v,w} \overset{\text{def}}{=} \{\langle v,u\rangle : \mu'(\langle v,u\rangle) \in C_w\}$, we first observe that for every v it holds that $\max_{w \neq \mu(v)}\{|C_{v,w}|\} \leq \frac{2}{3} \cdot (n-1)$, because otherwise we reach a contradiction to the maximality of μ by defining $\nu(v) = w$ and $\nu(\mu^{-1}(w)) = \mu(v)$, where w is the element obtaining the maximum, and $\nu(x) = \mu(x)$ otherwise.

Next, observe that there exists $W_v \subseteq [n] \setminus \{\mu(v)\}$ such that $B'_v = \bigcup_{w \in W_v} C_{v,w}$ satisfies both $|B'_v| \leq \frac{2}{3} \cdot (n-1)$ and $|B'_v| \geq |B_v|/3$. Now, consider the sets B'_v and $C_v \setminus B'_v$: On the one hand, in $\mu'(G'_n)$ there are $\Omega(|B'_v|)$ 1-colored edges connecting $\mu'(B'_v)$ and $\mu'(C_v \setminus B'_v)$, due to the subgraph of $\mu'(G'_n)$ induced by $\mu'(C_v)$ which equals subgraph of G'_n induced by C_v (which, in turn, is an expander). On the other hand, in G'_n there are no 1-colored edges between $\mu'(B'_v)$ and $\mu'(C_v \setminus B'_v)$, since $\mu'(B'_v) \subseteq \bigcup_{w \in W_v} C_w$ and $\mu'(C_v \setminus B'_v) \subseteq \bigcup_{w \in [n] \setminus W_v} C_w$.

We conclude that, in this case, the difference between G'_n and $\mu'(G_n)$ is $\sum_v \Omega(|B'_v|) = \sum_v \Omega(|B_v|) = \Omega(|T'|)$.

Case 2: $\sum_{v \in [n] : \mu(v) \neq v} |C'_v| = \Omega(|T'|)$, where $C'_v \overset{\text{def}}{=} \{\langle v,u\rangle \in C_v : \mu'(\langle v,u\rangle) \in C_{\mu(v)}\}$.

(This refers to the case that many vertices are mapped by μ' to an expander that is designated by μ, but this expander is not the one in which they reside (i.e., μ has many non-fixed-points).)

[34] This is equivalent to first converting G_n into a n-vertex clique while coloring an edge 2 if and only if it is in E_n.

Letting $\gamma > 0$ be a constant such that G_n is $\gamma \cdot n$-robustly self-ordered, we may assume that $\sum_{v\in[n]:\mu(v)\neq v} |C'_v| \geq (1 - 0.5\cdot\gamma)\cdot \sum_{v\in[n]:\mu(v)\neq v} |C_v|$, since otherwise we are done by Case 1.

By the γn-robust self-ordering of G_n, the difference between G_n and $\mu(G_n)$ is at least $\Delta \overset{\text{def}}{=} \gamma n \cdot |\{v \in [n] : \mu(v) \neq v\}|$. Assuming, for a moment, that $\mu'(C_v) = C_v$ for every v such that $\mu(v) \neq v$, the difference between G'_n and $\mu'(G'_n)$ is Δ, where the difference is due to edges colored 2 (i.e., the edges inherited from G_n). This amount is prorotional to the number of vertices in the current case, since

$$\Delta \;=\; \frac{\gamma n}{n-1} \cdot \sum_{v:\mu(v)\neq v} |C_v| \;>\; \gamma \cdot \sum_{v:\mu(v)\neq v} |C_v|.$$

In general, $\mu'(C_v) = C_v$ may not hold for some v, and in this case we may loss the contribution of the 2-colored edges incident at vertices in $\bigcup_{v\in[n]:\mu(v)\neq v}(C_v \setminus C'_v)$. Recalling that (by our hypothesis) the size of this set is at most $0.5\cdot\gamma\cdot\sum_{v:\mu(v)\neq v}|C_v|$, we are left with a contribution of at least $0.5\gamma\cdot\sum_{v:\mu(v)\neq v}|C'_v|$.

We conclude that, in this case, the difference between G'_n and $\mu'(G_n)$ is $\Omega(\sum_{v:\mu(v)\neq v}|C'_v|) = \Omega(|T'|)$.

Case 3: $\sum_{v\in[n]}|C''_v| = \Omega(|T'|)$, where $C''_v \overset{\text{def}}{=} \{\langle v,u\rangle \in C_v : \mu'(\langle v,u\rangle) \in C_v \setminus \{\langle v,u\rangle\}\}$.

(This refers to the case that many vertices are mapped by μ' to a different vertex in the same expander in which they reside.)[35]

(This case would have been easy to handle if the expanders used on the C_v's were robustly self-ordered. Needless to say, we want to avoid such an assumption. Instead, we rely on the fact that in G'_n different vertices in C_v are connected to different C_u's.)

We may assume that $\sum_{v\in[n]}|C''_v| \geq 2\cdot\sum_{v\in[n]}|\{\langle v,u\rangle \in C_v : \mu'(\langle v,u\rangle) \notin C_v\}|$, since otherwise we are done by either Case 1 or Case 2. Now, consider a generic $\langle v,u\rangle \in C''_v$, and let $w \neq u$ be such that $\mu'(\langle v,u\rangle) = \langle v,w\rangle$. Then, in $\mu'(G'_n)$ an edge colored either 0 or 2 connects $\langle v,w\rangle = \mu'(\langle v,u\rangle)$ to $\mu'(\langle u,v\rangle)$, since $\langle v,u\rangle$ and $\langle u,v\rangle$ are so connected in G'_n, whereas in G'_n an (even-colored) edge connects $\langle v,w\rangle$ to $\langle w,v\rangle \in C_w$. We consider two sub-cases.

- If $\mu'(\langle u,v\rangle) \in C_u$, then $\langle v,w\rangle$ contributes to the difference between $\mu'(G'_n)$ and G'_n, because in $\mu'(G'_n)$ vertex $\langle v,w\rangle$ is connected (by its even-colored edge) to a vertex in C_u whereas in G'_n vertex $\langle v,w\rangle$ is connected (by its even-colored edge) to a vertex in C_w.

  (Recall that w is uniquely determined by $\langle v,u\rangle \in C''_n$, since $\mu'(\langle v,u\rangle) = \langle v,w\rangle$, and so this contribution can be charged to $\langle v,u\rangle$.)
- If $\mu'(\langle u,v\rangle) \notin C_u$, then $\langle u,v\rangle$ contributes to the set $\bigcup_{x\in[n]}\{\langle x,y\rangle \in C_x : \mu'(\langle x,y\rangle) \notin C_x\}$, which (by the hypothesis) has size at most $0.5\cdot\sum_{v\in[n]}|C''_v|$

Hence, at least half of $\bigcup_{v\in[n]} C''_v$ appears in the first sub-case, which implies that, in this case, the difference between G'_n and $\mu'(G_n)$ is at least $\frac{1}{2}\cdot\sum_{v\in[n]}|C''_v| = \Omega(|T'|)$.

Hence, the difference between G'_n and $\mu'(G_n)$ is $\Omega(|T'|)$. ◀

[35] Note that if $\langle v,u\rangle \in C_v$ is not mapped by μ' to C_v, then either $\mu'(\langle v,u\rangle) \notin C_{\mu(v)}$ holds (i.e., Case 1) or $\mu'(\langle v,u\rangle) \in C_{\mu(v)}$ such that $\mu(v) \neq v$ (i.e., Case 2). Hence, if $\langle u,v\rangle \in T'$ is not counted in Cases 1 and 2, then it must be counted in Case 3.

8 Relation to Non-Malleable Two-Source Extractors

For $n = 2^\ell$, we reduce the construction of $\Omega(n)$-robustly self-ordered (dense) n-vertex graphs to the construction of non-malleable two-source extractors for $(\ell, \ell - O(1))$-sources. Recall that a random variable X is called an (ℓ, k)-source if X is distributed over $[2^\ell]$ and has min-entropy at least k (i.e., $\Pr[X = i] \leq 2^{-k}$ for every $i \in [2^\ell]$).[36] A function $\mathtt{E} : [2^\ell] \times [2^\ell] \to \{0,1\}^m$ is called a (standard) two-source (k, ϵ)-extractor if, for every two independent (ℓ, k)-sources X and Y, it holds that $\mathtt{E}(X, Y)$ is ϵ-close to the uniform distribution over $\{0,1\}^m$, denoted U_m. Our notion of a non-malleable two-source extractor, presented next, is a restricted case of the notions considered in [8, 7].[37]

▶ **Definition 8.1** (non-malleable two-source extractors). *A function* $\mathtt{nmE} : [2^\ell] \times [2^\ell] \to \{0,1\}^m$ *is called a* non-malleable two-source (k, ϵ)-extractor *if, for every two independent* (ℓ, k)*-sources* X *and* Y*, and for every two functions* $f, g : [2^\ell] \to [2^\ell]$ *that have no fixed-point* (i.e., $f(z) \neq z$ and $g(z) \neq z$ for every $z \in [2^\ell]$)*, it holds that* $(\mathtt{nmE}(X, Y), \mathtt{nmE}(f(X), g(Y)))$ *is* ϵ*-close to* $(U_m, \mathtt{nmE}(f(X), g(Y)))$*; that is,*

$$\frac{1}{2} \cdot \sum_{\alpha,\beta} \left| \Pr[(\mathtt{nmE}(X,Y), \mathtt{nmE}(f(X), g(Y))) = (\alpha, \beta)] - 2^{-m} \cdot \Pr[\mathtt{nmE}(f(X), g(Y)) = \beta] \right| \leq \epsilon. \quad (18)$$

The parameter ϵ *is called the* error *of the extractor.*

We shall be interested in the special case in which f and g are permutations. In this case, the foregoing condition (i.e., (18)) can be replaced by requiring that $(\mathtt{nmE}(X, Y), \mathtt{nmE}(f(X), g(Y)))$ is 2ϵ-close to the uniform distribution over $\{0,1\}^{m+m}$.[38] Furthermore, we shall focus on non-malleable two-source (k, ϵ)-extractors that output a single bit (i.e., $m = 1$), and in this case non-triviality mandates $\epsilon < 0.5$. In general, we view ϵ as a constant, but view ℓ and k as varying (or generic) parameters, and focus on the case of $k = \ell - O(1)$.

Recall that constructions of non-malleable two-source (k, ϵ)-extractors with much better parameters are known [7, Thm. 1]. In particular, these constructions support $k = \ell - \ell^{\Omega(1)}$, negligible error (i.e., $\epsilon = \exp(-\ell^{\Omega(1)})$), and $m = \ell^{\Omega(1)}$. We stress that, as is the norm in the context of randomness extraction, the extracting function is computable in polynomial-time (i.e., in $\mathrm{poly}(\ell)$-time).

We shall show that any non-malleable two-source $(\ell - O(1), 0.49)$-extractor (for sources over $[2^\ell]$) yields $\Omega(2^\ell)$-robustly self-ordered $O(2^\ell)$-vertex graphs. Actually, we shall show two such constructions: The first construction runs in $\mathrm{poly}(2^\ell)$-time, and the second construction provides strong constructability (a.k.a local computability) as claimed in Theorem 1.4. Both constructions use a similar underlying logic, which is more transparent in the first construction.

8.1 The first construction

For the first construction, we need the extractor to satisfy the following natural (and quite minimal) requirement, which we call quasi-orthogonality. We say that an extractor $\mathtt{nmE} : [2^\ell] \times [2^\ell] \to \{0,1\}$ is quasi-orthogonal (with error ϵ) if the following conditions hold:

[36] Indeed, for the sake of simplicity (of our arguments), we do not require that $\ell \in \mathbb{N}$, but rather only that $2^\ell \in \mathbb{N}$; consequently, we consider distributions over $[2^\ell]$ rather than over $\{0,1\}^\ell$.

[37] In particular, in [8, 7] it is only required that one of the two functions $f, g : [2^\ell] \to [2^\ell]$ has no fixed-points. There seems to be no concrete reason to prefer one of these three variants over the others. We mention that Definition 8.1 is strictly weaker than the definition of [8] (even in its simplified form [7, Def. 1.3]; see Appendix).

[38] In this case, $f(X)$ and $g(Y)$ have min-entropy at least k, which implies that $\mathtt{nmE}(f(X), g(Y))$ is ϵ-close to the uniform distribution over $\{0,1\}^m$.

1. *The residual function obtained from* $\mathtt{nmE}$ *by any fixing of one of its two arguments is almost unbiased*: For every $x \in [2^\ell]$ and every $\sigma \in \{0,1\}$ it holds that $|\{y \in [2^\ell] : \mathtt{nmE}(x,y) = \sigma\}| \leq (0.5+\epsilon) \cdot 2^\ell$; ditto for every $y \in [2^\ell]$ and the corresponding set $\{x \in [2^\ell] : \mathtt{nmE}(x,y) = \sigma]\}$.
2. *The residual functions obtained from* $\mathtt{nmE}$ *by any two different fixings of one of its two arguments are almost uncorrelated*: For every $\{x,x'\} \in \binom{[2^\ell]}{2}$ it holds that $|\{y \in [2^\ell] : \mathtt{nmE}(x,y) \neq \mathtt{nmE}(x',y)\}| \geq (0.5-\epsilon) \cdot 2^\ell$; ditto for every $\{y,y'\} \in \binom{[2^\ell]}{2}$ and the corresponding set $\{x \in [2^\ell] : \mathtt{nmE}(x,y) \neq \mathtt{nmE}(x,y')]\}$.

As shown in Proposition 8.2, any non-malleable two-source (k,ϵ)-extractor can be transformed (in poly(2^ℓ)-time) into a quasi-orthogonal one at a small degradation in the parameters (i.e., ϵ increases by an additive term of $O(2^{-(\ell-k)})$ and 2^ℓ decreases by an additive term of $O(2^k)$). Note that poly(2^ℓ)-time is acceptable when one aims at constructing $O(2^\ell)$-vertex graphs; however, aiming at strong/local constructability (as in Theorem 1.4), we shall avoid such a transformation in the second construction (presented in Section 8.2).

▶ **Proposition 8.2** (transforming non-malleable two-source extractors into ones that are quasi-orthogonal)**.** *For every $k \leq \ell - 3$, there exists a* poly(2^ℓ)*-time transformation that given a non-malleable two-source (k,ϵ)-extractor* $\mathtt{nmE} : [2^\ell] \times [2^\ell] \to \{0,1\}$*, returns a non-malleable two-source (k,ϵ')-extractor* $\mathtt{nmE} : [n'] \times [n'] \to \{0,1\}$ *such that $n' \geq 2^\ell - O(2^k)$ and* $\mathtt{nmE}'$ *is quasi-orthogonal with error $\epsilon' = \epsilon + O(2^k/n')$.*

Proof. Essentially, $\mathtt{nmE}'$ is obtained from $\mathtt{nmE}$ by simply discarding inputs that violate the quasi-orthogonality conditions. Letting $n = 2^\ell$, first note that the number of x's that violate the first condition is at most 2^{k+1}, because otherwise we obtain a contradiction to the hypothesis that $\mathtt{nmE}$ is a two-source (k,ϵ)-extractor (by letting X be uniform on the x's that satisfy $|\{y \in [n] : \mathtt{nmE}(x,y) = \sigma\}| > (0.5+\epsilon) \cdot n$ for either $\sigma = 0$ or $\sigma = 1$, and Y be uniform on $\{0,1\}^n$). Next, consider the residual (k,ϵ)-extractor $\mathtt{nmE}_1 : [n_1] \times [n_1] \to \{0,1\}$, where $n_1 \geq n - 2^{k+1}$, obtained by omitting the exceptional x's. Note that $\mathtt{nmE}_1$ satisfies the first quasi-orthogonality condition with respect to the first argument with error ϵ. Doing the same for the second argument yields a residual (k,ϵ)-extractor $\mathtt{nmE}_2 : [n_2] \times [n_2] \to \{0,1\}$, where $n_2 \geq n_1 - 2^{k+1}$ and $\mathtt{nmE}_2$ satisfies the first quasi-orthogonality condition (for both arguments) with error $\epsilon + \frac{2^{k+1}}{n_1}$. Likewise, we claim that there are at most 2^k disjoint pairs $\{x,x'\}$'s that violate the second condition (i.e., $|\{y \in [n_2] : \mathtt{nmE}_2(x,y) \neq \mathtt{nmE}_2(x',y)\}| \geq (0.5-\epsilon) \cdot n_2$), because otherwise we obtain a contradiction to the hypothesis that $\mathtt{nmE}_2$ is a *non-malleable* two-source (k,ϵ)-extractor (by using a function that maps each such x to its matched x'). And, again, we consider a residual extractor obtained by omitting the exceptional pairs. Doing the same for the y's, we obtained the desired extractor. ◀

Recall that non-malleable two-source extractors with much stronger parameters than we need (i.e., min-entropy $\ell - \ell^{\Omega(1)}$, negligible error, and $\ell^{\Omega(1)}$ bits of output), were constructed in [7, Thm. 1], but these extractors are not quasi-orthogonal. Employing Proposition 8.2, we obtain a quasi-orthogonal non-malleable two-source $(\ell-4, 0.1)$-extractor that can be used in the construction of Theorem 8.3. Essentially, the construction consists of a bipartite graph, with 2^ℓ vertices on each side, such that the edges between the two sides are determined by the extractor. In addition, we add a clique on one of the two sides so that the two sides are (robustly) distinguishable (see Fugure 1). We stress that the resulting $2^{\ell+1}$-vertex graph is $\Omega(2^\ell)$-robustly self-ordered as long as the non-malleable extractor is quasi-orthogonal and works for very mild parameters; that is, we only require error that is bounded away from 1/2 with respect to min-entropy $\ell - O(1)$.

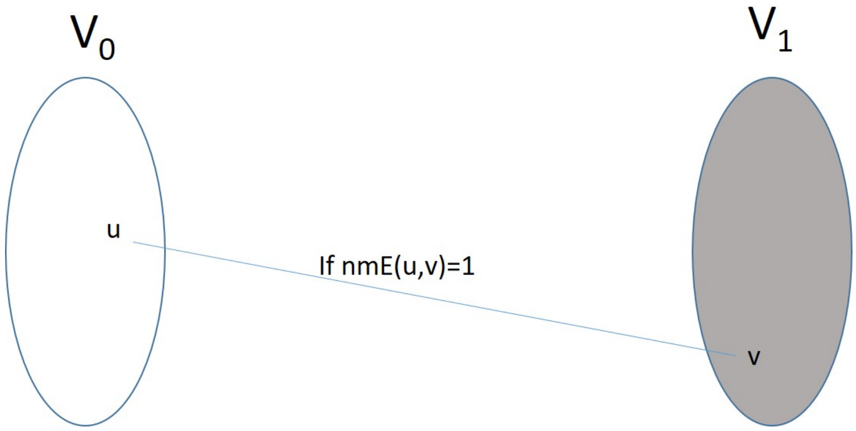


Figure 1 Illustrating the construction of Theorem 8.3.

▶ **Theorem 8.3** (using a quasi-orthogonal non-malleable two-source extractor to obtain a $\Omega(2^\ell)$-robustly self-ordered $O(2^\ell)$-vertex graph)**.** *For a constant $\epsilon \in (0, 0.5)$ varying $\ell \geq k$ such that $k \leq \ell - 2 + \log_2(0.5 - \epsilon) = \ell - O(1)$, suppose that $\mathtt{nmE} : [2^\ell] \times [2^\ell] \to \{0, 1\}$ is a quasi-orthogonal* (with error ϵ) *non-malleable two-source (k, ϵ)-extractor. Then, the $2^{\ell+1}$-vertex graph $G = (V_1 \cup V_0, E)$ such that $V_\sigma = \{\langle\sigma, i\rangle : i \in [2^\ell]\}$ and*

$$E \;=\; \{\{\langle 1, i\rangle, \langle 0, j\rangle\} : \mathtt{nmE}(i, j) = 1\} \cup \binom{V_1}{2} \tag{19}$$

is $\Omega(|V_1 \cup V_0|)$-robustly self-ordered. Furthermore, the claim holds even if the non-malleability condition (i.e., (18)) *holds only for permutations f and g.*

Indeed, the first set of edges, denoted E', corresponds to a bipartite graph between V_1 and V_0 that is determined by $\mathtt{nmE}$, and the second set corresponds to a 2^ℓ-vertex clique. Note that the extraction parameters are extremely weak; that is, the min-entropy may be very high (i.e., $k = \ell - O(1)$), the error may be an arbitrary non-trivial constant (i.e., $\epsilon < 1/2$), and we only extract one bit (i.e., $m = 1$).

Proof. Let $V = V_1 \cup V_0$, and consider an arbitrary (non-trivial) permutation $\mu : V \to V$. Intuitively, if μ maps a vertex of V_1 to V_0, then the difference in degrees of vertices in the two sets (caused by the clique edges) contributes at least $((2^\ell - 1) - 2\epsilon \cdot 2^\ell)/2$ units to the symmetric difference between G and $\mu(G)$, where here we use the first quasi-orthogonality condition. On the other hand, if μ maps $\langle 1, i\rangle \in V_1$ to $V_1 \setminus \{\langle 1, i\rangle\}$, then the difference in the neighborhoods caused by the bipartite graph contributes at least $(0.5 - \epsilon) \cdot 2^\ell/2$ units to the symmetric difference between G and $\mu(G)$. To prove this, we distinguish between the case that μ has relatively few non-fixed-points (in either V_0 or V_1), which is analyzed using the second quasi-orthogonality condition, and the case that μ has relatively many non-fixed-points (in both V_0 and V_1), which is analyzed using the non-malleability condition. Details follow.

Let $T = \{v \in V : \mu(v) \neq v\}$ denote the set of non-fixed-points of μ. Then, we consider two types of vertices: Those that belong to the set $T' = \bigcup_{\sigma \in \{0,1\}} \{v \in V_\sigma : \mu(v) \notin V_\sigma\} \subseteq T$ and those that belong to $T \setminus T'$. The threshold for distinguishing these cases is set to $K = (0.5 - \epsilon) \cdot 2^{\ell-2} = \Omega(|V|)$.

Case 1: $|T'| \geq K$.

(This refers to the case that many vertices are mapped by μ to the opposite side of the bipartite graph (V, E'), where "many" means $\Omega(|V|)$.)

Each vertex in T' contributes $(1 - 2\epsilon) \cdot 2^\ell - 1$ units to the symmetric difference between G and $\mu(G)$, because the degree of each vertex in V_1 is at least $(2^\ell - 1) + (0.5 - \epsilon) \cdot 2^\ell$, whereas the degree of each vertex in V_0 is at most $(0.5 + \epsilon) \cdot 2^\ell$, where we use the first quasi-orthogonality condition, which implies that the number of bipartite edges incident at each vertex is at least $(0.5 - \epsilon) \cdot 2^\ell$ and at most $(0.5 + \epsilon) \cdot 2^\ell$.

Hence, the symmetric difference between G and $\mu(G)$ is at least $((1 - 2\epsilon) \cdot 2^\ell - 1) \cdot |T'| = \Omega(|V|) \cdot |T'|$, since $2^\ell = \Omega(|V|)$. Using the case's hypothesis, we have $|T'| = \Omega(|V|) = \Omega(|T|)$, which means that in this case the difference between G and $\mu(G)$ is $\Omega(|V|) \cdot |T|$. We stress that the difference between G and $\mu(G)$ is at least $\Omega(|V|) \cdot |T'|$ also if the case hypothesis does not hold.

Case 2: $|T'| < K$.

(This refers to the case that few vertices are mapped by μ to the opposite side of the bipartite graph (V, E'), where "few" means less than $K \leq |V|/20$ (assuming $\epsilon \leq 0.1$).)

For every $\sigma \in \{0, 1\}$, let $V'_\sigma = V_\sigma \cap \mu(V_\sigma)$ and $T_\sigma = V'_\sigma \cap T$. Indeed, (T', T_0, T_1) is a three-way partition of T. Note that the size of the symmetric difference between G and $\mu(G)$ is lower-bounded by

$$|\{(v, u) \in V'_1 \times V'_0 : \texttt{nmE}(\mu(v), \mu(u)) \neq \texttt{nmE}(v, u)\}|, \qquad (20)$$

since, for any $(v, u) \in V'_1 \times V'_0$, it holds that $\mu(v)$ neighbors $\mu(u)$ in G if and only if $\texttt{nmE}(\mu(v), \mu(u)) = 1$, whereas $\mu(v)$ neighbors $\mu(u)$ in $\mu(G)$ if and only if v neighbors u in G which holds if and only if $\texttt{nmE}(v, u) = 1$.

We consider two sub-cases according to whether or not $\min(|T_0|, |T_1|)$ is relatively large. The threshold for distinguishing these sub-cases is also set to $K = (0.5 - \epsilon) \cdot 2^{\ell-2}$; note that $K = \Omega(|V|)$ and $K \geq 2^k$.

Case 2.1: $\min(|T_0|, |T_1|) < K$.

In this case we shall use the (second condition of) quasi-orthogonality of nmE.

Suppose, without loss of generality, that $|T_0| \leq |T_1|$, which implies $|T_0| < K$. Then, the contribution of each vertex $v \in T_1$ to (20) equals

$$\begin{aligned}
&|\{u \in V'_0 : \texttt{nmE}(\mu(v), \mu(u)) \neq \texttt{nmE}(v, u)\}| \\
&\quad \geq \quad |\{u \in V'_0 : \texttt{nmE}(\mu(v), u) \neq \texttt{nmE}(v, u)\}| - |T_0| \\
&\quad \geq \quad |\{u \in V_0 : \texttt{nmE}(\mu(v), u) \neq \texttt{nmE}(v, u)\}| - |T'| - |T_0| \\
&\quad \geq \quad (0.5 - \epsilon) \cdot 2^\ell - 2 \cdot K \\
&\quad = \quad (0.5 - \epsilon) \cdot 2^{\ell-1}
\end{aligned}$$

where the first inequality uses $\mu(u) = u$ for $u \in V'_0 \setminus T_0$, the second inequality uses $|V'_0| \geq |V_0| - |T'|$, the third inequality uses $\mu(v) \neq v$ along with the (second condition of) quasi-orthogonality of nmE (and the hypotheses regarding $|T'|$ and $|T_0|$), and the equality is due to $K = (0.5 - \epsilon) \cdot 2^{\ell-2}$.

Hence, in this case, the total contribution to (20) is $(0.5 - \epsilon) \cdot 2^{\ell-1} \cdot |T_1|$, which is $\Omega(|V|) \cdot (|T| - |T'|)$, since $|T_1| \geq (|T| - |T'|)/2$.

Case 2.2: $\min(|T_0|, |T_1|) \geq K$.

In this case we shall use the non-malleable feature of nmE.

Specifically, for each $\sigma \in \{0, 1\}$, let μ_σ denote the restriction of μ to T_σ. Essentially, using $K \geq 2^k$, the non-malleability condition of the (k, ϵ)-extractor nmE implies

$$|\{(i, j) \in T_0 \times T_1 : \texttt{nmE}(i, j) \neq \texttt{nmE}(\mu_0(i), \mu_1(j))\}| \geq (0.5 - \epsilon) \cdot |T_0| \cdot |T_1|.$$

This can be seen by letting X and Y be uniform over T_0 and T_1, respectively, and combining the fact that $\Pr[\mathtt{nmE}(\mu_0(X), \mu_1(Y)) \neq U_1] = 0.5$ with the non-malleability condition (while noting that $\mu_0 : T_0 \to \mu(T_0)$ and $\mu_1 : T_1 \to \mu(T_1)$ have no fixed-points).[39]

Hence, in this case, the total contribution to (20) is $(0.5 - \epsilon) \cdot |T_0| \cdot |T_1| = \Omega(|V|) \cdot (|T| - |T'|)$, where we use $\min(|T_0|, |T_1|) = \Omega(|V|)$.

Hence, in both sub-cases, the difference between G and $\mu(G)$ is $\Omega(|V|) \cdot (|T| - |T'|)$.

Recall that (by the last comment at Case 1) the difference between G and $\mu(G)$ is $\Omega(|V|) \cdot |T'|$. Combining this lower-bound with the conclusion of Case 2, the difference between G and $\mu(G)$ is $\Omega(|V|) \cdot |T|$. ◀

Digest

Note that the quasi-orthogonality of nmE was used in Cases 1 and 2.1, whereas the non-malleability of nmE (w.r.t derangements) was used in Case 2.2. In particular, Case 1 only uses the first condition of quasi-orthogonality, and does so in order to infer that the degrees of all vertices in the bipartite graph are approximately equal. In Case 2.1 the second quasi-orthogonality condition is used in order to assert that the neighborhoods of two different vertices in V_σ are significantly different. This is useful only when the number of non-fixed-points in $V_{1-\sigma}$ is relatively small. When the number of non-fixed-points is large but no vertex is mapped to the other side (i.e., $T' = \emptyset$), we only use Case 2.2, which does not refer to quasi-orthogonality at all. Hence, we have the following –

▶ Remark 8.4 (a special case of Theorem 8.3). For bipartite graphs $G = (V, E)$ such that $V = V_0 \cup V_1$ and $E \subseteq V_0 \times V_1$, we consider the special case of robust self-ordering that refers only to permutations $\mu : V \to V$ that are derangements that preserve the bipartition of V (i.e., μ has no fixed-points and $\mu(V_0) = V_0$).[40] In this case, assuming (only) that nmE is a non-malleable two-source (ℓ, ϵ)-extractor (i.e., the case of $k = \ell$), implies that, for any such μ, the size of the symmetric difference between G and $\mu(G)$ is $(0.5 \pm \epsilon) \cdot |V_0| \cdot |V_1|$. In particular, the quasi-orthogonality condition is not necessary, the proof of Theorem 8.3 simplifies, since $T' = \emptyset$ and $T_\sigma = V_\sigma = V'_\sigma$ hold, and the size of the symmetric difference between G and $\mu(G)$ equal the quantity in (20).

Interestingly, the special case of Theorem 8.3 asserted in Remark 8.4 can be reversed in the sense that a bipartite graph that is robustly self-ordered in the foregoing restricted sense is actually a non-malleable two-source $(\ell, 0.5 - \Omega(1))$-extractor.

▶ **Proposition 8.5** (a reversal of the special case of Theorem 8.3 (i.e., of Remark 8.4)). *Let $G = (V_0 \cup V_1, E)$ be a bipartite graph such that $|V_0| = |V_1|$ and $E \subseteq V_0 \times V_1$. Let $V = V_0 \cup V_1$, and suppose that for every derangement $\mu : V \to V$ such that $\mu(V_0) = V_0$ it holds that the size of the symmetric difference between G and $\mu(G)$ is $(0.5 \pm \epsilon) \cdot |V_0| \cdot |V_1|$. Then, $F : V_0 \times V_1 \to \{0, 1\}$ such that $F(x, y) = 1$ if and only if $\{x, y\} \in E$ is a non-malleable two-source $(\ell, \epsilon + \sqrt{2\epsilon} + o(1))$-extractor.*

Needless to say, the claim holds also if G is augmented by complete graph on the vertex-set V_1. Note that we lose a $\sqrt{2\epsilon} + o(1)$ term in the reversal.

[39] Formally, we should extend μ_0 and μ_1 to (arbitrary) derangements f and g, respectively. (Note that we may assume, w.l.o.g., that $|T_\sigma \cup \mu(T_\sigma)| \leq |V_\sigma| - 2$.) Lastly, note that (18) implies that $\Pr[\mathtt{nmE}(X, Y) \neq \mathtt{nmE}(f(X), g(Y))] \geq \Pr[U_1 \neq \mathtt{nmE}(f(X), g(Y))] - \epsilon = 0.5 - \epsilon$.

[40] That is, the requirement regarding the symmetric difference between G and $\mu(G)$ is made only for permutations μ that have no fixed-points and satisfy $\mu(V_0) = V_0$.

Proof. Let (f,g) and (X,Y) be as in Definition 8.1, and note that in this case X and Y are independent distributions that are each uniformly distributed on $[2^\ell]$. Define $\mu : V \to V$ such that $\mu(z) = f(z)$ if $z \in V_0$ and $\mu(z) = g(z)$ otherwise, and note that μ is a derangement that preserves the partition of V. Recall that $(\mu(x), \mu(y))$ contributes to the symmetric difference between G and $\mu(G)$ if and only if $F(\mu(x), \mu(y)) \neq F(x,y)$, since $\mu(x)$ is connected to $\mu(y)$ in $\mu(G)$ if and only if x is connected to y in G. Hence, by the hypothesis, we have

$$\Pr[F(X,Y) \neq F(\mu(X), \mu(Y))] = 0.5 \pm \epsilon. \tag{21}$$

Letting $p^\mu_{\sigma,\tau} \stackrel{\rm def}{=} \Pr[(F(X,Y), F(\mu(X), \mu(Y))) = (\sigma,\tau)]$, we have $p^\mu_{0,1} + p^\mu_{1,0} = 0.5 \pm \epsilon$, and using the fact that (X,Y) and $(\mu(X), \mu(Y))$ are identically distributed we have $p^\mu_{1,0} = p^\mu_{0,1}$ (since $p^\mu_{1,1} + p^\mu_{1,0} = p^\mu_{1,1} + p^\mu_{0,1}$). Hence, $p^\mu_{0,1} = 0.25 \pm 0.5\epsilon$. Lastly, we show that $p^\mu_{1,1} + p^\mu_{1,0} = 0.5 \pm \sqrt{\epsilon/2} + o(1)$, and conclude that $p^\mu_{1,1} = 0.25 \pm (0.5\epsilon + \sqrt{\epsilon/2} + o(1))$; it follows that F is a non-malleable (two-source) $(\ell, \epsilon + \sqrt{2\epsilon} + o(1))$-extractor.

To show that $p^\mu_{1,1} + p^\mu_{1,0} = 0.5 \pm \sqrt{\epsilon/2} + o(1)$, we first note that $p \stackrel{\rm def}{=} p^\mu_{1,1} + p^\mu_{1,0} = \Pr[F(X,Y){=}1]$ is actually oblivious of μ. Hence, by considering a random derangement μ that preserves V_0 (i.e., $\mu(V_0) = V_0$), we observe that, with overwhelmingly high probability (over the choice of μ), it holds that $\{(x,y) \in V_0 \times V_1 : F(x,y) \neq F(\mu(x), \mu(y))\}$ has size $(2p(1{-}p) {\pm} o(1)) {\cdot} |V_0| {\cdot} |V_1|$. Confronting this with (21), we infer that $p = 0.5 {\pm} (\sqrt{\epsilon/2} {+} o(1))$. ◀

Corollary

Combining Theorem 8.3 with the non-malleable two-source extractors of [7, Thm. 1], while using Proposition 8.2, we obtain an efficient construction of $\Omega(n)$-robustly self-ordered graphs (alas not a strongly explicit (aka locally computable) one).

▶ **Theorem 8.6** (constructing $\Omega(n)$-robustly self-ordered n-vertex graphs)**.** *There exist an algorithm that, on input n, works in $\mathrm{poly}(n)$-time and outputs an explicit description of an $\Omega(n)$-robustly self-ordered $O(n)$-vertex graph. Furthermore, each vertex in this graph has degree at least $0.24 \cdot n$ and at most $0.76 \cdot n$.*

The degree bounds follow by observing that the vertices in the graph described in Theorem 8.3 have degree at least $(0.5 - \epsilon) \cdot n/2$ and at most $(1.5 + \epsilon) \cdot n/2$, whereas [7, Thm. 1] provides for $\epsilon = o(1)$.

8.2 The second construction

Combining Theorem 8.3 with the non-malleable two-source extractors of [7, Thm. 1], while using Proposition 8.2, we obtained an efficient construction of $\Omega(n)$-robustly self-ordered n-vertex graphs (see Theorem 8.6). However, this construction is not locally computable (as postulated in Theorem 1.4), because the non-malleable two-source extractors of [7, Thm. 1] are not quasi-orthogonal and the transformation of Proposition 8.2 runs in time that is polynomial in the size of the resulting graph.

To avoid the foregoing transformation and prove Theorem 1.4, we employ a variant on the construction presented in Theorem 8.3. Rather than connecting two sets of vertices using a bipartite graph that corresponds to a *quasi-orthogonal* non-malleable two-source extractor, we connect three sets of vertices such that one pair of vertex-sets is connected by a (not necessarily quasi-orthogonal) non-malleable two-source extractor, whereas the other two pairs are connected by bipartite graphs that are merely quasi-orthogonal. In analogy to the definition of a quasi-orthogonal (two-source) extractor, we say that a bipartite graph on the vertex-set $X \cup Y$ is **quasi-orthogonal** (with error ϵ) if the following two conditions hold regarding its adjacency predicate $B : X \times Y \to \{0,1\}$:

1. *The degree of each vertex is approximately half the number of the vertices on the other side*: For each $x \in X$ (resp., $y \in Y$), it holds that $|\{y \in Y : B(x,y)=1\}| = (0.5 \pm \epsilon) \cdot |Y|$ (resp., $|\{x \in X : B(x,y)=1\}| = (0.5 \pm \epsilon) \cdot |X|$).
2. *Each pair of vertices on one side neighbors approximately a quarter of the vertices on the other side*: For every $x \neq x' \in X$, it holds that $|\{y \in Y : B(x,y) \neq B(x',y)\}| = (0.5 \pm \epsilon) \cdot |Y|$. Similarly, for $y \neq y' \in Y$.

We note that the inner-product (mod 2) extractor [9], denoted $E_2 : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}$, corresponds to a quasi-orthogonal bipartite graph for the case $X = Y = \{0,1\}^\ell \setminus \{0^\ell\}$. We will however need quasi-orthogonal bipartite graphs with different-sized sides, which can be obtained by a simple variant. Specifically, for the case of $X = \{0,1\}^\ell \setminus \{0^\ell\}$ and $Y = \{0,1\}^{\ell+2} \setminus \{0^{\ell+2}\}$, we use the function $B(x,y) = E_2(G(x),y)$, where $G : \{0,1\}^\ell \to \{0,1\}^{\ell+2}$ is a small-bias generator that satisfies $G(x) \neq 0^{\ell+2}$ and $G(x) \neq G(x')$ for every $x \neq 0^\ell$ and $x' \neq x$ (see Proposition 8.8, and note that $G(a,b,c,d) = (a,b,c,d,E_2(a,b),E_2(c,d))$ will do). We stress that the foregoing construction is strongly explicit (i.e., locally computable).

We shall also assume that the (bipartite graph corresponding to the) non-malleable extractor $\mathtt{nmE} : [2^\ell - 1] \times [2^\ell - 1] \to \{0,1\}$ has *linear degrees* in the sense that for every x it holds that $|\{y \in [2^\ell - 1] : \mathtt{nmE}(x,y)=1\}| \geq \epsilon' \cdot 2^\ell$ for some constant $\epsilon' > 0$. This can be enforced by starting with an arbitrary non-malleable two-source (k,ϵ')-extractor (e.g., the one of [7, Thm. 1]) and resetting pairs in $m = \epsilon' \cdot 2^\ell$ fixed perfect matchings to 1 (i.e., for each (x,y) in one of these matching, we reset $\mathtt{nmE}(x,y) \leftarrow 1$).[41] This increases the error of the extractor by an additive term of $m/2^k = 2^{\ell-k} \cdot \epsilon'$, which we can afford (e.g., $\epsilon' = 0.01$ and $k = \ell - 4$, yields extraction error $\epsilon < 0.2$). We stress that this transformation preserves polynomial-time computability of the extracting function.

▶ **Theorem 8.7** (using a non-malleable two-source extractor with linear degrees to obtain a $\Omega(2^\ell)$-robustly self-ordered $O(2^\ell)$-vertex graph)**.** *For any constants $\epsilon, \epsilon' \in (0, 0.5)$ and varying $k \leq \ell - 4$, where $\ell \in \mathbb{N}$, suppose that $\mathtt{nmE} : [2^\ell - 1] \times [2^\ell - 1] \to \{0,1\}$ is a non-malleable two-source (k,ϵ)-extractor such that for every x it holds that $|\{y \in [2^\ell - 1] : \mathtt{nmE}(x,y)=1\}| > \epsilon' \cdot 2^\ell$. Further suppose that $B : [2^\ell - 1] \times [2^{\ell+2} - 1] \to \{0,1\}$ is quasi-orthogonal with error $0.1 \cdot \epsilon'$. Then, the $(6 \cdot 2^\ell - 3)$-vertex graph $G = (V_0 \cup V_1 \cup V_2, E)$ such that $V_\sigma = \{\langle \sigma, i\rangle : i \in [2^{\ell_\sigma} - 1]\}$, where $\ell_0 = \ell_1 = \ell$ and $\ell_2 = \ell + 2$, and*

$$E \;=\; \{\{\langle 1,i\rangle, \langle 0,j\rangle\} : \mathtt{nmE}(i,j)=1\} \cup \{\{\langle \sigma,i\rangle, \langle 2,j\rangle\} : B(i,j)=1, \sigma \in \{0,1\}\} \cup \binom{V_1}{2} \cup \binom{V_2}{2} \quad (22)$$

is $\Omega(|V|)$-robustly self-ordered, where $V = V_0 \cup V_1 \cup V_2$. Furthermore, each vertex in this graph has degree at least $0.3 \cdot |V|$ and at most $0.9 \cdot |V|$.

Using the foregoing ingredients (including the non-malleable extractor of [7, Thm. 1]), Theorem 1.4 follows (see also Remark 8.9). Looking at (22), note that the first set of edges corresponds to a bipartite graph between V_1 and V_0 that is determined by $\mathtt{nmE}$, the second set corresponds the bipartite graphs between V_σ (for $\sigma \in \{0,1\}$) and V_2 that are determined by B, and the other two sets correspond to cliques on V_1 and on V_2. (See Figure 2.)

[41] For example, we may use the matchings $\{(z, z+i) : z \in [2^\ell - 1]\}$ for $i \in [m]$, where addition is mod $2^\ell - 1$. In addition, starting from an extractor that is defined over ℓ-bit strings, we may omit one of these strings (and obtain an extractor defined over $[2^\ell - 1]$).

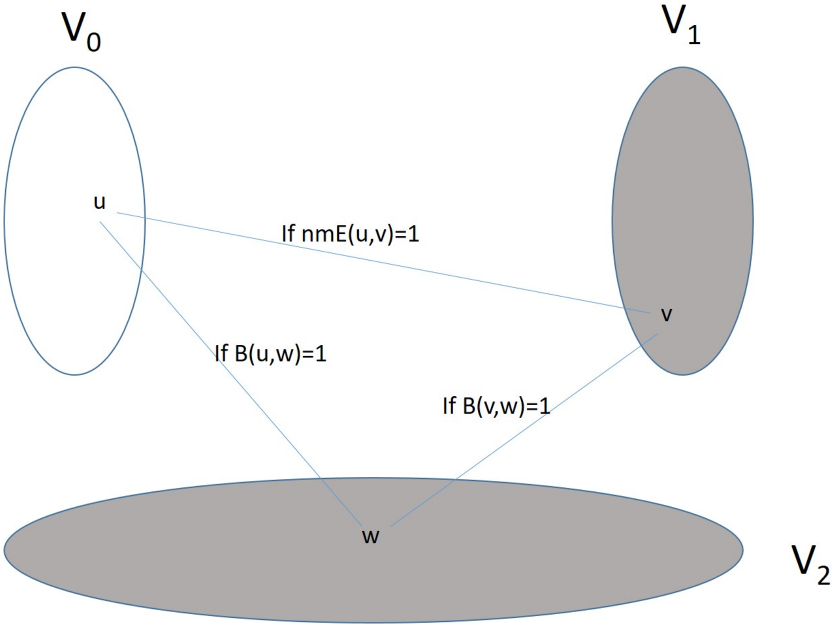


Figure 2 Illustrating the construction of Theorem 8.7.

Proof. Recall that $V = V_0 \cup V_1 \cup V_2$, and consider an arbitrary (non-trivial) permutation $\mu : V \to V$. Intuitively, if μ maps a vertex of V_0 (or V_1) to V_2, then the difference in degrees of vertices in the two sets (caused by the $|V_2|$-clique edges) contributes $\Omega(|V|)$ units to the symmetric difference between G and $\mu(G)$, where here we use the first quasi-orthogonality condition of B. A similar argument, which uses the V_1-clique edges and relies on the linear degrees of nmE, applies to a vertex of V_σ mapped to $V_{1-\sigma}$ for any $\sigma \in \{0,1\}$. On the other hand, if for some $\sigma \in \{0,1,2\}$ the bijection μ maps $\langle\sigma,i\rangle \in V_\sigma$ to $V_\sigma \setminus \{\langle\sigma,i\rangle\}$, then the difference in the neighborhoods caused by one of the two relevant bipartite graphs contributes $\Omega(|V|)$ units to the symmetric difference between G and $\mu(G)$. Here, we distinguishes between the case that μ has relatively few non-fixed-points in either V_0 or V_1, which is analyzed using the second quasi-orthogonality condition of B, and the case that μ has relatively many non-fixed-points in both V_0 and V_1, which is analyzed using the non-malleability condition of nmE. Indeed, the structure of the proof is similar to the one of Theorem 8.3, but the details are different in many aspects, and so we provide them below.

Let $T = \{v \in V : \mu(v) \neq v\}$ denote the set of non-fixed-points of μ. Then, we consider two types of vertices: Those that belong to the set $T' = \bigcup_{\sigma\in\{0,1,2\}}\{v \in V_\sigma : \mu(v) \notin V_\sigma\} \subseteq T$ and those that belong to $T \setminus T'$. The threshold for distinguishing these cases is set to $K = (0.5 - 0.1 \cdot \epsilon') \cdot |V_0|/4 = \Omega(|V|)$.[42] Recall that ϵ denotes the extraction error of nmE, whereas ϵ' is the fractional degree bound associated with its linear degrees feature, and $0.1 \cdot \epsilon'$ is the quasi-orthogonality error of B.

Case 1: $|T'| \geq K$.

(This refers to the case that many vertices are mapped by μ to a different part of the three-way partition (V_0, V_1, V_2) of V, where "many" means $\Omega(|V|)$.)

Each vertex in T' contributes $\Omega(|V|)$ units to the symmetric difference between G and $\mu(G)$, because of the differences in the degrees of vertices in the three parts. Specifically:

[42] The threshold is set depending on the quasi-orthogonality error of B. In the proof of Theorem 8.3, the threshold was set depending on the quasi-orthogonality error of nmE (which equaled its extraction error).

- Vertices in V_2 have degree at least $(|V_2|-1)+(0.5-0.1\epsilon')\cdot(|V_0|+|V_1|) > (5-0.2\epsilon')\cdot|V_0|-O(1)$, where the first term is due to the clique edges and the second term is due to the bipartite graphs connecting V_2 to V_0 and V_1 (and relies on the first quasi-orthogonality condition of B).
- Vertices in V_0 have degree at most $|V_1|+(0.5+0.1\epsilon')\cdot|V_2| < (3+0.4\epsilon')\cdot|V_0|+O(1)$, where the first term is due to the edges (determined by $\texttt{nmE}$) connecting V_0 to V_1 and the second term is due to the bipartite graph connecting V_0 to V_2.
- Vertices in V_1 have degree at least $(|V_1|-1)+\epsilon'\cdot|V_0|+(0.5-0.1\epsilon')\cdot|V_2| > (3+0.6\epsilon')\cdot|V_0|-O(1)$ and at most $(|V_1|-1)+|V_0|+(0.5+0.1\epsilon')\cdot|V_2| < (4+0.4\epsilon')\cdot|V_0|$. In both cases, the first term is due to clique edges, the second term is due to the edges connecting V_1 to V_0 (as determined by $\texttt{nmE}$), and the third term is due to the edges connecting V_1 to V_2 (as determined by B). The crucial fact is that the linear degrees of $\texttt{nmE}$ provides a non-trivial lower bound (of $\epsilon'\cdot|V_0|$) on the second term.

Hence, the difference in the degrees of vertices in the different parts is at least $0.2\epsilon'\cdot|V_0|-O(1)$, where the minimum is due to the difference between the degrees of vertices in V_1 and the degrees of vertices in V_0.

It follows that the symmetric difference between G and $\mu(G)$ is at least $(0.2\epsilon'\cdot|V_0|-O(1))\cdot|T'| = \Omega(|V|)\cdot|T'|$, since $|V_0|=\Omega(|V|)$ and $\epsilon'=\Omega(1)$. Using the case's hypothesis, we have $|T'|=\Omega(|V|)=\Omega(|T|)$, which means that in this case the difference between G and $\mu(G)$ is $\Omega(|V|)\cdot|T|$.

We stress that the difference between G and $\mu(G)$ is at least $\Omega(|V|)\cdot|T'|$ also if the case hypothesis does not hold.

Case 2: $|T'|<K$.

(This refers to the case that few vertices are mapped by μ to a different part of the three-way partition (V_0,V_1,V_2) of V.)

For every $\sigma\in\{0,1,2\}$, let $V'_\sigma = V_\sigma\cap\mu(V_\sigma)$ and $T_\sigma = V'_\sigma\cap T$. Indeed, (T',T_0,T_1,T_2) is a four-way partition of T. Note that the size of the symmetric difference between G and $\mu(G)$ is lower-bounded by

$$\begin{aligned}&|\{(v,u)\in V'_1\times V'_0 : \texttt{nmE}(\mu(v),\mu(u))\neq\texttt{nmE}(v,u)\}|\\ &+\ |\{(v,u)\in V'_1\times V'_2 : B(\mu(v),\mu(u))\neq B(v,u)\}|\\ &+\ |\{(v,u)\in V'_0\times V'_2 : B(\mu(v),\mu(u))\neq B(v,u)\}|\,,\end{aligned}\tag{23}$$

since, for any $(v,u)\in V'_1\times V'_0$, it holds that $\mu(v)$ neighbors $\mu(u)$ in G if and only if $\texttt{nmE}(\mu(v),\mu(u))=1$, whereas $\mu(v)$ neighbors $\mu(u)$ in $\mu(G)$ if and only if v neighbors u in G which holds if and only if $\texttt{nmE}(v,u)=1$. Ditto for the other two cases.

We consider two sub-cases according to whether or not $\min(|T_0|,|T_1|)$ is relatively large. The threshold for distinguishing these sub-cases is also set to $K=(0.5-0.1\cdot\epsilon')\cdot|V_0|/4$; note that $K=\Omega(|V|)$ and $K>0.1\cdot|V_0|>2^{\ell-4}\geq 2^k$.

Case 2.1: $\min(|T_0|,|T_1|)<K$.

In this case we shall use the quasi-orthogonality of B.

Suppose, without loss of generality, that $|T_0|\leq|T_1|$, which implies $|T_0|<K$.

Depending on the relative sizes of T_1 and T_2, we shall use either the quasi-orthogonal bipartite graph between V_1 and V_2 or the quasi-orthogonal bipartite graph between V_2 and V_0.

1. On the one hand, if $|T_1| > |T_2|$, then we consider the quasi-orthogonal bipartite graph between V_1 and V_2. The contribution of each vertex $v \in T_1$ to (23) equals
$$\begin{aligned} &|\{u \in V_2' : B(\mu(v), \mu(u)) \neq B(v,u)\}| \\ &\geq\ |\{u \in V_2' : B(\mu(v), u) \neq B(v,u)\}| - |T_2| \\ &>\ |\{u \in V_2 : B(\mu(v), u) \neq B(v,u)\}| - |T'| - |T_1| \\ &\geq\ (0.5 - 0.1 \cdot \epsilon') \cdot |V_2| - K - |V_0| \\ &>\ 0.6 \cdot |V_0| \end{aligned}$$
where the first inequality uses $\mu(u) = u$ for $u \in V_2' \setminus T_2$, the second inequality uses $|V_0'| \geq |V_0| - |T'|$ and the hypothesis $|T_2| < |T_1|$, the third inequality uses $\mu(v) \neq v$ along with the (second condition of) quasi-orthogonality of B (and the hypotheses $|T'| < K$ and the fact that $|T_1| \leq |V_1| = |V_0|$), and the fourth inequality uses $\epsilon' < 0.5$ and $|V_2| > 4 \cdot |V_0|$. So the total contribution in this sub-case is $|T_1| \cdot \Omega(|V|) \geq (|T| - |T'|) \cdot \Omega(|V|)$, since $|T_1| \geq \max(|T_0|, |T_2|)$ and $|T_0| + |T_1| + |T_2| = |T| - |T'|$.
2. On the other hand, if $|T_1| \leq |T_2|$, then we consider the quasi-orthogonal bipartite graph between V_2 and V_0. The contribution of each vertex $v \in T_2$ to (23) equals
$$\begin{aligned} &|\{u \in V_0' : B(\mu(u), \mu(v)) \neq B(u,v)\}| \\ &\geq\ |\{u \in V_0' : B(u, \mu(v)) \neq B(u,v)\}| - |T_0| \\ &\geq\ |\{u \in V_0 : B(u, \mu(v)) \neq B(u,v)\}| - |T'| - |T_0| \\ &\geq\ (0.5 - 0.1 \cdot \epsilon') \cdot |V_0| - 2 \cdot K \\ &=\ (0.5 - 0.1 \cdot \epsilon') \cdot |V_0|/2 \end{aligned}$$
where the first inequality uses $\mu(u) = u$ for $u \in V_0' \setminus T_0$, the second inequality uses $|V_0'| \geq |V_0| - |T'|$, the third inequality uses $\mu(v) \neq v$ along with the (second condition of) quasi-orthogonality of B (and the hypotheses regarding $|T'|$ and $|T_0|$), and the equality is due to $K = (0.5 - 0.1 \cdot \epsilon') \cdot |V_0|/4$. So the total contribution in this sub-case is $|T_2| \cdot \Omega(|V|) \geq (|T| - |T'|) \cdot \Omega(|V|)$, since $|T_2| \geq |T_1| \geq |T_0|$.

Hence, the total contribution (of Case 2.1) to (23) is $\Omega(|V|) \cdot (|T| - |T'|)$.

Case 2.2: $\min(|T_0|, |T_1|) \geq K$.

In this case we shall use the non-malleable feature of nmE.

Specifically, for each $\sigma \in \{0,1\}$, let μ_σ denote the restriction of μ to T_σ. Essentially, using $K \geq 2^k$, the non-malleability condition of the (k, ϵ)-extractor nmE implies

$$|\{(i,j) \in T_0 \times T_1 : \mathtt{nmE}(i,j) \neq \mathtt{nmE}(\mu_0(i), \mu_1(j))\}| \geq (0.5 - \epsilon) \cdot |T_0| \cdot |T_1|.$$

This can be seen by letting X and Y be uniform over T_0 and T_1, respectively, and combining the fact that $\Pr[\mathtt{nmE}(\mu_0(X), \mu_1(Y)) \neq U_1] = 0.5$ with the non-malleability condition (while noting that $\mu_0 : T_0 \to \mu(T_0)$ and $\mu_1 : T_1 \to \mu(T_1)$ have no fixed-points).[43]

Hence, in this case, the total contribution to (23) is $(0.5 - \epsilon) \cdot |T_0| \cdot |T_1| = \Omega(|V|^2)$, where we use $\min(|T_0|, |T_1|) = \Omega(|V|)$.

Hence, in both sub-cases, the difference between G and $\mu(G)$ is $\Omega(|V|) \cdot (|T| - |T'|)$.

[43] Formally, we should extend μ_0 and μ_1 to (arbitrary) derangements f and g, respectively. (Note that we may assume, w.l.o.g., that $|T_\sigma \cup \mu(T_\sigma)| \leq |V_\sigma| - 2$.) Lastly, note that (18) implies that $\Pr[\mathtt{nmE}(X,Y) \neq \mathtt{nmE}(f(X), g(Y))] \geq \Pr[U_1 \neq \mathtt{nmE}(f(X), g(Y))] - \epsilon = 0.5 - \epsilon$.

Recall that (by the last comment at Case 1) the difference between G and $\mu(G)$ is $\Omega(|V|)\cdot|T'|$. Combining this lower-bound with the conclusion of Case 2, the difference between G and $\mu(G)$ is $\Omega(|V|)\cdot|T|$. As for the degree bounds, note that each vertex has degree at most $(|V_2|-1)+(0.5-0.1\epsilon')\cdot(|V_0|+|V_1|)=(5+0.2\epsilon')\cdot|V_0|+O(1)$, and at least $(0.5-0.1\epsilon')\cdot|V_2|<(2-0.4\epsilon')\cdot|V_0|-O(1)$, where maximum (resp., minimum) is obtained by vertices in V_2 (resp., V_0). ◀

Digest

Compared to the construction used in Theorem 8.3, the construction in Theorem 8.7 decouples the non-malleable feature from the quasi-orthogonality feature, using non-malleable extractors for connecting one pair of vertex-sets and quasi-orthogonal functions to connect the other two pairs. The current analysis is slightly more complex because it has to handle the fact that these features hold for different pairs. Specifically, the quasi-orthogonality of B is used in Cases 1 and 2.1, whereas the non-malleability of `nmE` is used in Case 2.2. In particular, Case 1 only uses the first condition of quasi-orthogonality, and does so in order to infer that the degrees of all vertices in the bipartite graph determined by B are approximately equal. In Case 2.1 the second quasi-orthogonality condition is used in order to assert that the neighborhoods of two different vertices in V_σ (for every $\sigma\in\{0,1,2\}$) are significantly different. This is useful only when the number of non-fixed-points in the other side of the graph B is relatively small.

In light of the key role that quasi-orthogonal unbalanced bipartite graphs play in Theorem 8.7 and given their natural appeal, it feel adequate to provide a general construction of these graphs, which generalizes the construction outlined before Theorem 8.7 (for the case of $\ell'=\ell+2$).

▶ **Proposition 8.8** (quasi-orthogonal unbalanced bipartite graphs)**.** *For $S_\ell\overset{\text{def}}{=}\{0,1\}^\ell\setminus\{0^\ell\}$ let $G:S_\ell\to S_{\ell'}$ be small-bias generator with bias ϵ such that $G(s)\neq G(s')$ for every $s\neq s'$, and let E_2 denote the inner-product mod 2 function. Then, the bipartite graph described by the adjacency predicate $B:S_\ell\times S_{\ell'}\to\{0,1\}$ such that $B(x,y)=E_2(G(x),y)$ is quasi-orthogonal with error ϵ.*

(Note that the hypothesis implies $\epsilon>1/|S_{\ell'}|$. The definition of quasi-orthogonal bipartite graphs appears before Theorem 8.7.)

Proof Sketch. Our starting point is the fact that $E_2:S_{\ell'}\times S_{\ell'}\to\{0,1\}$ is quasi-orthogonal with error $1/|S_{\ell'}|$. The quasi-orthogonality feature of the first argument of B follows as a special case of the corresponding feature of E_2. Turning to fixings of the second argument of E_2 and letting X be uniform over S_ℓ, we observe that, for every $y\in S_{\ell'}$, the bit $B(G(X),y)$ is a linear combination of the bits of $G(X)$, and hence $\Pr[B(G(X),y)=1]=0.5\pm\epsilon$. Similarly, for $y\neq y'$, it holds that $B(G(X),y)\oplus B(G(X),y')=B(G(X),y\oplus y')$ is linear combination of the bits of $G(X)$. ◀

▶ Remark 8.9 (obtaining $\Omega(n)$-robustly self-ordered n-vertex graphs, for every n). Theorem 8.7 provides a construction of $\Omega(n)$-robustly self-ordered n-vertex graphs, for every n of the form $6\cdot2^\ell-3$, where $\ell\in\mathbb{N}$. A construction for every $n\in\mathbb{N}$ can be obtained by using a few minor modifications.

- Rather than using $|V_2|=2^{\ell+2}-1=4\cdot(|V_0|+1)-3$, we may use $|V_2|=n-2\cdot|V_0|$ such that $|V_0|=\Omega(n)$. Specifically, we still use $|V_0|=2^\ell-1$, for $\ell=\log_2 n-\Theta(1)$, along with $|V_2|\in[4\cdot|V_0|,10\cdot|V_0|]$. Doing so requires decreasing the quasi-orthogonality error of B to $0.04\epsilon'$ so that $0.04\epsilon'\cdot|V_2|\le0.4\cdot|V_0|$ still holds.

- More importantly, we need a construction of a quasi-orthogonal bipartite graph with an adjacency predicate $B : [2^\ell - 1] \times [n'] \to \{0,1\}$ such that $n' = n - 2 \cdot (2^\ell - 1) \geq 2n/3$. The solution is to associated $[n']$ with an easily enumerable small-bias space $S \subseteq \{0,1\}^{\ell+4} \setminus \{0^{\ell+4}\}$ and use $B(x,y) = E_2(G(x), y)$, where E_2 and G are as in Proposition 8.8. Specifically, for $t = \log_2 \log_2 \ell$ and $D = \lceil n' \cdot 2^t / 2^{\ell+4} \rceil$, we let S contain the n' lexicographically first strings in $S' \times \{0,1\}^{\ell+4-t}$, where S' is a small-bias sample space of size D over $\{0,1\}^t$ that is found by exhaustive search.[44]

8.3 Obtaining efficient self-ordering

We say that a self-ordered graph $G = ([n], E)$ is efficiently self-ordered if there exists a polynomial-time algorithm that, given any graph $G' = (V', E')$ that is isomorphic to G, finds the unique bijection $\phi : V' \to [n]$ such that $\phi(G') = G$ (i.e., the unique isomorphism of G' and G). Indeed, this isomorphism orders the vertices of G' in accordance with the original (or target) graph G.

Recall that in the case of bounded-degree graphs, we relied on the existence of a polynomial-time isomorphism test (see [29]) for efficiently self-ordering the robustly self-ordered graphs that we constructed. We cannot do so in the dense graph case, since a general polynomial-time isomorphism test is not known (see [1]). Instead, we augment the construction asserted in Theorem 1.4 so to obtain dense $\Omega(n)$-robustly self-ordered graphs that are efficiently self-ordered.[45]

▶ **Theorem 8.10** (strengthening Theorem 1.4). *There exist an infinite family of dense $\Omega(n)$-robustly self-ordered graphs $\{G_n\}_{n\in\mathbb{N}}$ and a polynomial-time algorithm that, given $n \in \mathbb{N}$ and a pair of vertices $u, v \in [n]$ in the n-vertex graph G_n, determines whether or not u is adjacent to v in G_n. Furthermore, these graphs are efficiently self-ordered, and the degrees of vertices in G_n reside in* $[0.06n, 0.73n]$.

Proof. Our starting point is the construction of m-vertex graphs that are $\Omega(m)$-robustly self-ordered (see Theorem 1.4, which uses Theorem 8.7). Recall that the vertices in these graphs have degree that ranges between $0.3 \cdot m$ and $0.9 \cdot m$ (see Theorem 8.7).

The idea is to use two such graphs, G_1 and G_2, one with m vertices and the other with $4 \cdot m$ vertices, where $m = n/5$, and connect them in a way that assists finding the ordering of vertices in each of these two graphs. Specifically, we designate a set, denoted S_1, of $s \overset{\text{def}}{=} 2\sqrt{\log_2 n}$ vertices in $G_1 = ([m], E_1)$, and a set, denoted S_2, of $\ell \overset{\text{def}}{=} \binom{s}{2} \in [\log_2 n, 2\log_2 n]$ vertices in $G_2 = (\{m+1, ..., 5m\}, E_2)$, and use them as follows:

- Connect each vertex in S_2 to two different vertices in S_1, while noting that each vertex in S_1 is connected to $2\ell/s = o(\ell)$ vertices of S_2.
- Connect each vertex in $R_1 \overset{\text{def}}{=} [m] \setminus S_1$ to a different set of neighbors in S_2 such that each vertex in R_1 has at least $\ell/2$ neighbors in S_2.

[44] Note that for every $z = (z', z'') \in \{0,1\}^{\ell+4} \setminus \{0^{\ell+4}\}$ and $Y = (Y', Y'')$ that is uniformly distributed over S such that $|z'| = |Y'| = t$ it holds that

$$\mathrm{E}[(-1)^{E_2(z,Y)}] = \mathrm{E}[(-1)^{E_2(z',Y')}] \cdot \mathrm{E}[(-1)^{E_2(z'',Y'')}]$$

where the absolute value of each of the factors is $o(1)$ if the corresponding fixed string (i.e., z' or z'') is non-zero. Specifically, note that Y' (resp., Y'') is $o(1)$-close to being uniformly distributed over S' (resp., $\{0,1\}^{\ell+4-t}$).

[45] Unlike in the bounded degree case (see Section 4.4), we do not know how to construct $\Omega(n)$-robustly self-ordered graphs that support *local* self-ordering. We mention that $\Omega(n)$-robustly self-ordered graphs with information-theoretically local self-ordering do exist [22].

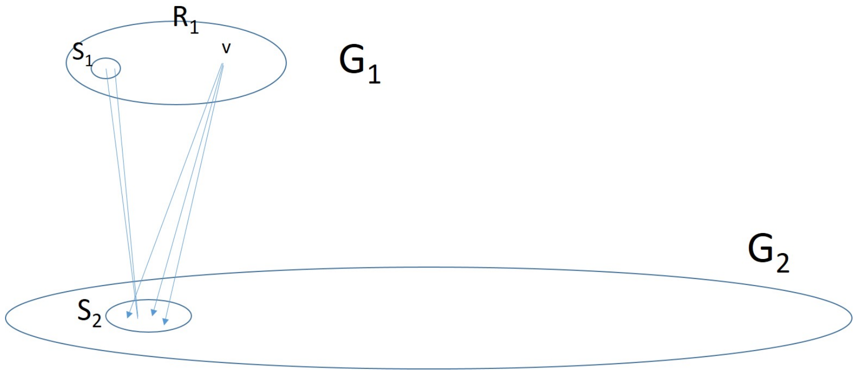


Figure 3 The construction of Theorem 8.10.

- Connect each vertex in $R_2 \overset{\text{def}}{=} \{m+1, ..., 5m\} \setminus S_2$ to a different set of neighbors in R_1 such that each vertex in R_2 has two neighbors in R_1 and each vertex in R_1 has at most eight neighbors in R_2.

(See Figure 3.) Denote the resulting graph by $G = ([n], E)$, and note that the vertices of G_1 have degree at most $0.9 \cdot m + \ell$, whereas the vertices of G_2 have degree at least $0.3 \cdot 4m$. Given an isomorphic copy of the G, we can find the unique isomorphism (i.e., its ordering) as follows:

1. Identify the vertices that belong to G_1 by virtue of their lower degree.
2. Identify the set S_1 as the set of vertices that belong to G_1 and have $2\ell/s = o(\ell)$ neighbors in G_2.
   (Recall that each vertex in R_1 has at least $\ell/2$ neighbors in S_2.)
3. Identify the set S_2 as the set of vertices that belong to G_2 and have (two) neighbors in S_1.
4. For each possible ordering of S_1, order the vertices of S_2 by their neighborhood in S_1, and order the vertices of R_1 according to their neighborhood in S_2.
   If the resulting ordering (of $S_1 \cup R_1$) yields an isomorphism to G_1, them continue. Otherwise, try the next ordering of S_1.
5. Order the vertices of R_2 according to their neighborhood in R_1.

Note that by the asymmetry of G_1, there exists a unique ordering of its vertices, and a unique ordering of S_1 that fits it and leads the procedure to successful termination. One the other hand, the number of possible ordering of S_1 is $s! = n^{o(1)}$, which means that the procedure is efficient.

It is left to show that the graph G is $\Omega(n)$-robustly self-ordered. Let $\gamma \in (0, 1]$ be a constant such that that G_1 (resp., G_2) is $\gamma \cdot m$-robustly self-ordered (resp., $\gamma \cdot 4m$-robustly self-ordered). Then, fixing an arbitrary permutation $\mu : [n] \to [n]$, and letting $T = \{v \in [n] : \mu(v) \neq v\}$, we consider the following cases.

Case 1: $|\{v \in [m] : \mu(v) \in [m]\}| > \gamma \cdot |T|/10$.
In this case, we get a contribution of at least $\Omega(m \cdot |T|)$ units to the symmetric difference between G and $\mu(G)$, because of the difference in degree between vertices in $[m]$ and outside $[m]$. (Recall that the former have degree at most $0.9 \cdot m + \ell < m$, whereas the latter have degree at least $0.3 \cdot 4m = 1.2 \cdot m$.)

Case 2: $t \overset{\text{def}}{=} |\{v \in [m] : \mu(v) \in [m]\}| \leq \gamma \cdot |T|/10$.
In this case, at least $(1 - 0.1\gamma) \cdot |T|$ vertices in T are mapped by μ to the side in which they belong (i.e., each of these vertices v satisfies $v \in [m]$ if and only if $\mu(v) \in [m]$). Let $T_1 \overset{\text{def}}{=} \{v \in T \cap [m] : \mu(v) \in [m]\}$ and $T_2 \overset{\text{def}}{=} \{v \in T \setminus [m] : \mu(v) \notin [m]\}$. Then, the

vertices in T_1 contribute at least $|T_1| \cdot \gamma \cdot m - t \cdot m$ units to the symmetric difference between G and $\mu(G)$, where the negative term is due to possible change in the incidence with vertices that did not maintain their side. Similarly, the vertices in T_2 contribute at least $|T_2| \cdot \gamma \cdot 4m - t \cdot 4m$ units to the symmetric difference. Hence, it total, we get a contribution of at least $(|T| - 2t) \cdot \gamma \cdot m - t \cdot 5m = \Omega(m \cdot |T|)$.

The claims follows.[46] ◀

Digest

The n-vertex graph constructed in the proof of Theorem 8.10 is proved to be $\Omega(n)$-robustly self-ordered by implicitly using the following claim.

▷ Claim 8.11 (combining two $\Omega(n)$-robustly self-ordered graphs). For $i \in \{1,2\}$, let $G_i = (V_i, E_i)$ be an $\Omega(n)$-robustly self-ordered graph, and consider a graph $G = (V_1 \cup V_2, E_1 \cup E_2 \cup E)$ such that E contain edges with a single vertex in each V_i; that is, G consists of G_1 and G_2 and an arbitrary bipartite graph that connects them. If the maximun degree in G of each vertex in V_1 is smaller by an $\Omega(n)$ term from the minimum degree of each vertex in V_2, then G is $\Omega(n)$-robustly self-ordered.

Indeed, Claim 8.11 is analogous to Claim 4.3 (which refers to bounded-degree graphs). We also comment that $\Omega(n)$-robustly self-ordered graph maintain this feature also when $o(n)$ edges are added (and/or removed) from the incidence of each vertex.

9 Application to Testing Dense Graph Properties

In Section 5, we demonstrated the applicability of robustly self-ordered bounded-degree graphs to the study of testing graph properties in the bounded-degree graph model. In the current section, we provide a corresponding demonstration for the regime of dense graphs. Hence, we refer to testing graph properties in the dense graph model, which was introduced in [18] and is surveyed in [16, Chap. 8]. In this model, graphs are represented by their adjacency predicate, and distances are measured as the ratio of the number of differing incidences to the maximal number of edges.

Background

We represent a graph $G = ([n], E)$, by the adjacency predicate $g : [n] \times [n] \to \{0, 1\}$ such that $g(u, v) = 1$ if and only if $\{u, v\} \in E$, and oracle access to a graph means oracle access to its adjacency predicate (equiv., adjacency matrix). The distance between the graphs $G = ([n], E)$ and $G' = ([n], E')$ is defined as the fraction of entries (in the adjacency matrix) on which the two graphs disagree.

▶ **Definition 9.1** (testing graph properties in the dense graph model). *A* tester *for a graph property Π is a probabilistic oracle machine that, on input parameters n and ϵ, and oracle access to an n-vertex graph $G = ([n], E)$ outputs a binary verdict that satisfies the following two conditions.*

1. *If $G \in \Pi$, then the tester accepts with probability at least $2/3$.*
2. *If G is ϵ-far from Π, then the tester accepts with probability at most $1/3$, where G is* ϵ-far *from Π if for every n-vertex graph $G' = ([n], E') \in \Pi$ the adjacency matrices of G and G' disagree on at least $\epsilon \cdot n^2$ entries.*

[46] Note that the degree of each vertex in G_1 is at least $0.3m = 0.06n$, whereas the degree of each vertex in G_2 is at most $0.9 \cdot 4m + s < 0.73n$.

The query complexity of a tester for Π is a function (of the parameters n and ϵ) that represents the number of queries made by the tester on the worst-case n-vertex graph, when given the proximity parameter ϵ.

Our result

We present a general reduction of testing any property Φ of (bit) strings to testing a corresponding graph property Π. Loosely speaking, n-bit long strings will be encoded as part of an $O(\sqrt{n})$-vertex graph, which is constructed using $\Omega(\sqrt{n})$-robustly self-ordered $\Theta(\sqrt{n})$-vertex graphs. This reduction is described in Construction 9.2 and its validity is proved in Lemma 9.3. Denoting the query complexities of Φ and Π by Q_Φ and Q_Π, respectively, we get $Q_\Phi(n,\epsilon) \leq Q_\Pi(O(n^{1/2}),\Omega(\epsilon))$. Thus, lower bounds on the query complexity of testing Φ, which is a property of "ordered objects" (i.e., bit strings), imply lower bounds on the query complexity of testing Π, which is a property of "unordered objects" (i.e., graphs).

Our starting point is the construction of m-vertex graphs that are $\Omega(m)$-robustly self-ordered. Actually, wishing Π to preserve the computational complexity of Φ, we use a construction of graphs that are efficiently self-ordered, as provided by Theorem 8.10. Recall that the vertices in these graphs have degree that ranges between $0.06 \cdot m$ and $0.73 \cdot m$.

The idea is to use two such graphs, G_1 and G_2, one with m vertices and the other with $49 \cdot m$ vertices, where $m = \sqrt{n}$, and encode an n-bit string in the connection between them. Specifically, we view the latter string as a m-by-m matrix, denoted $(s_{i,j})_{i,j\in[m]}$, and connect the i^{th} vertex of G_1 to the j^{th} vertex of G_2 if and only if $s_{i,j} = 1$.

▶ **Construction 9.2** (from properties of strings to properties of dense graphs). *Suppose that $\{G_m = ([m], E_m)\}_{m\in\mathbb{N}}$ is a family of $\Omega(m)$-robustly self-ordered graphs. For every $n \in \mathbb{N}$, we let $m = \sqrt{n}$, and proceed as follows.*

- *For every $s \in \{0,1\}^n$ views as $(s_{i,j})_{i,j\in[m]} \in \{0,1\}^{m\times m}$, we define the graph $G'_s = ([50m], E'_s)$ such that*

$$E'_s = E_m \cup \{\{m+i, m+j\} : \{i,j\} \in E_{49m}\} \cup \{\{i, m+j\} : i,j \in [m] \wedge s_{i,j} = 1\} \quad (24)$$

  That is, G'_s consists of a copy of G_m and a copy of G_{49m} that are connected by a bipartite graph that is determined by s.
- *For a set of strings Φ, we define $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$ as the set of all graphs that are isomorphic to some graph G'_s such that $s \in \Phi$; that is,*

$$\Pi_n = \{\pi(G'_s) : s \in (\Phi \cap \{0,1\}^n) \wedge \pi \in \mathrm{Sym}_{50m}\} \quad (25)$$

  where Sym_{50m} denote the set of all permutations over $[50m]$.

Note that, given a graph of the form $\pi(G'_s)$, the vertices of G_m are easily identifiable (as having degree at most $0.73m + m < 1.8m$).[47] The foregoing construction yields a local reduction of Φ to Π, where locality means that each query to G'_s can be answered by making a constant number of queries to s. The (standard) validity of the reduction (i.e., $s \in \Phi$ if and only if $G'_s \in \Pi$) is based on the fact that G_m and G_{49m} are asymmetric.

In order to be useful towards proving lower bounds on the query complexity of testing Π, we need to show that the foregoing reduction is "distance preserving" (i.e., strings that are far from Φ are transformed into graphs that are far from Π). The hypothesis that G_m and G_{49m} are $\Omega(m)$-robustly self-ordered is pivotal to showing that if the string s is far from Φ, then the graph G'_s is far from Π.

[47] In contrast, the vertices of G_{49m} have degree at least $0.06 \cdot 49m > 2.9m$.

▶ **Lemma 9.3** (preserving distances). *If $s \in \{0,1\}^n$ is ϵ-far from Φ, then the $50m$-vertex graph G'_s (as defined in Construction 9.2) is $\Omega(\epsilon)$-far from Π.*

Proof. We prove the contrapositive. Suppose that G'_s is δ-close to Π. Then, for some $r \in \Phi$ and a permutation $\pi : [50m] \to [50m]$, it holds that G'_s is δ-close to $\pi(G'_r)$, which means that these two graphs differ on at most $\delta \cdot (50m)^2$ vertex pairs. If $\pi(i) = i$ for every $i \in [2m]$, then s must be $O(\delta)$-close to r, since $s_{i,j} = 1$ (resp., $r_{i,j} = 1$) if and only if i is connected to $m + j$ in G'_s (resp., in $\pi(G'_r) = G'_r$).[48] Unfortunately, the foregoing condition (i.e., $\pi(i) = i$ for every $i \in [2m]$) need not hold in general.

In general, the hypothesis that $\pi(G'_r)$ is δ-close to G'_s implies that π maps at most $O(\delta m)$ vertices of $[m]$ to $\{m + 1, ..., 2m\}$, and maps to $[m]$ at most $O(\delta m)$ vertices that are outside it. This is the case because each vertex of $[m]$ has degree smaller than $0.73m + m < 1.8m$, whereas the other vertices have degree at least $0.06 \cdot 49m > 2.9m$.

Turning to the vertices $i \in [m]$ that π maps to $[m] \setminus \{i\}$, we upper-bound their number by $O(\delta m)$, since the difference between $\pi(G'_r)$ and G'_s is at most $\delta \cdot (50m)^2$, whereas the hypothesis that G_m is $c \cdot m$-robustly self-ordered implies that the difference between $\pi(G'_r)$ and G'_s (or any other graph G'_w) is at least

$$\Delta = c \cdot m \cdot |\{i\in[m]:\pi(i) \neq i\}| - m \cdot |\{i\in[m]:\pi(i) \notin [n]\}|.$$

(Hence, $|\{i \in [m] : \pi(i) \neq i\}| \leq \frac{\Delta + m \cdot O(\delta m)}{cm} = O(\delta m)$.) The same considerations apply to the vertices $i \in \{m + 1, ..., 2m\}$ that π maps to $\{m + 1, ..., 2m\} \setminus \{i\}$; their number is also upper-bounded by $O(\delta m)$.

For every $k \in \{1, 2\}$, letting $I_k = \{i\in[m]:\pi((k-1)\cdot m+i)=(k-1)\cdot m+i\}$, observe that $D \overset{\text{def}}{=} |\{(i, j) \in I_0 \times I_1 : r_{i,j} \neq s_{i,j}\}| \leq \delta \cdot (50m)^2$, since $r_{i,j} \neq s_{i,j}$ implies that $\pi(G'_r)$ and G'_s differ on the vertex-pair $(i, m + j)$. Recalling that $m - |I_k| = O(\delta m)$, it follows that

$$|\{(i, j) \in [m] : r_{i,j} \neq s_{i,j}\}| \leq ((m - |I_1|) - (m - |I_2|)) \cdot m + D = O(\delta m^2).$$

Hence, s is $O(\delta)$-close to $r \in \Phi$, and the claims follows. ◀

10 The Case of Intermediate Degree Bounds

While Section 2–6 study bounded-degree graphs and Sections 7–9 study dense graphs (i.e., constant edge density), in this section we shall consider graphs of intermediate degree bounds. That is, for every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \in [\Omega(1), n]$, we consider n-vertex graphs of degree bound $d(n)$. In this case, the best robustness we can hope for is $\Omega(d(n))$, and we shall actually achieve it for all functions d.

▶ **Theorem 10.1** (robustly self-ordered graphs for intermediate degree bounds). *For every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n)$ is computable in $\mathrm{poly}(n)$-time, there exists an efficiently constructable family of graphs $\{G_n\}_{n\in\mathbb{N}}$ such that G_n has maximal degree $d(n)$ and is $\Omega(d(n))$-robustly self-ordered.*

We prove Theorem 10.1 in three parts, each covering a different regime of degree-bounds (i.e., $d(n)$'s). Most of the range (i.e., $d(n) = \Omega(\log n)^{0.5}$) is covered by Theorem 10.2, whereas Theorem 10.3 handles small degree-bounds (i.e., $d(n) = O(\log n)^{0.499}$) and Theorem 10.5

[48] Hence, G'_s is δ-close to G'_r implies that $|\{i, j \in [n] : s_{i,j} \neq r_{i,j}\}| \leq \delta \cdot (50m)^2$, which means that s is $\frac{(50m)^2\delta}{n}$-close to r. (Recall that $m = \sqrt{n}$.)

handles the degree-bounds that are in-between. One ingredient in the proof of Theorem 10.5 is a transformation of graphs that makes them expanding, while preserving their degree and robustness parameters up to a constant factor. This transformation, which is a special case of Theorem 10.4, is of independent interest.

▶ **Theorem 10.2** (robustly self-ordered graphs for large degree bounds). *For every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \geq O(\sqrt{\log n})$ is computable in* poly(n)*-time, there exists an efficiently constructable family of graphs $\{G_n\}_{n\in\mathbb{N}}$ such that G_n has maximal degree $d(n)$ and is $\Omega(d(n))$-robustly self-ordered.*

The graphs will consist of connected components of size $d(n)$, and in this case $d(n) = \Omega(\sqrt{\log n})$ is necessary, since these components must be different.

Proof Sketch. We combine ideas from Construction 9.2 with elements of the proof of Theorem 4.2. Specifically, as in Construction 9.2, we shall use constructions of m-vertex and $9m$-vertex graphs that are $\Omega(m)$-robustly self-ordered, but here we set $m = d(n)/10$ and use $n/d(n)$ different $d(n)$-vertex graphs that are based on the foregoing two graphs. As in the proof of Theorem 4.2, these ($10m$-vertex) graphs will be far from being isomorphic to one another and will form the connected components of the final n-vertex graph.

Our starting point is the construction of m-vertex graphs that are $\Omega(m)$-robustly self-ordered. Specifically, we may use Theorem 8.6 and note that in this case the vertices in these m-vertex graph have degree that ranges between $0.24 \cdot m$ and $0.76 \cdot m$. Furthermore, these graphs have extremely high conductance; that is, in each of these graphs, the number of edges crossing each cut (in the graph) is at least $\Omega(m)$ times the number of vertices in the smaller side (of the cut).

The idea is to use two such graphs, G_1 and G_2, one with $m \overset{\text{def}}{=} 0.1 \cdot d(n)$ vertices and the other with $0.9 \cdot d(n) = 9 \cdot m$ vertices, and connect them in various ways as done in Section 4.2. Specifically, using an error correcting code with constant rate and constant relative distance and weight, denoted $C : [2^k] \to \{0,1\}^{m^2}$, we obtain a collection of $2^k \geq n/d(n)$ strongly connected $d(n)$-vertex graphs such that the i^{th} graph consists of copies of G_1 and G_2 that are connected according to the codeword $C(i)$; more specifically, we use the codeword $C(i)$ (viewed as an m-by-m matrix) in order to determine the connections between the vertices of G_1 and the first $0.1 \cdot d(n)$ vertices of G_2. The final n-vertex graph, denoted G, consists of $n/d(n)$ connected components that are the first $n/d(n)$ graphs in this collection.[49]

The analysis adapts the analysis of the construction presented in the proof of Theorem 4.2. Towards this analysis, we let $G_j^{(i)}$ denote the i^{th} copy of G_j; that is, the copy of G_j that is part of the i^{th} connected component of G. Hence, for each $i \in [n/d(n)]$, the i^{th} connected component of G is isomorphic to a graph that consists of copies of $G_1 = ([m], E_1)$ and $G_2 = (\{m+1, ..., 10m\}, E_2)$ such that for every $u, v \in [m]$ the vertex u (of $G_1^{(i)}$) is connected to the vertex $m + v$ (of $G_2^{(i)}$) if and only if $C(i)_{u,v} = 1$. Loosely speaking, considering an arbitrary permutation $\mu : [n] \to [n]$, we proceed as follows.[50]

- The discrepancy between the degrees of vertices in copies of G_1 and G_2 (i.e., degree smaller than $0.76m + m$ versus degree at least $0.24 \cdot 9m$) implies that each vertex that resides in a copy of G_1 and is mapped by μ to a copy of G_2 yields a contribution of $\Omega(d(n))$ units to the symmetric difference between G and $\mu(G)$.

[49] Note that we used $2^k \geq n/d(n)$ and $m^2 = O(k)$, where $m = 0.1 \cdot d(n) > \sqrt{k}$. This setting allows for handling any $d(n) \geq O(\sqrt{\log n})$.

[50] These cases are analogous to the cases treated in the proof of Theorem 4.2, with the difference that we merged Cases 2&3 (resp., Cases 4&5) into our second (resp., third) case.

- Let $\mu'(i)$ (resp., $\mu''(i)$) denote the index of the connected component to which μ maps a plurality of the vertices that reside in $G_1^{(i)}$ (resp., of $G_2^{(i)}$). Then, the extremely high conductance of G_1 (resp., G_2) implies that the vertices that resides in $G_1^{(i)}$ (resp., of $G_2^{(i)}$) and are mapped by μ to a connected component different from $\mu'(i)$ (resp., $\mu''(i)$) yields an average contribution of $\Omega(d(n))$ units per each of these vertices.
- The lower bound on the number of edges between $G_1^{(i)}$ and $G_2^{(i)}$ implies that every i such that $\mu'(i) \neq \mu''(i)$ yields a contribution of $\Omega(d(n)^2)$ units, where we assume that few vertices fell to the previous case (i.e., are mapped by μ in disagreement with the relevant plurality vote). (Analogously to the proof of Theorem 4.2, each of these few exceptional vertices reduces the contribution by at most $d(n)$ units.)
- The $\Omega(d(n))$-robust self-ordering of G_1 (resp., G_2) implies that each vertex that reside in $G_1^{(i)}$ (resp., of $G_2^{(i)}$) and is mapped by μ to a different location in $G_1^{(\mu'(i)}$ (resp., in $G_2^{(\mu''(i))}$) yields a contribution of $\Omega(d(n))$ units. Again, this assumes that few vertices fell to the penultimate case, whereas each of these few vertices reduces the contribution by one unit (per each vertex in the current case).
- The distance between the codewords of C implies that every i such that $\mu'(i) = \mu''(i) \neq i$ yields a contribution of $\Omega(d(n)^2)$, where we assume that few vertices fell to the previous cases.

As in the proof of Theorem 4.2, there may be a double counting across the different cases, but this only means that we overestimate the contribution by a constant factor. Overall the size of the symmetric difference is $\Omega(d(n))$ times the number of non-fixed-points of μ. ◀

Handling smaller degree bounds

Theorem 10.2 is applicable only for degree bounds that are at least $O(\log n)^{0.5}$. A different construction allows handling degree bounds up to $O(\log n)^{0.499}$, which leaves a small gap (which we shall close in Theorem 10.5).

▶ **Theorem 10.3** (robustly self-ordered graphs for small degree bounds). *For every every constant $\epsilon > 0$, and every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \in [\Omega(1), (\log n)^{0.5-\epsilon}]$ is computable in* poly(n)*-time, there exists an efficiently constructable family of graphs $\{G_n\}_{n\in\mathbb{N}}$ such that G_n has maximal degree $d(n)$ and is $\Omega(d(n))$-robustly self-ordered.*

In this case, the graphs will consist of connected components of size $\frac{\Theta(\log n)}{d(n)\cdot\log\log n} > d(n)$.

Proof Sketch. Setting $m(n) \overset{\text{def}}{=} \frac{\Theta(\log n)}{d(n)\cdot\log\log n} > d(n) \cdot (\log n)^{\epsilon}$, we proceed in three steps.

1. We first tighten the proof of Theorem 6.1 such that it establishes that, with probability at least $1 - \exp(-\Omega(d(n) \cdot \log m(n)) = 1 - o(1)$, a $d(n)$-regular $m(n)$-vertex multi-graph generated by the random permutation model is $\Omega(d(n))$-robustly self-ordered and expanding. The fact that the proof extends to a varying degree bound is implicit in the proof of Theorem 6.1, and the higher robustness is obtained by using smaller sets J_i's (see Footnote 33).

   Then, we extend the argument (as done in Step 1 of Remark 6.2) and show that, for any set $\mathcal{G}$ of $t < n$ multi-graphs (which is each $d(n)$-regular and has $m(n)$ vertices), with probability at least $1 - t \cdot \exp(-\Omega(d(n) \cdot \log m(n)) = 1 - o(1)$, a random $d(n)$-regular $m(n)$-vertex multi-graph (as generated above) is both $\Omega(d(n))$-robustly self-ordered and expanding and far from being isomorphic to any multi-graph in $\mathcal{G}$. Here two $d(n)$-regular $m(n)$-vertex multi-graphs are said to be far apart if they disagree on $\Omega(d(n) \cdot m(n))$ vertex-pairs. (Note that the probability that such a random multi-graph is close to being isomorphic to a fixed multi-graph is at most $\exp(-\Omega(d(n) \cdot m(n) \log(m(n)/d(n)))) = o(1/n^2)$, where the last inequality is due to the setting of $m(n)$.)[51]

Note that this multi-graph may have parallel edges and self-loops, but their number can be upper-bounded with high probability. Specifically, for $t = 1/\epsilon$, with probability at least $1 - O(d(n)^t/m(n)^{t-1})$, no vertex has t (or more) self-loops and no vertex is incident to $t+1$ (or more) parallel edges. Hence, omitting all self-loops and all parallel edges leaves us with a simple graph that is both $\Omega(d(n))$-robustly self-ordered (and expanding) and far from being isomorphic to any graph in $\mathcal{G}$.

2. Next, using Step 1, we show that one can construct in poly(n)-time a collection of $n/m(n)$ graphs such that each graph is $d(n)$-regular, has $m(n)$ vertices, is $\Omega(d(n))$-robustly self-ordered and expanding, and the graphs are pairwise far from being isomorphic to one another.

   As in Step 2 of Remark 6.2, this is done by iteratively finding robustly self-ordered $d(n)$-regular $m(n)$-vertex expanding graphs that are far from being isomorphic to all prior ones, while relying on the fact that $m(n)^{d(n)\cdot m(n)} = \text{poly}(n)$ (by the setting of $m(n)$).

3. Lastly, we use the graphs constructed in Step 2 as connected components of an n-vertex graph, and obtain the desired graph.

Note that we have used $m(n) > (\log n)^\epsilon \cdot d(n)$ and $d(n) \cdot m(n) \cdot \log m(n) = \Theta(\log n)$, which is possible if (and only if) $d(n) \leq (\log n)^{0.5-\Theta(\epsilon)}$. ◀

Obtaining strongly connected graphs

The graphs constructed in the proofs of Theorems 10.2 and 10.3 consists of many small connected components; specifically, we obtain n-vertex graphs of maximum degree $d(n)$ with connected components of size $\max(O(d(n)), o(\log n))$ that are $\Omega(d(n))$-robustly self-ordered. We point out that the latter graphs can be transformed into ones with asymptotically maximal expansion (under any reasonable definition of this term), while preserving their maximal degree and robustness parameter (up to a constant factor). This is a consequence of the following general transformation.

▶ **Theorem 10.4** (the effect of super-imposing two graphs). *For every $d, d' : \mathbb{N} \to \mathbb{N}$ and $\rho : \mathbb{N} \to \mathbb{R}$, let G and G' be n-vertex graphs such that G is $\rho(n)$-robustly self-ordered and has maximum degree $d(n)$, and G' has maximum degree $d(n)$. Then, the graph obtained by super-imposing G and G' is $(\rho(n) - d'(n))$-robustly self-ordered and has maximum degree $d(n) + d'(n)$.*

Note that Theorem 10.4 is not applicable to the constructions of bounded-degree graphs obtained in the first part of this paper, because their robustness parameter was a constant smaller than 1. (This is due mostly to Construction 2.3, but also occurs in the proof of Theorem 4.2.)[52] A typical application of Theorem 10.4 may use $d'(n) = \rho(n)/2 \geq 3$. (Recall that $\rho(n) \leq d(n)$ always holds.)

[51] For starters, the probability that an edge that appears in the fixed multi-graph appears in the random graph is $d(n)/m(n)$. Intuitively, these events are sufficiently independent so to prove the claim; for example, we may consider the neighborhoods of the first $m(n)/2$ vertices in the random graph, and an iterative process in which they are determined at random conditioned on all prior choices.

[52] In contrast, the construction of Theorem 10.3, which builds upon the proof of Theorem 6.1, does yield $\Omega(d)$-robustly self-ordered graphs of maximum degree d, for sufficiently large constant d.

Proof. Fixing any permutation μ of the vertex set, note that the contribution of each non-fixed-point of μ to the symmetric difference between $G \cup G'$ and $\mu(G \cup G')$ may decrease by at most $d'(n)$ units due to G'. ◀

Closing the gap between Theorems 10.2 and 10.3

Recall that these theorems left few bounding functions untreated; essentially, these were functions $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \in [(\log n)^{0.499}, O(\log n)^{0.5}]$. We close this gap now.

▶ **Theorem 10.5** (robustly self-ordered graphs for the remaining degree bounds)**.** *For every $d : \mathbb{N} \to \mathbb{N}$ such that $d(n) \in [(\log n)^{1/3}, (\log n)^{2/3}]$ is computable in $\mathrm{poly}(n)$-time, there exists an efficiently constructable family of graphs $\{G_n\}_{n\in\mathbb{N}}$ such that G_n has maximal degree $d(n)$ and is $\Omega(d(n))$-robustly self-ordered.*

In this case, the graphs will consist of connected components of size $2 \log n$.

Proof Sketch. We apply the proof strategy of Theorem 10.2, while using the graphs obtained by combining Theorems 10.2 and 10.4. Specifically, setting $\ell = \log n$, while noting that $d(n) \geq \ell^{1/3} \gg O(\log \ell)^{1/2}$, we use the construction of ℓ-vertex $\Omega(d(n))$-robustly self-ordered graphs of degree at most $d(n)/2$ that are expanding, which is obtained by combining the latter two results. Furthermore, we shall use the fact that these graphs have degree at least $d(n)/200$, and will also use the same construction with degree bound $d(n)/300$. Using these two graphs, we shall construct $n/2\ell$ different ℓ-vertex graphs that are far from being isomorphic to one another, and these will form the connected components of the final n-vertex graph.

Our starting point is the construction of ℓ-vertex graphs that, for some constant $\gamma \in (0, 1)$, are $\gamma \cdot d(n)$-robustly self-ordered and have maximum degree $d(n)/4$ and minimum degree $d(n)/100$. Such graphs are obtained by Theorem 10.2, while setting $m = d(n)/40$. Using Theorem 10.4 (with $d'(n) = \gamma \cdot d(n)/4$), we transform these graphs to ones of maximum degree $d(n)/2$ and asymptotically maximal conductance (i.e., in each of these graphs, the number of edges crossing each cut (in the graph) is at least $\Omega(d(n))$ times the number of vertices in the smaller side (of the cut)). We denote the resulting graph G_1, and apply the same process while setting $m = d(n)/600$ so to obtain a graph of maximum degree $d(n)/300$, denoted G_2.

Next, we connect G_1 and G_2 in various ways so to obtain $n/2\ell$ graphs that are far from being isomorphic to one another. This is done by a small variation on the proof of Theorem 10.2. Specifically, we fix $d(n)/2$ disjoint perfect matchings between the vertices of G_1 and the vertices G_2, and use the error correcting code to determine which of these $\ell \cdot d(n)/2 = \omega(\log n)$ edges to include in the code. More specifically, using an error correcting code with constant rate and constant relative distance and weight, denoted $C : [2^k] \to \{0, 1\}^{\ell \cdot d(n)/2}$, we obtain a collection of $n/2\ell < 2^k$ strongly connected 2ℓ-vertex graphs such that the i^{th} graph consists of copies of G_1 and G_2 that are connected according to the codeword $C(i)$; that is, the $(r, c)^{\text{th}}$ bit of the codeword $C(i)$ (viewed as an $d(n)/2$-by-ℓ matrix) determines whether the c^{th} edge of the r^{th} matching is included in the i^{th} graph. The final n-vertex graph, denoted G, consists of these $n/2\ell$ graphs as its connected components.

The analysis is almost identical to the analysis provided in the proof of Theorem 10.2, since the key facts used there hold here too (although the construction is somewhat different). The key facts are that the degrees of vertices in G_1 and G_2 differ in $\Omega(d(n))$ units, that the relative conductance of the connected components is $\Omega(d(n))$, that G_1 and G_2 are both $\Omega(d(n))$-robustly self-ordered, and that the bipartite graphs (used in the different connected components) are far away from one another. ◀

References

1 L. Babai. Graph isomorphism in quasipolynomial time. In *48th ACM Symposium on the Theory of Computing*, pages 684–697, 2016.

2 E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM Journal on Computing*, 36 (4):889–974, 2006.

3 A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *43rd IEEE Symposium on Foundations of Computer Science*, pages 93–102, 2002.

4 B. Bollobas. The asymptotic number of unlabelled regular graphs. *J. Lond. Math. Soc.*, 26:201–206, 1982.

5 B. Bollobas. Distinguishing vertices of random graphs. *North-Holland Mathematics Studies*, 62:33–49, 1982.

6 J. Bourgain and A. Gamburd. Uniform expansion bounds for cayley graphs of $\mathrm{SL}_2(f_p)$. *Annals of Mathematics,*, pages 625–642, 2008.

7 E. Chattopadhyay, V. Goyal, and X. Li. Non-malleable extractors and codes, with their many tampered extensions. In *48th STOC*, pages 285–298, 2016.

8 M. Cheraghchi and V. Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *11th TCC*, pages 440–464, 2014.

9 B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

10 I. Dinur, O. Goldreich, and T. Gur. Every set in p is strongly testable under a suitable encoding. In *10th ITCS*, pages 30:1–30:17, 2019.

11 I. Dinur and O. Reingold. Assignment-testers: Towards a combinatorial proof of the pcp-theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006.

12 Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *41st STOC*, pages 601–610, 2009.

13 D. Ellis. Lecture 13: The expansion of random regular graphs. *Lecture notes, Algebraic Methods in Combinatorics*, 2011.

14 P. Erdos and A. Renyi. Asymmetric graphs. *Acta Mathematica Hungarica*, 14(3):295–315, 1963.

15 E. Fischer and L. Fortnow. Tolerant versus intolerant testing for boolean properties. *Theory of Computing*, 2(9):173–183, 2006.

16 O. Goldreich. *Introduction to Property Testing.* Cambridge University Press, 2017.

17 O. Goldreich. On testing hamiltonicity in the bounded degree graph model. *ECCC*, TR19-109, 2020.

18 O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, 1998.

19 O. Goldreich, M. Krivelevich, I. Newman, and E. Rozenberg. Hierarchy theorems for property testing. *Computational Complexity*, 21(1):129–192, 2012.

20 O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

21 O. Goldreich and A. Wigderson. Constructing large families of pairwise far permutations: Good permutation codes based on the shuffle-exchange network. *ECCC*, TR20-192, 2020.

22 O. Goldreich and A. Wigderson. Non-adaptive vs adaptive queries in the dense graph testing model. *ECCC*, TR20-160, 2020.

23 O. Goldreich and A. Wigderson. Robustly self-ordered graphs: Constructions and applications to property testing. *ECCC*, TR20-149, 2020.

24 C.S. Greenhill, S. Janson, J.H. Kim, and N.C. Wormald. Permutation pseudographs and contiguity. *Combinatorics, Probability and Computing*, 11:273–298, 2002.

25 S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin (New Series) of the American Mathematical Society*, 43(4):439–561, 2006.

26 J.H. Kim, B. Sudakov, and V.H. Vu. On the asymmetry of random regular graphs and random graphs. *Random Structures & Algorithms*, 21(3-4):216–224, 2002.

27 A. Lubotzky. Discrete groups, expanding graphs and invariant measures. *Progress in mathematics*, 125, 1994.

28 A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

29 E.M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Science*, 25(1):42–65, 1982.

30 M. Parnas, D. Ron, and R. Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Science*, 72(6):1012–1042, 2006.

31 R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

Appendix: On Definitions of Non-Malleable Two-Source Extractor

Recall that Definition 8.1 differs from [7, Def. 1.3] only in the scope of the "tampering functions" f and g. Whereas Definition 8.1 requires *both f and g* to have no fixed-point, in [7, Def. 1.3] it is only required that *either f or g* has no fixed-point. In both cases, the extraction condition is captured by (18) and is applied to the eligible functions f and g (and to random variables X and Y of sufficiently high min-entropy).

We show that Definition 8.1 is strictly weaker than [7, Def. 1.3]. To see this, let $E : \{0,1\}^{n-1} \times \{0,1\}^n \to \{0,1\}^m$ be a non-malleable extractor under [7, Def. 1.3] (say, for constant error and constant deficiency). Actually, we will only use the hypothesis that (18) holds for f and g such that g has no fixed-point (i.e., we make no requirement of f). Now, let $E'(bx', y) = E(x', y)$, where $b \in \{0,1\}$.

1. Clearly, E' violates (18) for $g(y) = y$ and $f(bx') = \bar{b}x'$, where $\bar{b} = 1 - b$, since $E'(f(bx'), g(y)) = E(x', y) = E'(bx', y)$. Hence, E' does not satisfy [7, Def. 1.3].
2. To see that E' satisfies Definition 8.1, consider any f and g that have no fixed-points, and distributions $X = (B, X')$ and Y of low deficiency. Define a random process $F : \{0,1\}^{n-1} \to \{0,1\}^n$ such that $F(x') = f(bx')$, where b is selected according to the residual distribution of B conditioned on $X' = x'$ (i.e., $\Pr[F(x') = z] = \Pr[f(X) = z | X' = x']$). Then, letting $f'(x)$ (resp., $F'(x')$) be the $(n-1)$-bit suffix of $f(x)$ (resp., of $F(x')$), we have

$$\begin{aligned}(E'(X,Y), E'(f(X), g(Y))) &= (E(X',Y), E(f'(BX'), g(Y))) \\ &= (E(X',Y), E(F'(X'), g(Y))),\end{aligned}$$

which is close to $(U_m, E(F'(X'), g(Y)))$, by the hypothesis regrading E (since g has no fixed-point), while also using a convexity argument (for F'). Using $(U_m, E(F'(X'), g(Y))) = (U_m, E'(F(X'), g(Y))) = (U_m, E'(f(X), g(Y)))$, we conclude that $(E'(X,Y), E'(f(X), g(Y)))$ is close to $(U_m, E'(f(X), g(Y)))$.

Barriers for Recent Methods in Geodesic Optimization

W. Cole Franks
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

Philipp Reichenbach
Institut für Mathematik, Technische Universität Berlin, Germany

Abstract

We study a class of optimization problems including matrix scaling, matrix balancing, multidimensional array scaling, operator scaling, and tensor scaling that arise frequently in theory and in practice. Some of these problems, such as matrix and array scaling, are convex in the Euclidean sense, but others such as operator scaling and tensor scaling are *geodesically convex* on a different Riemannian manifold. Trust region methods, which include box-constrained Newton's method, are known to produce high precision solutions very quickly for matrix scaling and matrix balancing (Cohen et. al., FOCS 2017, Allen-Zhu et. al. FOCS 2017), and result in polynomial time algorithms for some geodesically convex problems like operator scaling (Garg et. al. STOC 2018, Bürgisser et. al. FOCS 2019). One is led to ask whether these guarantees also hold for multidimensional array scaling and tensor scaling.

We show that this is not the case by exhibiting instances with exponential *diameter bound*: we construct polynomial-size instances of 3-dimensional array scaling and 3-tensor scaling whose approximate solutions all have doubly exponential condition number. Moreover, we study convex-geometric notions of complexity known as margin and gap, which are used to bound the running times of all existing optimization algorithms for such problems. We show that margin and gap are exponentially small for several problems including array scaling, tensor scaling and polynomial scaling. Our results suggest that it is impossible to prove polynomial running time bounds for tensor scaling based on diameter bounds alone. Therefore, our work motivates the search for analogues of more sophisticated algorithms, such as interior point methods, for geodesically convex optimization that do not rely on polynomial diameter bounds.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Continuous optimization; Mathematics of computing → Combinatorics

Keywords and phrases Geodesically Convex Optimization, Weight Margin, Moment Polytope, Diameter Bounds, Tensor Scaling, Matrix Scaling

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.13

Related Version *Long Version*: `https://arxiv.org/abs/2102.06652`

Funding *Philipp Reichenbach*: acknowledges funding by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 787840).

Acknowledgements The authors thank Jason Altschuler, Peter Bürgisser, Visu Makam, Adam Sawicki and Michael Walter for helpful discussions. Furthermore, the authors thank the anonymous referees for helpful comments and suggestions.

1 Introduction

We study a class of optimization problems ubiquitous in theoretical computer science, machine learning, quantum information theory and statistics. The programs we consider are continuous optimization problems over matrix groups. More precisely, they can be posed as Euclidean norm minimization over the closure of a group orbit. The programs span two historically distinct contexts: In one context, the optimization problems are convex, and in the other they are not convex but rather *geodesically convex* on a suitable manifold.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 13; pp. 13:1–13:54
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The *commutative setting*, in which the underlying group is Abelian, captures matrix scaling, matrix balancing and array scaling, which arise in scientific computing and optimal transport [17, 46]. Such problems fall into the framework of unconstrained geometric programming. Though these problems are convex, there are at least two reasons to study them further. Firstly, they are of such practical importance that speed matters. Naïvely applying powerful algorithms like ellipsoid and interior point methods can be impractically slow. Hence, it is important to understand when faster methods can succeed. Matrix scaling and balancing, in particular, have enjoyed some success stories - there are fast algorithms to obtain high precision solutions [16, 3], and there are more general upper bounds [14]. Secondly, the algorithms developed for the commutative setting are candidates for generalization to our second setting, which takes place in the less well-understood arena of geodesically convex optimization.

The second context, which we call the *noncommutative setting*, arises when the underlying group is non-Abelian. The noncommutative setting captures problems like operator and tensor scaling [25, 13], the quantum marginal problem [11] and statistical estimators such as Tyler's M estimator [23] and maximum likelihood estimates for matrix and tensor normal models [6]. Deciding whether the value of the optimization problem is zero or not is equivalent to deciding a central polynomial identity testing (P.I.T.) problem in invariant theory known as the *null cone* problem. It is hoped that efficient optimization algorithms will result in efficient algorithms for the null-cone problem. One approach to complexity lower bounds, geometric complexity theory, suggests that these P.I.T. problems should be in P [43, 26], and the optimization approach has resulted in polynomial time algorithms in some cases [25, 2]. The optimization problems that arise in the noncommutative setting are not convex in the Euclidean sense, but rather *geodesically convex*, a notion of convexity on a Riemannian manifold. Currently, the only implementable algorithms for geodesically convex optimization are analogues of gradient descent and trust region methods [1, 53, 2]. There are, as of yet, no efficiently implementable geodesically convex counterparts to the interior point or cutting plane methods.

In both the commutative and noncommutative settings, algorithms are typically analysed using two quantities. One is *diameter*, or how far approximate minimizers can be from the origin. The other is a geometric measure of well-conditionedness known as *margin* (or *gap* in the noncommutative case), which has several variants in the literature and appears in two primary ways. Firstly, the smaller the margin, the higher the degree of precision required to decide if the value of the optimization problem is zero or not [12, 30]. Secondly, the larger the margin, the smaller the diameter [48, 50, 12, 14]. In this paper we show the following:

i) In the commutative setting, and in particular for array scaling, approximate minimizers for the functions we study can have doubly exponential condition number. That is, the problems have exponential diameter. As a consequence, popular classes of algorithms such as gradient descent and trust region methods cannot produce high-precision solutions in polynomial time in general. This result applies in the noncommutative setting as well, which provides evidence that even cutting plane methods are unlikely to produce high-precision solutions in polynomial time. This shows it is necessary to develop powerful methods like the interior point method in the geodesically convex setting.

ii) In the commutative and noncommutative settings, we study the margin and gap, respectively, which appear in running time bounds for all existing algorithms. We prove that these measures can be exponentially small in the input size for several problems including array scaling and tensor scaling. In the commutative case, this gives evidence that existing algorithms for array scaling do not run in near-linear time. In the noncommutative case, our results show that margin-based analyses like [12] cannot prove polynomial time guarantees for deciding the null cone problem for tensor scaling using trust region methods.

We use the remainder of the introduction to describe both settings in more detail, state our main results precisely, and discuss previous work. For both the commutative and noncommutative settings, we proceed in the following order. We start with an introduction and motivation of the setting, continue with diameter bounds and afterwards treat bounds on the margin and gap, respectively. We end each setting with a short discussion of the main proof techniques.

1.1 The commutative setting: matrix scaling and its relatives

Matrix scaling and array scaling

Consider the matrix scaling problem: given a nonnegative matrix A, find nonnegative diagonal matrices X, Y such that XAY is doubly stochastic (i.e. has row and column sums equal to one). The matrices, if they exist, can be found by the exceedingly simple and fast alternating minimization method known as *Sinkhorn's algorithm*. It is frequently used in practice, e.g. for quickly approximating the solution to optimal transport problems [17].

Like all other algorithms for matrix scaling, Sinkhorn's algorithm is typically analyzed through optimization. One finds that X and Y are $e^{\operatorname{diag}(x)}, e^{\operatorname{diag}(y)}$, where $x, y \in \mathbb{R}^n$ are solutions to the following optimization problem:

$$\inf_{x,y\in\mathbb{R}^n} \sum A_{ij} e^{x_i+y_j-\bar{x}-\bar{y}} \tag{1}$$

for $\bar{z} := \frac{1}{n}\sum z_i$ (c.f. [35]). Moreover, the infimum is greater than zero if and only if A is *approximately scalable*, i.e. the row and column sums of XAY can be made arbitrarily close to one for X, Y nonnegative, diagonal.

More generally, given a finite set $\Omega \subseteq \mathbb{R}^m$ and a nonnegative function $p : \Omega \to \mathbb{R}_{\geqslant 0}$, define the *capacity* [30] as the value of the unconstrained geometric program

$$\operatorname{cap}(p) := \inf_{x\in\mathbb{R}^m} f_p(x) := \inf_{x\in\mathbb{R}^m} \sum_{\omega\in\Omega} p_\omega e^{\omega\cdot x}. \tag{2}$$

The capacity is positive if and only if zero is in the *Newton polytope* $\operatorname{conv}(\operatorname{supp} p)$. Matrix scaling arises when $m = 2n$ and $\Omega = \{(\varepsilon_i, \varepsilon_j) : i, j \in [n]\}$ for $\varepsilon_k := e_k - \frac{1}{n}\mathbb{1}_n$, where $e_k \in \mathbb{R}^n$ is the k^{th} canonical unit vector and $\mathbb{1}_n \in \mathbb{R}^n$ denotes the all-ones vector. In this case Equation (2) reduces to precisely Equation (1), and $\|\nabla \log f_p(x)\|$ measures the deviation of p from doubly stochastic.

Matrix balancing, in which we instead wish to find a scaling for which the i^{th} row and column sum match, arises when $m = n$ and $\Omega = \{e_i - e_j : i \neq j \in [n]\}$. When $m = 3n$ and $\Omega = \{(\varepsilon_i, \varepsilon_j, \varepsilon_k) : i, j, k \in [n]\}$ we obtain the 3-dimensional *array scaling problem*. In analogy to matrix scaling, in array scaling one has an array p of numbers in $(\mathbb{R}^n_{\geqslant 0})^{\otimes 3}$ and seeks positive vectors $X, Y, Z \in \mathbb{R}^n_{\geqslant 0}$ so that the array q with entries $q_{ijk} = p_{ijk}X_iY_jZ_k$ is *tristochastic*. That is, the sum over every *slice* is equal to one, i.e. $\sum_{j,k} q_{i_0,j,k} = \sum_{i,k} q_{i,j_0,k} = \sum_{i,j} q_{i,j,k_0} = 1$ for all $i_0, j_0, k_0 \in [n]$. If it is possible to satisfy these equations to arbitrary precision we say p is approximately scalable. As for matrix scaling, p is approximately scalable if and only if $\operatorname{cap}(p) > 0$. In the same manner, we obtain d-dimensional array scaling for $m = dn$ and

$$\Omega = \Omega_{n,d} := \{\varepsilon_i \colon i \in [n]\}^d \subseteq (\mathbb{R}^n)^d. \tag{3}$$

We can think of subsets of $\Omega_{n,d}$ as d-uniform, d-partite hypergraphs. Up to an additive shift by $-\frac{1}{n}\mathbb{1}_{nd}$, the elements of $\Omega_{n,d}$ are indicator vectors of the edges in such hypergraphs. For $d = 2$, the matrix p is scalable if and only if the bipartite graph corresponding to $\operatorname{supp} p$ contains a perfect matching, but this is not the case for $d \geqslant 3$ (indeed, d-partite hypergraph matching is NP-hard).

Algorithms for array scaling

Array scaling serves the same role for speeding up multimarginal transport as matrix scaling for optimal transport, and yet again there is a simple and fast alternating minimization algorithm that produces ε-tristochastic scalings in time $O(1/\varepsilon^2)$ [5, 39]. Moreover, algorithms to approximate the capacity arise in varied settings including radial isotropic position [33], entropy maximization [50], and approximate counting [7].

It is natural to ask if there are *high-precision* algorithms for array scaling with $\log(1/\varepsilon)$ dependence on the error and linear or mild dependence on the number of nonzero entries. For matrix scaling and matrix balancing, several works have shown that trust regions and interior point methods *can* obtain such guarantees [16, 3]. Our work is concerned with whether the performance of such algorithms carries over to array scaling and the computation of the capacity in general.

1.1.1 Diameter lower bounds

Guarantees for many iterative algorithms in convex optimization require *diameter bounds*, or bounds on the distance R from the starting point to an ε-approximate solution. Trust region methods, also called *box-constrained Newton's method*, are iterative algorithms that, at each step, move to the best solution within a typically small distance D of the previous solution. By their nature, trust region methods take at least R/D steps to produce an ε-approximate solution. Gradient descent for Lipschitz functions also depends quadratically on a diameter bound, and cutting plane methods typically use diameter bounds to control the volume of a starting region.

Known diameter upper and lower bounds

For matrix scaling and matrix balancing, it has been shown in [16] that one may take $R = O(n \log(w_A/\varepsilon))$, where w_A is the ratio between the sum of the entries of the matrix and the least nonzero entry. For 3-dimensional array scaling, the best upper bound of which we are aware is $R = O(n^{3/2} 2^{6n} \log(1/\varepsilon))$, which follows from the general upper bound of [50] on diameter bounds for unconstrained geometric programming. There is also a diameter bound for array scaling in the multimarginal transport context that is polynomial in the input size assuming the tensor has no nonzero entries [39].

Regarding diameter *lower* bounds, in the context of computing maximum entropy distributions it was shown that there is some bounded set $\Omega \subset \mathbb{Z}^m$ in a $\operatorname{poly}(m)$ size ball such that there are *no* ε-approximate minimizers of norm $\operatorname{poly}(m, \log 1/\varepsilon)$ for f_p as in Equation (2) [50].

Main theorem

Where do the polynomial diameter bounds for matrix scaling (i.e. 2-dimensional array scaling) transition to the superpolynomial diameter bounds for general Ω? We show that this transition takes place in the next simplest problem, the 3-dimensional array scaling problem.

▶ **Theorem 1.1.** *There is an absolute constant $C > 0$ and an array $p_{ijk} \in (\mathbb{R}^n_{\geqslant 0})^{\otimes 3}$ with $O(n)$ nonzero entries, each of bit-complexity $O(n)$, that satisfies the following property. For all $0 < \varepsilon \leqslant \exp(-Cn^2 \log n)$ and $(x, y, z) \in \mathbb{R}^{3n}$, if*

$$f_p(x, y, z) \leqslant \operatorname{cap}(p) + \varepsilon$$

then $\|(x, y, z)\|_2 = \Omega\left(2^{n/3} \log(1/\varepsilon)\right)$.

To emphasize that the difficulties do not lie in an additive vs multiplicative approximation, we remark that our array p has unit sum and $\operatorname{cap}(p) = 1/2$. By a simple duplication trick, the same bound holds for d-dimensional array scaling with $d \geqslant 3$; see Corollary 3.7.

Implications of Theorem 1.1 and relation to the literature

Theorem 1.1 shows that trust region methods for array scaling with polynomial step size cannot provide high-precision solutions in $\operatorname{poly}(n, \log(1/\varepsilon))$ time for $d \geqslant 3$. Moreover, gradient descent on the Lipschitz convex function $\log f_p$ has a bounded step size, and so also cannot provide high precision solutions in polynomial time.

In [50, Section 2.1] the authors ask whether there is Ω whose elements are Boolean (up to an additive shift) with a superpolynomial diameter lower bound. As subsets of $\Omega_{n,d}$ are automatically of this form, we answer their open problem in the affirmative. Our lower bound on $\log R$ is tight up to constant factors by the diameter upper bound from [50] mentioned above; moreover the logarithmic dependence on ε is best possible. Determining the correct constant in the exponent is an interesting open direction. We believe that that the requirement that ε is very small is an artifact of our specific construction and proof strategy, and thus can probably be relaxed significantly.

Lastly, we remark that [14] bounds the diameter for f_p by a polynomial in the *facet gap*, i.e. the minimum distance between an element of $\operatorname{supp} p$ and an affine hull of a facet of the Newton polytope. The construction in Theorem 1.1 has exponentially small facet gap; see Corollary 3.6.

1.1.2 Margins: the geometry of scaling problems

Many computational aspects of the capacity rely on the convex geometry of the finite set $\Omega \subseteq \mathbb{R}^m$. Consider the following quantity, which we call the *margin* of Ω. The margin is the minimum *positive* distance from a convex hull of a subset of Ω to the origin. Formally,

▶ **Definition 1.2** (Margin). *For a finite set $\Omega \subseteq \mathbb{R}^m$, define the margin $\gamma(\Omega)$ by*

$$\gamma(\Omega) := \min\left\{\operatorname{dist}\left(0, \operatorname{conv}(S)\right) \mid S \subseteq \Omega,\ 0 \notin \operatorname{conv}(S)\right\}.$$

We point out that for all considered capacity problems in this paper, the margin is actually the *weight margin* (c.f. [12] and our Definition 4.3) of a certain group representation. For example, the margin for array scaling is the weight margin for tensor scaling. We now discuss how the margin enters in decision problems and diameter bounds.

Margin as a precision parameter for the decision problem

To illustrate how the margin enters the decision problem of whether $\operatorname{cap}(p) > 0$, consider matrix scaling. To certify that the capacity of a matrix is nonzero, we compute ε-doubly stochastic scalings for some ε smaller than the distance to doubly stochastic attained by any matrix that is *not* approximately scalable. This turns out to be precisely $\gamma(\Omega_{n,2})$. More generally, it is a classical fact that for p with support contained in Ω, the gradient $\nabla \log f_p(x)$ can take any value in the Newton polytope of p. Thus, $\operatorname{cap}(p) > 0$ if and only if there is some x with $\|\nabla \log f_p(x)\| \leqslant \gamma(\Omega)$.

For matrix scaling and matrix balancing, it is known that $\gamma(\Omega)$ is on the order of $n^{-3/2}$, despite the exponential number of subsets $S \subseteq \Omega$! This luck can be attributed to the extraordinary geometry of Ω in these cases, whose elements form the rows of a totally unimodular matrix (up to a shift). On the other hand, for d-dimensional array scaling for $n = 2$, the margin $\gamma(\Omega_{2,d})$ is on the order of the margin of the d-dimensional hypercube $\{\pm 1\}^d$, which satisfies $\gamma\big(\{\pm 1\}^d\big) = d^{-\frac{d}{2}(1+o(1))}$ by [4]. However, between the extreme cases $\Omega_{n,2}$ (matrix scaling) and $\Omega_{2,d}$ (the hypercube), very little is known.

Margin and related quantities for diameter bounds

In addition to their role in the decision problem, margins and related quantities can be used to prove diameter bounds for Equation (2). The work [12] proves the diameter bound $\operatorname{poly}(\gamma(\Omega)^{-1}, \log(1/\varepsilon))$. In [50] it is shown that the diameter is polynomial in the logarithm of the minimum nonzero p_ω and a quantity called the *unary facet complexity*. The latter is defined as the maximal length of an integer normal vector of a face of the Newton polytope $\operatorname{conv}(\operatorname{supp} p)$. In the case of d-dimensional arrays, one can use Cramer's rule to crudely bound the unary facet complexity by $(d+1)^{dn}$. In the case when 0 is in the relative interior of the Newton polytope, [48] has shown that there is a minimizer with Euclidean norm $O(\log |\operatorname{supp} p|/\eta)$, where η is the distance from 0 to the boundary of the Newton polytope. The diameter bounds in [48, 50] were used to design ellipsoid methods that are tractable even for $|\operatorname{supp} p|$ very large, and in [14] they were used to bound the running time of interior point methods.

Main theorem

One is led to ask if the margin remains large for array scaling when $d \geqslant 3$. We show that this is not the case. In fact, the margin becomes exponentially small in nd for $d \geqslant 3$. What follows is stated in more detail later in Theorem 2.1.

▶ **Theorem 1.3.** *Let $d \geqslant 3$ and $n \geqslant 2$. Let $\Omega_{n,d} = \{\varepsilon_i : i \in [n]\}^d \subseteq (\mathbb{R}^n)^d$, where $\varepsilon_j := e_j - \frac{1}{n}\mathbb{1}_n$. There exists a constant $C > 0$, independent of n and d, such that $\gamma(\Omega_{n,d}) \leqslant 2^{-Cnd}$.*

That is, there are d-dimensional arrays $p \in (\mathbb{R}^n_{\geqslant 0})^{\otimes d}$ such that the d-tuple of marginals of p is at distance at most 2^{-Cnd} from $\frac{1}{n}(\mathbb{1}_n, \dots, \mathbb{1}_n)$, yet the support of p does not admit an array with uniform marginals, i.e. $\operatorname{cap}(p) = 0$. We note that the support of the array p we construct has $O(nd)$ elements.

Implications of Theorem 1.3 and relation to the literature

We remark that the construction yields a tensor whose Newton polytope has a facet exponentially close to the origin. Therefore, the bound proved in [14] on the number of iterations for interior point methods on 3-tensors is $\Omega(k^{3/2} + k^{1/2}\log(1/\varepsilon))$ for tensors with $O(k)$ nonzero entries.

Theorem 1.3 aligns with existing results showing that the $d > 2$ array case is more complex than the matrix case. Indeed, it is known that the polytope of arrays with uniform marginals, known as the *d-index axial assignment polytope*, has many more vertices when $d \geqslant 3$ and that the vertices can have exponential entries [40]. In contrast, for $d = 2$ this polytope (known as the Birkhoff-von Neumann polytope) has integral vertices by the Birkhoff-von Neumann theorem.

The exponential rate of decay in Theorem 1.3 is tight up to log factors: [12, Theorem 6.10 Item 3] shows that the margin for d-dimensional array scaling is at least $(n\sqrt{d})^{-dn-1}$. It is interesting to ask whether the true bound is $2^{-\Theta(nd)}$ as in our upper bound or $2^{-\Theta(nd(\log n+\log d))}$ as in the lower bound. [4] shows that the latter is correct in the case $n = 2$.

1.1.3 Proof techniques for the commutative setting

We first discuss the techniques for proving our margin bounds. Theorem 1.3 is proven by explicit construction of witness sets $\Gamma_{n,d} \subseteq \Omega_{n,d} := \{\varepsilon_i \colon i \in [n]\}^d$, i.e. $0 \notin \operatorname{conv}(\Gamma_{n,d})$ but zero is exponentially close to $\operatorname{conv}(\Gamma_{n,d})$. This is done by using that $\sum_i n^{-1}\varepsilon_i$ is the unique way to express zero as a convex combination of the ε_i, compare Lemma 2.2, and by heavily exploiting the combinatorics of $\Omega_{n,d}$. For example, in the case $d = 3$ and $n \geqslant 3$ the key combinatorial idea builds on a construction by Kravtsov in [38]. Kravtsov's motivation is to characterize the non-integer vertices of the 3-index axial assignment polytope. He explicitly constructs a certain non-integer vertex with maximal support [38, Theorem 1 with $k = 0$] which has an exponentially small entry.

By definition of the 3-index axial assignment polytope, the support of this vertex corresponds to a subset $S \subseteq \Omega_{n,3}$ with $0 \in \operatorname{conv}(S)$. Removing the element of S corresponding to the small entry in Kravtsov's vertex yields our witness set $\Gamma_{n,3}$ with a convex hull very close to zero. In fact, the whole idea generalizes (in a technical way) whenever $d = 6r - 3$, $r \geqslant 1$ and $n \geqslant 3$, see section 2.3. For $n = 2$ and $d \geqslant 3$, the bound follows from the existing work [4], as mentioned before. While the construction in that work via $\{-1, 1\}$ matrices yields a stronger bound, we provide a different construction of $\{-1, 1\}$ matrices[1], which has the additional property of *freeness*. The latter will prove useful when we adapt Theorem 1.3 to the noncommutative case.

We now discuss the proof of the diameter lower bound, Theorem 1.1. The high level idea is as follows. We first construct a subset $\Omega_0 \subseteq \Omega_{n,3}$ with $0 \in \operatorname{conv}(\Omega_0)$ such that there is another element $\omega \in \Omega_{n,3}$ exponentially close to $\operatorname{conv}(\Omega_0)$, much like our construction of the witness set for small margin discussed above. We then choose an appropriate array p supported on $\Omega_0 \cup \omega$. This suggests that the only approximate minimizers of f_p have a very large component in the direction x from ω to $\operatorname{conv}(\Omega_0)$, because as $y \in \mathbb{R}^m$ tends to a minimizer of f_p the term $e^{y\cdot\omega}$ should vanish compared to the others. This reasoning requires that y is approximately a multiple of x; to enforce this we also ensure that zero is far into the relative interior of $\operatorname{conv}(\Omega_0)$.

The structure of this argument bears some similarity to that in [50], which uses the construction of [4]. The main difference is that the set $\Omega_{n,3}$ in the 3-dimensional array scaling problem consists of vectors of very specific structure: up to an additive shift of $-\frac{1}{n}\mathbb{1}_{3n}$, they are Boolean vectors in $\mathbb{R}^{3n}$ with exactly one nonzero entry among indices in the intervals $[1, n], [n + 1, 2n], [2n + 1, 3n]$. Thus, our construction of Ω_0 must consist of vectors of this special form and not simply bounded integral vectors as in [50]. This is the main additional technical contribution of our construction.

[1] The $(-1, 1)$ matrices from our construction are obtained by replacing all two's in the entries of A_{2r} (6) with -1.

1.2 The noncommutative setting

In the noncommutative setting, we consider a group G acting on $\mathbb{C}^m$.[2] The optimization problem we investigate is given by the *capacity* of a vector $v \in \mathbb{C}^m$ (c.f. [12]):

$$\operatorname{cap}(v) := \inf_{g \in G} f_v(g) := \inf_{g \in G} \|g \cdot v\|^2. \tag{4}$$

For the majority of this paper we work with the *tensor scaling action*, in which $G = \mathrm{SL}(n, \mathbb{C})^d$, the group of d-tuples of complex matrices with determinant one, acts on $v \in (\mathbb{C}^n)^{\otimes d}$ by $(g_1, \ldots, g_d) \cdot v = (g_1 \otimes \cdots \otimes g_d)v$. The corresponding representation is always denoted by $\pi_{n,d}$. Sometimes we also consider the *operator scaling action*, in which $\mathrm{SL}(n)^2$ acts on $v \in (\mathbb{C}^n)^{\otimes 2} \otimes \mathbb{C}^k$ by $(g_1, g_2) \cdot v = (g_1 \otimes g_2 \otimes I_k)v$.

Though Equation (4) looks quite different from Equation (2), one can show that restricting Equation (4) to a certain Abelian subgroup of G (a torus) and making a change of variables yields an instance of Equation (2) (c.f. [12]). For example, restricting the tensor scaling action to the diagonal matrices in G amounts precisely to the array scaling problem from the previous subsection. Likewise, restricting to diagonal matrices in the operator scaling action yields an instance of matrix scaling.

Relation to null cone problem and Geometric Complexity Theory

We study Equation (4) because it is deeply connected to invariant theory through a well-known connection between group orbits and invariant polynomials: zero is in the closure of an orbit of a vector v if and only if every non-constant homogeneous G-invariant polynomial vanishes on v, i.e. if v is in the null-cone. Null-cone membership is a well-studied polynomial identity testing (P.I.T.) problem. One approach to complexity lower bounds, geometric complexity theory, suggests that null-cone membership should be in P [43, 26].

Solving Equation (4) directly allows one to study the null-cone problem through optimization: one notes that $\operatorname{cap}(v) = 0$ if and only if v is in the null cone. In fact, Equation (4) is a geodesically convex optimization problem over a certain Riemannian manifold. Algebraic and optimization-based algorithms have, independently and nearly concurrently, resulted in polynomial time algorithms for nearly the same set of P.I.T. problems arising in invariant theory [22, 43, 25, 34, 20, 2], including the null-cone problem for the operator scaling and simultaneous conjugation action. However, neither approach has succeeded in solving the null-cone problem for the 3-tensor action. Recent degree lower bounds for invariant polynomials for the 3-tensor action pose significant challenges for the algebraic approach [21]. It is natural to ask whether the optimization approach can overcome these challenges.

Algorithms for computing the capacity

A nonzero tensor $w = g \cdot v$ attains the capacity when w has all *quantum marginals* equal to I_n/n. The quantum marginals of a tensor w, analogous to the sums along slices of an array, are the three $n \times n$ matrices $M_1 M_1^\dagger, M_2^\dagger, M_3 M_3^\dagger$ for the $n \times n^2$ matrices M_1, M_2, M_3 known as *flattenings* of $w/\|w\|$. For operator scaling, the capacity is attained when the first two quantum marginals are I_n/n. To compute the capacity, existing algorithms attempt to find g such that the quantum marginals of $g \cdot v$ are all close to I_n/n. There are alternating minimization algorithms that can attain distance ε in time $\operatorname{poly}(n, 1/\varepsilon)$ [25, 13], and for the

[2] Technically we require that G is a reductive group over $\mathbb{C}$ which acts rationally on $\mathbb{C}^m$. All the group actions in this paper satisfy this assumption.

operator scaling this is possible in $\mathrm{poly}(n, \log(1/\varepsilon))$ time [2]. However, for 3-tensor scaling, running time $\mathrm{poly}(1/\varepsilon)$ is not sufficient to efficiently decide null-cone membership, and the only algorithms with $\log(1/\varepsilon)$ dependence on ε have an exponential dependence on n [12].

To explain the increased complexity, we discuss a noncommutative analogue of the Newton polytope known as the *moment polytope*, denoted $\Delta_G(v)$. In particular, $0 \notin \Delta_G(v)$ if and only if v is in the null-cone (i.e. $\mathrm{cap}(v) = 0$).[3] For tensor scaling, the moment polytope is the set of tuples of spectra of the quantum marginals as w ranges over $\overline{G \cdot v}$, shifted by $-\frac{1}{n}(\mathbb{1}_n, \mathbb{1}_n, \mathbb{1}_n)$. The *gap* of the action of G, i.e. the minimum positive distance from 0 to a moment polytope $\Delta_G(v)$, is a noncommutative generalization of the margin. Whereas the operator scaling and simultaneous conjugation actions have polynomially large gaps, we show that the gap for the tensor scaling action is exponentially small. Scaling algorithms amount to outer ε-approximation algorithms for $\Delta_G(v)$, which is why $\mathrm{poly}(1/\varepsilon)$-time algorithms do not suffice to decide null-cone membership. Like for the margin, the smaller gap corresponds to a larger diameter, which is why so far no algorithm has had running time $\mathrm{poly}(n, \log(1/\varepsilon))$.

1.2.1 Diameter lower bound for noncommutative scaling

Here we describe how diameter bounds cause the state-of-the-art algorithms to be slow for the tensor scaling action. We begin by discussing geodesically convex optimization. In general Equation (4) is not convex, but rather *geodesically convex*. That is, G can be viewed as a manifold in such a way that the function $g \mapsto \|g \cdot v\|^2$ is convex along "geodesics" of the form $\gamma(t) = e^{tH} g$ for H Hermitian. The manifold we consider is not exactly G but rather a quotient P of it; we will make this more precise later in Section 4.5. For $G = \mathrm{SL}(n)^d$, the manifold P is the set of tuples of positive-definite matrices with determinant one. P is equipped with the geometry on positive-definite matrices known in statistics as the Fisher-Rao metric, and studied in depth in e.g. [9]. Though we do not need many details of this geometry here, one can think of the distance between $g, h \in G$ as a bound on the logarithms of the singular values of $g^{-1}h$. In particular, the geodesic "ball" of radius R about the identity in G is the intersection of G with the set $\{U \exp(A) : A \text{ Hermitian}, \|A\|_F \leqslant R, U \text{ unitary}\}$. Note that the ball of radius $\sqrt{n}R$ includes all elements of G whose singular values are in $[e^{-R}, e^R]$. [4]

The existing algorithms to compute Equation (4) adapt simple first order methods, such as gradient descent, and second order methods, such as trust regions, to the geodesically convex setting [53, 2, 12]. As in the commutative case, to run in polynomial time such algorithms require that an ε-approximate solution is contained in a geodesic ball of radius $\mathrm{poly}(n^d, \log(1/\varepsilon))$. However, for 3-tensors we have the following diameter lower bound.

▶ **Theorem 1.4** (Noncommutative diameter lower bound). *There is a constant $C > 0$ such that the following holds. For all $\varepsilon \leqslant \exp(-Cn^2 \log n)$, there is a tensor $v = v(\varepsilon) \in (\mathbb{C}^n)^{\otimes 3}$ with $O(n)$ nonzero entries of bit complexity $O(\log n + \log(1/\varepsilon))$, and a geodesic ball $B = B(\varepsilon)$ of radius $\Omega\left(2^{n/3} \log(1/\varepsilon)\right)$ about the identity in $\mathrm{SL}(n)^3$, such that*

$$\inf_{g \in B} \|g \cdot v\|^2 \geqslant \mathrm{cap}(v) + \varepsilon.$$

To emphasize that the difficulties are not caused by requiring additive approximation, we remark that the vector v satisfies $1/4 \leqslant \mathrm{cap}(v) \leqslant 1$ and $1/2 \leqslant \|v\| \leqslant 1$. A duplication trick analogous to Corollary 3.7 yields the same diameter bound for $d \geqslant 3$, but for the action of G simultaneously on a tuple of tensors rather than on a single one. See Corollary 4.24.

[3] Moment polytope membership is an interesting problem in and of itself; for $d = 3$, for generic $v \in (\mathbb{C}^n)^{\otimes 3}$, $\Delta_G(v)$ is the *Kronecker polytope* arising in representation theory and quantum information theory. Deciding membership in this polytope is known to be in NP ∩ coNP but not known to be in P [10].

[4] We define exponentials, Hermitian-ness, and Frobenius norm on tuples by treating them as block diagonal matrices.

Implications of Theorem 1.4 and relation to the literature

Theorem 1.4 shows that trust region methods with constant step size cannot ε-approximate the capacity in $\mathrm{poly}(n, 1/\varepsilon)$ time for 3-tensors. It also shows that cutting plane methods are unlikely to do so. Cutting plane methods, such as ellipsoid, require an exponential bound on the volume of a known region containing an approximate optimizer. This is the case for Rusciano's non-constructive query upper bound for cutting plane methods on manifolds of non-positive curvature [47], which is essentially tight [32][5]. The volume of a ball in the manifold we consider grows exponentially in the radius (see Section 4.5), so this query bound will be exponential. Regarding tightness, the best upper bound known to the authors for the diameter bound in the noncommutative case is $O(n(\sqrt{3}n)^{1+3n}\log(1/\varepsilon))$, which can be deduced from the diameter and margin bounds [12, Proposition 5.6, Theorem 6.10]. This matches our lower bound up to logarithmic factors in the exponent. As with Theorem 1.1, Theorem 1.4 holds only values of ε that are very small (though still of polynomial bit-complexity). It would be very interesting to prove a version of Theorem 1.1 for ε larger than the gap, which is $\exp(-O(n))$. This would imply that trust region methods cannot solve the null-cone problem for the 3-tensor action in polynomial time.

1.2.2 Gaps: the geometry of noncommutative scaling problems

In analogy to the commutative case, one typically attempts to certify $\mathrm{cap}(v) > 0$, i.e. $0 \in \Delta_G(v)$, by finding a tensor $g \cdot v$ such that all the quantum marginals are close to $\frac{1}{n}I_n$. In order to certify $\mathrm{cap}(v) > 0$ their distance to $\frac{1}{n}(I_n, I_n, \ldots, I_n)$ must be at most a certain quantity, which we call the *gap*.

▶ **Definition 1.5** (Gap). *The* gap[6] *for the d-tensor scaling problem is*

$$\gamma_G(\pi_{n,d}) := \min\left\{\mathrm{dist}\left(0, \Delta_G(v)\right) \mid v \in (\mathbb{C}^n)^{\otimes d},\ v \neq 0,\ 0 \notin \Delta_G(v)\right\}.$$

If the gap is exponentially small, high-precision algorithms will be necessary to decide if $\mathrm{cap}(v) > 0$. In operator scaling, the gap is known to be $\Omega(n^{-3/2})$ [29], which explains why we do not need high-precision algorithms for the decision problem in that case. In addition to its role in the decision problem, the inverse of the gap[7] is used to control the diameter bound [12]! In that sense, the presence of a small gap can explain both the need for high precision algorithms and the slowness of existing high-precision algorithms. We show that, indeed, the tensor scaling action has an exponentially small gap for $d \geqslant 3$.

▶ **Theorem 1.6.** *There is a constant $C > 0$ such that for all $d \geqslant 3$ and $n \geqslant 2$, there are non-zero tensors $v \in (\mathbb{C}^n)^{\otimes d}$ such that $0 \notin \Delta_G(v)$ but $\mathrm{dist}(0, \Delta_G(v)) \leqslant 2^{-Cdn}$. That is, the gap for d-tensor scaling satisfies*

$$\gamma_G(\pi_{n,d}) \leqslant 2^{-Cdn}.$$

A detailed statement on bounds for the gap can be found in Theorem 4.11, and we show in Appendix C how to fill in the missing values of n, d to obtain Theorem 1.6. Since the gap is larger than the margin (c.f. Proposition 4.6), Theorem 1.6 is at least as tight as Theorem 1.3, i.e. the exponent Cnd is tight up to an $O(\log n + \log d)$ factor.

[5] [32] applies to the hyperbolic plane, which is a totally geodesic submanifold of the manifold P we consider.

[6] This notion can be defined similarly for any rational representation π of a reductive group G, see Definition 4.3. This definition of the gap is already described in [12].

[7] actually, a smaller quantity known as *weight margin*

Interestingly, for local dimension $n = 2$ [42, Main result] shows that $\mathrm{dist}(0, \Delta_G(v))^2$ for some moment polytope $\Delta_G(v) \not\ni 0$ tends for $d \to \infty$ to the Gamma distribution $\Gamma(1/2, 2d)$, where $2d$ is the rate parameter. Therefore, the witnesses of the exponential behaviour in Theorem 4.11(a) are quite rare. Moreover, the authors numerically found several tensors of format $(\mathbb{C}^2)^{\otimes d}$ with $\mathrm{dist}(0, \Delta_G(v))$ at most $\exp(-d)$; Theorem 1.6 confirms that this exponential behavior is the case for all n and d.

Margin and gap results for other group actions

In addition to the tensor scaling action, we also consider some other actions of groups G of interest in computational invariant theory. The first is the action of the special linear group on the space of homogeneous d-forms $\mathbb{C}[x_1, \ldots, x_n]_d$, in which $G = \mathrm{SL}(n)$ acts by $g \cdot p(x) = p(g^{-1}x)$ for $p \in \mathbb{C}[x_1, \ldots, x_n]_d$. Homogeneous d-forms were among the objects studied earliest in computational invariant theory, and much of the theory was developed to catalogue invariants of the $\mathrm{SL}(n)$ action on forms [52]. Still, deciding null-cone membership for $d = 3$ seems challenging. After extending the definition of the gap to other group actions in Section 4, we explain the difficulty by showing that the gap for this action is also inverse exponential in n as soon as $d \geqslant 3$, see Theorem 4.17. This shows that the diameter bound in [12] becomes exponentially large in n.

The other group action we consider is the action of $\mathrm{SL}(n)^d$ on *quivers* with d vertices. A quiver is a directed multigraph, and a quiver representation is a labelling of the vertex set Q_0 of the quiver with finite-dimensional vector spaces and the edge set Q_1 with a linear map from the vector space at the tail of the edge to the vector space at the head of the edge. Given a quiver representation A with vertices labeled by $\mathbb{C}^{n_x}$ for $x \in Q_0$ and edges $e : x \to y$ labeled with matrices A_e, the group $G = \prod_{x \in Q_0} \mathrm{SL}(n_x)$ acts on A by $(g \cdot A)_e = g_y A g_x^{-1}$. Quiver representations include the operator scaling action, and an action used to bound the Brascamp-Lieb constant in analysis. In Section 4.6 we show that the *(weight) margin* can become exponentially small as the number of vertices grows. For this, we exhibit a quiver with $d - 1$ arrows, d vertices of dimension n and weight margin $O(n^{-d})$, see Theorem 4.25. This bound shows that the diameter bound computed in [12] can become exponentially large in d. Furthermore, when allowing n copies of each arrow in the constructed quiver, i.e. $n(d - 1)$ arrows in total, we can ensure the same bound for the gap, Theorem 4.25.

1.2.3 Proof technique in the noncommutative case: Freeness

Regarding the idea of the proof, we may transfer both the diameter lower bound and the gap upper bound to the commutative case by virtue of the tensors we construct having *free support.*

A tensor has free support if any two distinct $(d - 1)$-dimensional slices of the tensor have disjoint support. This condition ensures that, even after being acted on by any diagonal group elements, the tensor's quantum marginals are all diagonal. This allows us to restrict to the action of the diagonal matrices and thereby reduce to the commutative (array scaling) case. Thus, we may obtain the same bounds on the tensor gap as for the array margin. However, this requires additional care to ensure freeness of our constructions. This is why we cannot naïvely use the construction of [4] for d-tensors with $n = 2$. Regarding the noncommutative diameter bound, we show that for tensors with free support the diameter bound matches that of the commutative problem obtained by restricting to the diagonal. To do this, we project the group elements to the set of diagonal elements, and use the properties of spaces of non-positive curvature to show that this projection moves the point nearer to the origin and decreases the function value.

The idea and the concept of freeness generalize to rational representations of reductive groups [24].[8] The key statement is given in full generality in Proposition 4.8. This proposition is needed to prove bounds on the gap for the action on homogeneous polynomials and for the action on quivers. Interestingly, in [21] the concept of freeness is used in a similar way[9] to prove exponential lower bounds on the degree of invariants for actions on cubic forms and 3-tensors. There, *free* is called *uncramped* and it is used crucially to prove closedness of certain orbits.

Freeness also played a role in the numerical results by Sawicki and Maciążek, which were obtained by applying the algorithm of [41] to several free tensors of local dimension two.

1.3 Organization of the paper

We begin with the commutative case, which is split into the study of the margin in Section 2 and diameter bounds in Section 3. Then we move to the noncommutative case in Section 4. The appendix contains some representation-theoretic background and proofs of technical lemmas, as well as a glossary of notation.

2 The geometry of commutative scaling problems

The purpose of this section is to show the following theorem on the margin of d-dimensional array scaling. Recall that the latter arises for $\Omega_{n,d} := \{\varepsilon_i : i \in [n]\}^d \subseteq (\mathbb{R}^n)^d$.

▶ **Theorem 2.1** (Margin for array scaling). *The margin of $\Omega_{n,d} \subseteq (\mathbb{R}^n)^d$ is bounded as follows.*

(a) *If $n = 2$ and $d \geqslant 3$, then $\gamma\left(\Omega_{2,d}\right) \leqslant 2^{-\frac{d}{2}+1}$.*

(b) *If $n \geqslant 3$ and $d = 3$, then $\gamma(\Omega_{n,3}) \leqslant 2^{-n+1}$.*

(c) *If $n \geqslant 3$ and $d = 6r - 3$ for some integer $r \geqslant 2$, then*

$$\gamma(\Omega_{n,d}) \leqslant \frac{\sqrt{6}}{(n-1)\sqrt{r}}\, 2^{-r(n-1)+1} \leqslant 2^{-r(n-1)+1} = 2^{-\frac{(d+3)(n-1)}{6}+1}.$$

By "padding" the tensors appropriately, one sees that a bound for $\gamma(\Omega_{n,d})$ also applies to $\gamma(\Omega_{n,d+1})$ (see Proposition C.1). Combining this result with Theorem 2.1 above implies Theorem 1.3 from the introduction. The next three subsections each prove one of the parts of Theorem 2.1; the construction for part (a) with $n = 2$ is slightly different and the construction for part (c), $d > 3$ builds on the one for part (b), $d = 3$.

To prove the results, we will frequently use the following simple lemma. Recall that an *affine linear combination* of $v_1, \ldots, v_k \in \mathbb{R}^m$ is $\lambda_1 v_1 + \cdots + \lambda_k v_k$ for $\lambda_i \geqslant 0, \sum_{i=1}^k \lambda_i = 1$. The affine hull $\operatorname{Aff}(S)$ of a set $S \subset \mathbb{R}^m$ is the set of all affine linear combinations of finite subsets of S, or equivalently the affine space (i.e. translate of a subspace) of lowest dimension containing S.

▶ **Lemma 2.2.** *In $\mathbb{R}^n$ we have*

$$\sum_{i=1}^{n} \frac{1}{n}\, \varepsilon_i = 0_n \tag{5}$$

and this is the only affine linear combination of $\varepsilon_1, \ldots, \varepsilon_n$ giving zero.

[8] This concept is also implicitly contained in [49, Lemma 7.1] and can at least be traced back to [18] as *strong orthogonality*.

[9] Indeed, [21, Theorem 6.5] is used to show the vanishing of the moment map at a vector. First, freeness is used as in Proposition 4.8 to ensure that one can restrict to the moment map for the maximal torus. Second, condition (2) of [21, Theorem 6.5] just states that the moment map for the torus action vanishes at the vector.

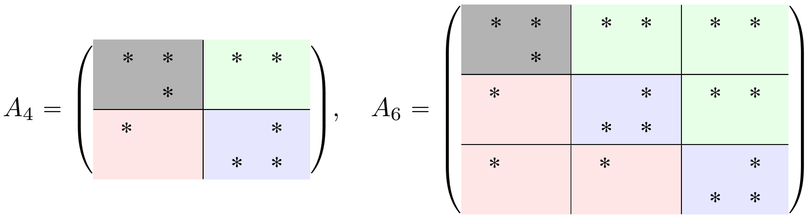


Figure 1 The positions of the ones in A_4 and A_6 are marked by $*$ in the following figure and the cells are colored according to whether they belong to A_2, B_1, B_2 or B_3.

Proof. One calculates directly that $\sum_i \frac{1}{n}\,\varepsilon_i = 0_n$. To show uniqueness of this affine combination, we note that the vectors $e_2, \ldots, e_n, \mathbb{1}_n$ are linearly independent. Thus, $\varepsilon_2, \ldots, \varepsilon_n$ are linearly independent. On the other hand, $\varepsilon_1, \ldots, \varepsilon_n$ are linearly dependent. Therefore $\{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n \mid \sum_i \lambda_i\,\varepsilon_i = 0_n\}$ is a one-dimensional subspace of $\mathbb{R}^n$, which yields the uniqueness of the affine linear combination. ◀

2.1 Local dimension two: the hypercube

In this subsection we prove part (a) of Theorem 2.1 by showing that the margin of $\Omega_{2,d}$ is exponentially small in d. This follows from [4], but we present a new construction which has the additional property of *freeness*, which we discuss later in Section 4. Recall that

$$\Omega_{2,d} = \left\{(\varepsilon_{i_1}, \ldots, \varepsilon_{i_d}) \mid i_1, \ldots, i_d \in [2]\right\} \subseteq \left(\mathbb{R}^2\right)^d.$$

In the following we construct a subset of $\Omega_{2,d}$, which witnesses the exponentially small margin. For this, we construct a matrix with entries in $[2]$, and each row of the matrix will correspond to an element of $\Omega_{2,d}$. For example, the row $(1, 2, 2)$ would correspond to $(\varepsilon_1, \varepsilon_2, \varepsilon_2) \in \Omega_{2,3}$. To do so, we begin with the matrices

$$A_2 := \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix},\ B_1 := \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix},\ B_2 := \begin{pmatrix} 1 & 2 \\ 2 & 2 \end{pmatrix},\ B_3 := \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix},$$

and define recursively

$$A_{2r+2} := \begin{pmatrix} & & & B_1 \\ & A_{2r} & & \vdots \\ & & & B_1 \\ B_2 & \cdots & B_2 & B_3 \end{pmatrix} = \begin{pmatrix} A_2 & B_1 & \cdots & B_1 \\ B_2 & B_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & B_1 \\ B_2 & \cdots & B_2 & B_3 \end{pmatrix} \tag{6}$$

for $r \geqslant 1$. Figure 1 is supplied as a visualization aid.

We remark that the entry of A_{2r} at position (i, j) is independent of r and denote it by $a(i, j)$. We set for $r \geqslant 1$

$$\begin{aligned} \Gamma_{2,2r} &:= \left\{\left(\varepsilon_{a(i,1)}, \varepsilon_{a(i,2)}, \ldots, \varepsilon_{a(i,2r)}\right) \mid i \in [2r]\right\} \subseteq \Omega(\pi_{2,2r}) \subseteq \left(\mathbb{R}^2\right)^{2r}, \\ \Gamma_{2,2r+1} &:= \left\{\left(\varepsilon_{a(i,1)}, \varepsilon_{a(i,2)}, \ldots, \varepsilon_{a(i,2r)}, \varepsilon_{\chi(i)}\right) \mid i \in [2r]\right\} \subseteq \Omega(\pi_{2,2r+1}) \subseteq \left(\mathbb{R}^2\right)^{2r+1}, \end{aligned}$$

where $\chi\colon \mathbb{N} \to \{1, 2\}$, $i \mapsto i \mod 2$. That is, $\Gamma_{2,2r}$ is the subset of $\Omega_{2,2r}$ induced by the rows of A_{2r} and $\Gamma_{2,2r+1}$ is obtained by alternatingly appending ε_1 or ε_2 to the $2r$-many elements of $\Gamma_{2,2r}$.

▶ **Lemma 2.3.** *For $r \geqslant 1$ it holds that $0 \notin \operatorname{Aff}(\Gamma_{2,2r})$ and $0 \notin \operatorname{Aff}(\Gamma_{2,2r+1})$.*

Proof. By construction, $0 \in \operatorname{Aff}(\Gamma_{2,2r+1})$ implies $0 \in \operatorname{Aff}(\Gamma_{2,2r})$, so it suffices to prove $0 \notin \operatorname{Aff}(\Gamma_{2,2r})$. We proceed by induction on $r \geqslant 1$. For $r = 1$, it is clear that $0 \notin \operatorname{Aff}(\Gamma_{2,2}) \subseteq \mathbb{R}^2 \times \{\varepsilon_1\}$. Now assume that $0 \notin \operatorname{Aff}(\Gamma_{2,2r})$. For the sake of contradiction, let

$$\sum_{i=1}^{2r+2} \lambda_i \left(\varepsilon_{a(i,1)}, \varepsilon_{a(i,2)}, \ldots, \varepsilon_{a(i,2r+2)}\right) = 0 \in \left(\mathbb{R}^2\right)^{2r+2} \tag{7}$$

be an affine linear combination of $\Gamma_{2,2r+2}$. Then equation (7) gives in each of the $(2r+2)$-many $\mathbb{R}^2$-components the affine linear combination $2^{-1}(\varepsilon_1 + \varepsilon_2) = 0$, by Lemma 2.2. Considering the scalar factor of ε_1 in the first, the penultimate and the last $\mathbb{R}^2$-component respectively, we conclude

$$\underbrace{\sum_{j=1}^{r+1} \lambda_{2j-1}}_{\text{first}} = \frac{1}{2} = \underbrace{\lambda_{2r+2} + \sum_{j=1}^{r} \lambda_{2j-1}}_{\text{penultimate}} = \frac{1}{2} = \underbrace{\lambda_{2r+2} + \sum_{j=1}^{r+1} \lambda_{2j-1}}_{\text{last}}$$

by construction of A_{2r+2}. Hence, $\lambda_{2r+2} = 0$ using the first and last component. Furthermore, the first and penultimate column give $\lambda_{2r+1} = \lambda_{2r+2} = 0$. Therefore, the first $2r$-many components in Equation (7) show $0 \in \operatorname{Aff}(\Gamma_{2,2r})$, which contradicts our induction hypothesis. ◀

▶ **Lemma 2.4.** *For $r \geqslant 1$ it holds that $\operatorname{dist}(0, \operatorname{conv}(\Gamma_{2,2r})) \leqslant 2^{-r+\frac{1}{2}}$ and $\operatorname{dist}(0, \operatorname{conv}(\Gamma_{2,2r+1})) \leqslant 2^{-r+\frac{1}{2}}$.*

Proof. We first prove the inequality for $\operatorname{conv}(\Gamma_{2,2r})$. For $i \in [2r]$ let $\omega_i := \left(\varepsilon_{a(i,1)}, \ldots, \varepsilon_{a(i,2r)}\right) \in \left(\mathbb{R}^2\right)^{2r}$ be the weight in $\Gamma_{2,2r}$ that corresponds to the i^{th} row of A_{2r}. Consider the convex combination

$$(x_1, \ldots, x_{2r}) := 2^{-r}(\omega_{2r-1} + \omega_{2r}) + \sum_{l=1}^{r-1} 2^{-l-1}(\omega_{2l-1} + \omega_{2l}) \in \left(\mathbb{R}^2\right)^{2r}. \tag{8}$$

Note that $x_i \in \mathbb{R}^2$. We will argue that $(x_1, \ldots, x_{2r}) = 2^{-r+1}(0_2, \ldots, 0_2, \varepsilon_1)$. Since x is a convex combination of the elements in $\Gamma_{2,2r}$, the statement then follows from $\|\varepsilon_1\| = \sqrt{2}^{-1}$.

We consider A_{2r} like in its construction (6) as a $r \times r$ block matrix with block entries being 2×2 matrices. For $m \in [r]$ the two weights ω_{2m-1} and ω_{2m} correspond to the m^{th} block row of A_{2r} and have the same scalar factor in (8). Hence, whenever for $i \in [2r]$ the i^{th} column of the m^{th} block row of A_{2k} contains exactly one entry equal to one (and so the other entry equals two), then the contribution of ω_{2m-1} and ω_{2m} to x_i cancels due to $\varepsilon_1 + \varepsilon_2 = 0_2$. In particular, in (8) all contributions of block entries equal to B_1 cancel. Therefore the last column of A_{2r} gives

$$x_{2r} = 2^{-r}(\varepsilon_1 + \varepsilon_1) = 2^{-r+1}\varepsilon_1.$$

Furthermore, $x_1 = x_3 = \ldots = x_{2r-1} = 0_2$ using that also the first columns of A_2, of B_2 and of B_3 contain exactly one entry equal to one. For $r = 1$ we are done. If $r \geqslant 2$, then reading off the second column of A_{2r}, we find

$$x_2 = \underbrace{2^{-2}(\varepsilon_1 + \varepsilon_1)}_{\text{first block row}} + \underbrace{2^{-r}(\varepsilon_2 + \varepsilon_2)}_{\text{last block row}} + \sum_{l=2}^{r-1} \underbrace{2^{-l-1}(\varepsilon_2 + \varepsilon_2)}_{\text{middle rows}} = 2^{-1}(\varepsilon_1 + \varepsilon_2) = 0_2.$$

Analogously, as B_1 does not contribute we compute for $j = 2, 3, \dots, r-1$ that

$$x_{2j} = \underbrace{2^{-j-1}(\varepsilon_1 + \varepsilon_1)}_{j^{th}\text{ block row}} + \underbrace{2^{-r}(\varepsilon_2 + \varepsilon_2)}_{\text{last block row}} + \sum_{l=j+1}^{r-1} \underbrace{2^{-l-1}(\varepsilon_2 + \varepsilon_2)}_{\text{in between rows}} = 2^{-j}(\varepsilon_1 + \varepsilon_2) = 0_2,$$

because the second columns of B_2 and B_3 are, respectively, $(2,2)^T$ and $(1,1)^T$. This proves the inequality in the case $\Gamma_{2,2r}$.

By construction, for $\Gamma_{2,2r+1}$ the same convex combination works, because the last $\mathbb{R}^2$-component does not contribute as the entries of the weights alternate between ε_1 and ε_2. ◀

Finally, Lemma 2.3 and Lemma 2.4 together yield Theorem 2.1(a), noting that for odd $d = 2r + 1$ one has $-r + 1/2 = -(d/2) + 1$.

2.2 3-tensors

The main goal of this section is to show that the margin of $\Omega_{n,3}$ is exponentially small in n, i.e. to show Theorem 2.1(b). To do so, we set

$$\mathfrak{W}_n := \bigcup_{s=2}^{n} \{(s,1,s), (s,s,1), (s-1,s,s)\} \subseteq [n] \times [n] \times [n] \tag{9}$$

and consider the corresponding subset

$$\Gamma_{n,3} := \big\{(\varepsilon_i, \varepsilon_j, \varepsilon_k) \mid (i,j,k) \in \mathfrak{W}_n\big\} \subseteq \Omega_{n,3}. \tag{10}$$

The key combinatorial idea, which is presented in the following lemma, is due to [38, Theorem 1 with $k = 0$].[10] According to [38] the special case $k = 0$ is already contained in [37, Theorem 9].

▶ **Lemma 2.5.** *Let $n \geqslant 3$. For $(i,j,k) \in [n]^3 \backslash \big(\mathfrak{W}_n \cup \{(1,1,1)\}\big)$ set $\lambda_{i,j,k} := 0$. Moreover, define*

$$\lambda_{1,1,1} := 2^{-n+1}, \quad \lambda_{1,2,2} := 1 - 2^{-n+1}, \quad \lambda_{n,1,n} = \lambda_{n,n,1} := 2^{-1}$$

and for $s = 2, 3, \dots, n-1$

$$\lambda_{s,1,s} = \lambda_{s,s,1} := 2^{-n+s-1}, \quad \lambda_{s,s+1,s+1} := 1 - 2^{-n+s}.$$

Then the following equations hold:

$$\left(\forall i \in [n] : \sum_{j,k=1}^{n} \lambda_{i,j,k} = 1\right), \left(\forall j \in [n] : \sum_{i,k=1}^{n} \lambda_{i,j,k} = 1\right), \left(\forall k \in [n] : \sum_{i,j=1}^{n} \lambda_{i,j,k} = 1\right). \tag{11}$$

In particular, $\sum_{i,j,k} \lambda_{i,j,k} = n$.

Proof. This is [38, Theorem 1 with $k = 0$]. Alternatively, the statement can be checked by straightforward computation. ◀

[10] In [38] Kravtsov extensively studies so-called complete r-noninteger vertices (r-CNVs) of the three-index axial assignment polytope. For $k \in \{0, 1, \dots, n-2\}$, [38, Theorem 1] states explicitly a $(3n-2-k)$-CNV, among these we use the $(3n-2)$-CNV (i.e. $k = 0$). Moreover, [38, Theorem 2] states that such r-CNVs of the three-index axial assignment polytope actually only occur for $r \in \{2n, 2n+1, \dots, 3n-2\}$, and the later theorems in [38] fully characterize the r-CNVs and study their combinatorial properties.

▶ **Example 2.6.** *To visualize Lemma 2.5 it is helpful to consider the slices Λ_i given by $(\Lambda_i)_{j,k} = \lambda_{i,j,k}$. For $n = 4$ one has*

$$\Lambda_1 = \frac{1}{8}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \Lambda_2 = \frac{1}{8}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\Lambda_3 = \frac{1}{8}\begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}, \quad \Lambda_4 = \frac{1}{8}\begin{pmatrix} 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{pmatrix}.$$

For $n = 5$ one has

$$\Lambda_1 = \frac{1}{16}\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \Lambda_2 = \frac{1}{16}\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\Lambda_3 = \frac{1}{16}\begin{pmatrix} 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \Lambda_4 = \frac{1}{16}\begin{pmatrix} 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 \end{pmatrix}, \quad \Lambda_5 = \frac{1}{16}\begin{pmatrix} 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

▶ **Lemma 2.7.** *For $n \geqslant 3$, it holds that* $\operatorname{dist}\big(0, \operatorname{conv}(\Gamma_{n,3})\big) \leqslant 2^{-n+1}$.

Proof. Define $\lambda_{i,j,k} \geqslant 0$ for all $i, j, k \in [n]$ as in Lemma 2.5. Note that $\sum_{i=1}^{n} \varepsilon_i = 0$; thus Lemma 2.5 implies

$$\sum_{i,j,k} \lambda_{i,j,k}(\varepsilon_i, \varepsilon_j, \varepsilon_k) = 0_{3n}, \quad \text{equivalently} \quad -2^{-n+1}(\varepsilon_1, \varepsilon_1, \varepsilon_1) = \sum_{(i,j,k)\in\mathfrak{W}_n} \lambda_{i,j,k}(\varepsilon_i, \varepsilon_j, \varepsilon_k).$$

Normalizing the latter equation we obtain

$$x := -\frac{1}{c\,2^{n-1}}(\varepsilon_1, \varepsilon_1, \varepsilon_1) \in \operatorname{conv}(\Gamma_{n,3}), \quad \text{where} \quad c := \sum_{(i,j,k)\in\mathfrak{W}_n} \lambda_{i,j,k} = n - 2^{-n+1} \geqslant \sqrt{3}.$$

Finally, $\|\varepsilon_1\|^2 \leqslant 1$ implies $\|x\| \leqslant c^{-1}2^{-n+1}\sqrt{3} \leqslant 2^{-n+1}$. ◀

To finish the proof of Theorem 2.1(b) we are left to show $0 \notin \operatorname{conv}(\Gamma_{n,3})$. We actually prove the stronger statement $0 \notin \operatorname{Aff}(\Gamma_{n,3})$.

▶ **Lemma 2.8.** *The zero vector is not contained in the affine hull of $\Gamma_{n,3}$.*

Proof. For a proof by contradiction we assume $0 \in \operatorname{Aff}(\Gamma_{n,3})$. Then there exist $a_s, b_s, c_s \in \mathbb{R}$ for $s = 2, 3, \ldots, n$ such that $\sum_s a_s + b_s + c_s = 1$ and

$$\sum_{s=2}^{n} \big(a_s(\varepsilon_s, \varepsilon_1, \varepsilon_s) + b_s(\varepsilon_s, \varepsilon_s, \varepsilon_1) + c_s(\varepsilon_{s-1}, \varepsilon_s, \varepsilon_s) \big) = (0_n, 0_n, 0_n) \in (\mathbb{R}^n)^3.$$

In each of the three $\mathbb{R}^n$-components we obtain 0_n as an affine linear combination of $\varepsilon_1, \ldots, \varepsilon_n$. Applying Lemma 2.2 to the coefficient of ε_{s-1} in the first component, respectively to the coefficient of ε_s in the second and third component yields

$$a_{s-1} + b_{s-1} + c_s = \frac{1}{n} \quad \text{for } s = 2, 3, \ldots, n \tag{12}$$

respectively

$$b_s + c_s = a_s + c_s = \frac{1}{n} \quad \text{for } s = 2, 3, \ldots, n \tag{13}$$

where we necessarily set $a_1 = b_1 := 0$. Equation (12) for $s = 2$ is $c_2 = n^{-1}$ and hence $a_2 = b_2 = 0$ by (13) for $s = 2$. But now (12) for $s = 3$ gives $c_3 = n^{-1}$ and we can proceed inductively to conclude $c_s = n^{-1}$ and $a_s = b_s = 0$ for all $s = 2, 3, \ldots, n$. This gives the contradiction $1 = \sum_{s=2}^{n}(a_s + b_s + c_s) = \frac{n-1}{n}$, so we must have $0 \notin \operatorname{Aff}(\Gamma_{n,3})$. Another contradiction arises when one applies Lemma 2.2 to the coefficient ε_n in the first component, which yields $a_n + b_n = n^{-1}$. ◀

2.3 *d*-tensors

In this subsection we show that the margin of $\Omega_{n,d}$ is inverse exponential in nd for $n, d \geqslant 3$, proving part (c) of Theorem 2.1.

Let us give some intuition for our construction. The main idea is to recycle the construction from the previous subsection for some multiple of n, i.e. considering $\mathfrak{W}_{rn}$ for $r \geqslant 2$. Thereby, the main challenge is to ensure that the constructed subset of $\Omega_{n,d}$ does not contain zero in its convex hull. We can try to extend the elements of $\Omega_{n,3}$ to elements of $\Omega_{n,d}$. One natural idea is duplicate each component $d/3$ times, i.e. when $d = 6$ the vector $(\varepsilon_i, \varepsilon_j, \varepsilon_k) \in \Omega_{n,3}$ becomes $(\varepsilon_i, \varepsilon_i, \varepsilon_j, \varepsilon_j, \varepsilon_k, \varepsilon_k) \in \Omega_{n,6}$. However, we need a subset of $\Omega_{n,d}$ with rn many elements to imitate the construction from the previous subsection. We still extend the elements of $\Omega_{n,3}$ in this way, but will additionally "shift" and "twist" by some functions $\sigma_1, \ldots, \sigma_{2r-1} \colon [rn] \to [n]$, so that the elements of our set will look like

$$\left(\varepsilon_{\sigma_1(i)}, \ldots, \varepsilon_{\sigma_{d/3}(i)}, \varepsilon_{\sigma_1(j)}, \ldots, \varepsilon_{\sigma_{d/3}(j)}, \varepsilon_{\sigma_1(k)}, \ldots, \varepsilon_{\sigma_{d/3}(k)}\right)$$

for $d/3 = 2r - 1$ and (i, j, k) in $\mathfrak{W}_{rn}$. We now set about choosing the functions σ_k. For this, let $n \geqslant 3$ and fix a natural number $r \geqslant 2$. It is convenient to use an *adjusted* modulo n function $\operatorname{mod}'\ n$ that takes values in $[n]$, i.e. instead of zero it outputs n. For $i \in [r]$ we consider

$$\sigma_i \colon [rn] \to [n], \quad j \mapsto \left\lceil \frac{j + (i-1)}{r} \right\rceil \quad \operatorname{mod}'\ n$$

$$\sigma_{r+i} := \sigma_1 \circ (r - i + 1 \quad r + 1) \colon [rn] \to [n]$$

where $(r - i + 1 \quad r + 1)$ denotes the corresponding transposition in the symmetric group of $[rn]$.[11] We only need the first $2r - 1$ of these functions and combine them to obtain

$$\sigma \colon [rn] \to [n]^{2r-1}, \quad j \mapsto \big(\sigma_1(j), \sigma_2(j), \ldots, \sigma_{2r-1}(j)\big).$$

▶ **Example 2.9.** *For $r = 3$ the functions $\sigma_1, \sigma_2, \ldots, \sigma_6$ are sketched by the following table.*

j	1	2	3	4	5	6	$\cdots$	$3n-5$	$3n-4$	$3n-3$	$3n-2$	$3n-1$	$3n$
σ_1	1	1	1	2	2	2	$\cdots$	$n-1$	$n-1$	$n-1$	n	n	n
σ_2	1	1	2	2	2	3	$\cdots$	$n-1$	$n-1$	n	n	n	1
σ_3	1	2	2	2	3	3	$\cdots$	$n-1$	n	n	n	1	1
σ_4	1	1	2	1	2	2	$\cdots$	$n-1$	$n-1$	$n-1$	n	n	n
σ_5	1	2	1	1	2	2	$\cdots$	$n-1$	$n-1$	$n-1$	n	n	n
σ_6	2	1	1	1	2	2	$\cdots$	$n-1$	$n-1$	$n-1$	n	n	n

For $r = 3$ and $n = 5$ the functions $\sigma_1, \sigma_2, \ldots, \sigma_6$ are given by the following table.

[11] We stress that we always take σ_1 (and *not* σ_i) to define σ_{r+i}.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
σ_1	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5
σ_2	1	1	2	2	2	3	3	3	4	4	4	5	5	5	1
σ_3	1	2	2	2	3	3	3	4	4	4	5	5	5	1	1
σ_4	1	1	2	1	2	2	3	3	3	4	4	4	5	5	5
σ_5	1	2	1	1	2	2	3	3	3	4	4	4	5	5	5
σ_6	2	1	1	1	2	2	3	3	3	4	4	4	5	5	5

▶ **Remark 2.10.** *By construction, each element of $[n]$ is attained exactly r-times by σ_k, $k \in [2r-1]$. Moreover, the definition of $\sigma_1, \ldots, \sigma_r$ yields that σ is injective.*

For $i, j, k \in [rn]$ we introduce the short-hand

$$\varepsilon_{\sigma(i)} := \big(\varepsilon_{\sigma_1(i)}, \varepsilon_{\sigma_2(i)}, \ldots, \varepsilon_{\sigma_{2r-1}(i)}\big) \in (\mathbb{R}^n)^{2r-1}$$

$$\varepsilon_{\sigma(i),\sigma(j),\sigma(k)} := \big(\varepsilon_{\sigma_1(i)}, \ldots, \varepsilon_{\sigma_{2r-1}(i)}, \varepsilon_{\sigma_1(j)}, \ldots, \varepsilon_{\sigma_{2r-1}(j)}, \varepsilon_{\sigma_1(k)}, \ldots, \varepsilon_{\sigma_{2r-1}(k)}\big) \in (\mathbb{R}^n)^{6r-3}$$

and we set[12]

$$\mathfrak{J}_r := \big\{(s,1,s), (s,s,1) \mid s = 2,3,\ldots,r\big\} \subseteq \mathbb{Z}^3.$$

In the following we show that the convex hull of the set

$$\Gamma_{n,6r-3} = \big\{\varepsilon_{\sigma(i),\sigma(j),\sigma(k)} \mid (i,j,k) \in \mathfrak{W}_{rn} \backslash \mathfrak{J}_r\big\} \subseteq \Omega_{n,6r-3} \subseteq \Big((\mathbb{R}^n)^{2r-1}\Big)^3$$

does not contain the zero vector, but is very close to it.

▶ **Lemma 2.11.** *For $n \geqslant 3$ and $r \geqslant 2$ it holds that $0 \notin \operatorname{Aff}(\Gamma_{n,6r-3})$.*

Below we give the proof in the special case $r = 3$, in which all main ideas of the general proof become apparent and visible. The proof for the general statement is given in Appendix D and certainly looks technical at a first encounter. Therefore, we strongly suggest that the reader first reads the proof for $r = 3$ below.

Proof of Lemma 2.11 for $r = 3$. For the sake of contradiction assume that $0 \in \operatorname{Aff}(\Gamma_{n,15})$. Then there are coefficients $a_s, b_s, c_s \in \mathbb{R}$, where $2 \leqslant s \leqslant 3n$, such that $a_2 = a_3 = b_2 = b_3 = 0$, $\sum_s (a_s + b_s + c_s) = 1$ and

$$\sum_{s=2}^{3n} \big(a_s\, \varepsilon_{\sigma(s),\sigma(1),\sigma(s)} + b_s\, \varepsilon_{\sigma(s),\sigma(s),\sigma(1)} + c_s\, \varepsilon_{\sigma(s-1),\sigma(s),\sigma(s)}\big) = 0 \in (\mathbb{R}^n)^{15}. \tag{14}$$

The bulk of our work will consist of proving the equations

$$b_2 + c_2 = b_3 + c_3 = \ldots = b_{3n} + c_{3n} \tag{15}$$

$$a_2 + c_2 = a_3 + c_3 = \ldots = a_{3n} + c_{3n}. \tag{16}$$

[12] One could suggest to consider the set $\{\varepsilon_{\sigma(i),\sigma(j),\sigma(k)} \mid (i,j,k) \in \mathfrak{W}_{rn}\}$, but this still won't ensure that zero is not in the convex hull. The intuition behind is, that $\Gamma_{n,3}$ from the last section is "nearly at the limit", i.e. $0 \notin \operatorname{conv}(\Gamma_{n,3})$ but $0 \in \operatorname{conv}(\Gamma_{n,3} \cup \{(\varepsilon_1, \varepsilon_1, \varepsilon_1)\})$. Now the function σ "introduces $2r-2$ additional linear relations" as $\varepsilon_{\sigma(i)} \in (\mathbb{1}_n^\perp)^{2r-1}$, since the orthogonal complement $\mathbb{1}_n^\perp \subseteq \mathbb{R}^n$ has codimension one while $(\mathbb{1}_n^\perp)^{2r-1} \subseteq (\mathbb{R}^n)^{2r-1}$ has codimension $2r-1$. Thus, it is reasonable to remove $2r-2$ many elements from $\mathfrak{W}_{rn}$.

From here we will derive a contradiction. We now set about proving Equations (15) and (16). Rewrite the left-hand-side of Equation (14) as the collection for $k \in [5]$ of the following affine linear combinations of $\varepsilon_1, \dots, \varepsilon_n$ in $\mathbb{R}^n$:

$$\sum_{s=2}^{3n} \left(a_s\, \varepsilon_{\sigma_k(s)} + b_s\, \varepsilon_{\sigma_k(s)} + c_s\, \varepsilon_{\sigma_k(s-1)}\right) = 0 \tag{17}$$

$$\sum_{s=2}^{3n} \left(a_s\, \varepsilon_{\sigma_k(1)} + b_s\, \varepsilon_{\sigma_k(s)} + c_s\, \varepsilon_{\sigma_k(s)}\right) = 0 \tag{18}$$

$$\sum_{s=2}^{3n} \left(a_s\, \varepsilon_{\sigma_k(s)} + b_s\, \varepsilon_{\sigma_k(1)} + c_s\, \varepsilon_{\sigma_k(s)}\right) = 0. \tag{19}$$

If we expand each expression as an affine linear combination of the ε_l, then by Lemma 2.2 the coefficient of ε_l must be n^{-1} for all $l \in [n]$. Translating this for equation (17) with $k = 2$, $l = 2, \dots, n$ and using Example 2.9 we obtain

$$(a_{m-3} + a_{m-2} + a_{m-1}) + (b_{m-3} + b_{m-2} + b_{m-1}) + (c_{m-2} + c_{m-1} + c_m) = \frac{1}{n} \tag{20}$$

for $m = 6, 9, 12, \dots, 3n$. A similar calculation for $k = 1, 3$ and $l = 2, \dots, n$ shows Equation (20) holds for all $5 \leqslant m \leqslant 3n+1$, where we set $c_{3n+1} := 0$.

Similarly for Equation (18) with $l = 2, \dots, n$ and $k = 1, 2, 3$ we obtain for $4 \leqslant m \leqslant 3n$ that

$$(b_{m-2} + c_{m-2}) + (b_{m-1} + c_{m-1}) + (b_m + c_m) = \frac{1}{n} \tag{21}$$

and the same equations with "b" replaced by "a" when considering Equation (19).

In the following we prove Equation (15). Subtracting (21) from (21) with values of m differing by one, we deduce that

$$\begin{aligned} & b_2 + c_2 = b_5 + c_5 = \ldots = b_{3n-1} + c_{3n-1} \\ & b_3 + c_3 = b_6 + c_6 = \ldots = b_{3n} + c_{3n}, \\ \text{and} \quad & b_4 + c_4 = b_7 + c_7 = \ldots = b_{3n-2} + c_{3n-2}. \end{aligned}$$

Next we deduce Equation (15) by showing $b_2 + c_2 = b_3 + c_3 = b_4 + c_4$.

To do so, we apply Lemma 2.2 to (18) for the coefficient of ε_2 using Example 2.9, which yields for $k = 4, 5$ the equations

$$(b_3 + c_3) + (b_5 + c_5) + (b_6 + c_6) = \frac{1}{n} \tag{22}$$

$$(b_2 + c_2) + (b_5 + c_5) + (b_6 + c_6) = \frac{1}{n} \tag{23}$$

respectively. Subtracting the two shows $b_2 + c_2 = b_3 + c_3$, and we have $b_3 + c_3 = b_4 + c_4$ via subtracting (22) from (21) for $m = 6$. This completes the proof of Equation (15); using Equation (19) we similarly deduce Equation (16).

To get a contradiction we show that $a_s = b_s = c_s = 0$ for all $s = 2, 3, \dots, 3n$. For this, we set $a := \sum_s a_s$ and $b := \sum_s b_s$, and recall that we have defined $a_2 = a_3 = b_2 = b_3 = 0$. This time we use Lemma 2.2 applied to the coefficient of ε_1 in (17), in (18) and in (19) respectively for $k = 1$ to get

$$c_2 + c_3 + c_4 = \frac{1}{n}, \qquad a + c_2 + c_3 = \frac{1}{n} \qquad \text{and} \qquad b + c_2 + c_3 = \frac{1}{n} \tag{24}$$

respectively. We deduce from these three equations that $a = b = c_4$. Furthermore, $b_2 = b_3 = 0$ shows that (21) for $m = 4$ is $b_4 + (c_2 + c_3 + c_4) = n^{-1}$. Subtracting from the latter the left-hand equation in (24) yields $b_4 = 0$. Similarly, $a_4 = 0$ follows from $a_2 = a_3 = 0$ and the analogous equation of (21) with a's replaced by b's.

Now, (20) for $m = 5$ simplifies to $c_3 + c_4 + c_5 = n^{-1}$. Thus, $c_2 = c_5$ with (24) and therefore $a_5 = b_5 = 0$ by (15), (16) and $a_2 = b_2 = 0$. This simplifies (20) for $m = 6$ to $c_4 + c_5 + c_6 = n^{-1}$. Hence, $c_3 = c_6$ as we also have $c_3 + c_4 + c_5 = n^{-1}$ and we get via (15) and (16) that $a_6 = b_6 = 0$. The latter in turn shows that (20) for $m = 7$ becomes $c_5 + c_6 + c_7 = n^{-1}$, so $c_4 = c_7$ and $a_7 = b_7 = 0$ by, again, (15) and (16).

It should have become apparent that we can proceed inductively in the same manner with (20) for $m = 5, \ldots, 3n + 1$; thereby using (15) and (16) to deduce $a_s = b_s = 0$ for all $s = 2, 3, \ldots, 3n$. In particular, $a = b = c_4 = 0$. Finally, Equation (15) implies $c_4 = c_s$ for all $s = 2, 3, \ldots, 3n$, which gives the desired contradiction. ◀

We finish the proof of part (*c*) of Theorem 2.1 by showing the following Lemma.

▶ **Lemma 2.12.** *Let $n \geqslant 3$ and $r \geqslant 2$. Then*

$$\operatorname{dist}\big(0, \operatorname{conv}(\Gamma_{n,6r-3})\big) \leqslant \frac{\sqrt{6}}{(n-1)\sqrt{r}}\, 2^{-r(n-1)+1} \leqslant 2^{-r(n-1)+1}.$$

Proof. We set $N := rn$ and for $i, j, k \in [N]$ we set $\lambda_{i,j,k}$ as in Lemma 2.5 applied for the dimension N. Then Equation (11) of Lemma 2.5 yields

$$\begin{aligned}
&\sum_{i,j,k=1}^{N} \lambda_{i,j,k}\, \big(\varepsilon_{\sigma(i)}, \varepsilon_{\sigma(j)}, \varepsilon_{\sigma(k)}\big) \\
&= \sum_{i,j,k=1}^{N} \lambda_{i,j,k}\, \big(\varepsilon_{\sigma(i)}, 0, 0\big) + \sum_{i,j,k=1}^{N} \lambda_{i,j,k}\, \big(0, \varepsilon_{\sigma(j)}, 0\big) + \sum_{i,j,k=1}^{N} \lambda_{i,j,k}\, \big(0, 0, \varepsilon_{\sigma(k)}\big) \\
&= \sum_{i=1}^{N} \big(\varepsilon_{\sigma(i)}, 0, 0\big) + \sum_{j=1}^{N} \big(0, \varepsilon_{\sigma(j)}, 0\big) + \sum_{k=1}^{N} \big(0, 0, \varepsilon_{\sigma(k)}\big) = \sum_{i=1}^{N} \varepsilon_{\sigma(i),\sigma(i),\sigma(i)} = 0 \in (\mathbb{R}^n)^{6r-3},
\end{aligned}$$

where we used in the last step equation (5) and Remark 2.10, i.e. that each element of $[n]$ is attained exactly r-many times by all $\sigma_k \colon [rn] \to [n]$, $k \in [2r-1]$. Because $\mathfrak{W}_N$ contains the support of λ apart from the element $(1, 1, 1)$, we have

$$\sum_{(i,j,k) \in \mathfrak{W}_N \setminus \mathfrak{J}_r} \lambda_{i,j,k}\, \varepsilon_{\sigma(i),\sigma(j),\sigma(k)} \tag{25}$$

$$= -\lambda_{1,1,1}\, \varepsilon_{\sigma(1),\sigma(1),\sigma(1)} - \sum_{(i,j,k) \in \mathfrak{J}_r} \lambda_{i,j,k}\, \varepsilon_{\sigma(i),\sigma(j),\sigma(k)} =: x \in (\mathbb{R}^n)^{6r-3}, \tag{26}$$

which is an element in the positive cone of $\Gamma_{n,6r-3} = \{\varepsilon_{\sigma(i),\sigma(j),\sigma(k)} \mid (i,j,k) \in \mathfrak{W}_N \setminus \mathfrak{J}_r\}$. Normalizing the latter equation with

$$c := \sum_{(i,j,k) \in \mathfrak{W}_N \setminus \mathfrak{J}_r} \lambda_{i,j,k} = \sum_{i,j,k=1}^{N} \lambda_{i,j,k} - \left(\lambda_{1,1,1} + \sum_{(i,j,k) \in \mathfrak{J}_r} \lambda_{i,j,k} \right) \geqslant N - 1$$

shows $c^{-1}x \in \operatorname{conv}(\Gamma_{n,6r-3})$. To bound the norm of $c^{-1}x$ we compute

$$\begin{aligned}
\lambda_{1,1,1} + \sum_{(i,j,k) \in \mathfrak{J}_r} \lambda_{i,j,k} &= 2^{-N+1} + \sum_{s=2}^{r} \left(\lambda_{s,1,s} + \lambda_{s,s,1}\right) \\
&= 2^{-N+1} + \sum_{s=2}^{r} \big(2^{-N+s-1} + 2^{-N+s-1}\big) = \sum_{s=1}^{r} 2^{-N+s} < 2^{-N+r+1}.
\end{aligned}$$

Finally, using $\|\varepsilon_{i_1,i_2,\ldots,i_{6r-3}}\| \leqslant \sqrt{6r-3}$ for any $i_1, i_2, \ldots, i_{6r-3} \in [n]$ together with the triangle inequality on Equation (26) implies

$$\|c^{-1}x\| \leqslant \frac{\sqrt{6r-3}}{N-1}\, 2^{-N+r+1} \leqslant \frac{\sqrt{6}}{(n-1)\sqrt{r}}\, 2^{-N+r+1} \leqslant 2^{-N+r+1} = 2^{-r(n-1)+1},$$

where we used $n \geqslant 3$ and $r \geqslant 2$ for $\sqrt{6} \leqslant (n-1)\sqrt{r}$. ◀

2.4 Polynomial scaling

A simple example of Equation (2) is the minimization of an n-variate homogeneous polynomial of degree d with nonnegative coefficients over the set $x_1, \ldots, x_n > 0$, $\prod x_i = 1$, as studied in [30]. In this case the sets $\operatorname{conv}(S)$ for $S \subseteq \Omega$ are Newton polytopes of homogeneous polynomials, and the minimum of a polynomial is bounded below if and only if the Newton polytope contains $\frac{d}{n}\mathbb{1}_n$. If the polynomials are hyperbolic of degree n, as in [30], their Newton polytope either contains $\mathbb{1}_n$ or is at least $1/\sqrt{n}$ away from it. However, we show that for general homogeneous polynomials the margin can get exponentially small in n even for $d = 3$.

Minimizing a degree d homogeneous polynomial $\sum_{\alpha \in \mathbb{Z}^n_{\geqslant 0}} p_\alpha x^\alpha$ with nonnegative coefficients over the set $x_1, \ldots, x_n > 0$, $\prod x_i = 1$ is the same as computing Equation (2) for

$$\Omega' := \left\{ -\alpha + \frac{d}{n}\mathbb{1}_n \;\middle|\; \alpha \in (\mathbb{Z}_{\geqslant 0})^n \text{ with } |\alpha| = d \right\}. \tag{27}$$

If $n = dm$ for some integer $m \geqslant 1$, then we have $-\Omega_{m,d} \subseteq \Omega'$. Therefore, Theorem 2.1(b) and (c) and the padding from Appendix C directly yield the following.

▶ **Corollary 2.13** (Margin for Polynomial scaling). *Fix some $d \geqslant 3$ and assume $n = dm$ for some $m \geqslant 3$. Let Ω' be as in Equation (27). Then*

$$\gamma(\Omega') \leqslant \gamma(\Omega_{m,d}) \leqslant 2^{-m+1} = 2^{-\frac{n}{d}+1}.$$

and for $d \geqslant 9$ we even have

$$\gamma(\Omega') \leqslant \gamma(\Omega_{m,d}) \leqslant 2^{-\left\lfloor \frac{(m-1)(d+3)}{6} \right\rfloor + 1} \approx 2^{-\frac{n}{6}}.$$

Thus, for fixed $d \geqslant 3$ and $n \to \infty$ the margin of Ω' can be exponentially small in n. In terms of polynomials, this states that the Newton polytope of a degree $d \geqslant 3$ homogeneous polynomial can be exponentially close to the origin without containing it.

3 Diameter bounds in the commutative case

In this section we describe an array such that all approximate scalings are very ill conditioned, proving Theorem 1.1. Let us define the diameter bound.

▶ **Definition 3.1.** *Let $\varepsilon \to 0$ and $f : \mathbb{R}^m \to \mathbb{R}$. The* diameter bound *$D_f(\varepsilon)$ is defined as the infimum over $R > 0$ such that*

$$\inf_{\|x\| \leqslant R} f(x) \leqslant \varepsilon + \inf_{x \in \mathbb{R}^m} f(x).$$

Thus, Theorem 1.1 is equivalent to the statement that $D_f(\varepsilon) = \Omega(2^{n/3} \log(1/\varepsilon))$ for $\varepsilon \leqslant e^{-Cn^2 \log n}$. We now give a proof outline for Theorem 1.1.

3.1 Proof outline

The high-level intuition applies not only to array scaling but to the capacity in general. Recall that the array scaling capacity is

$$\inf_{x \in \mathbb{R}^{3n}} \sum_{\omega \in \Omega} p_\omega e^{\omega \cdot x}$$

for $\Omega = \Omega_{n,3} = \{e_i - \frac{1}{n}\mathbb{1}_n : i \in [n]\} \subseteq \mathbb{R}^{3n}$. We build both the support $\Omega' \subseteq \Omega_{n,3}$ and the entries p in the following way. We construct a set $\Omega_0 \subseteq \Omega_{n,3}$, another element $\omega \in \Omega_{n,3}$, and an array q with the following properties.

1. The set $\Omega_0 \subseteq \Omega_{n,3}$ should be the support of a tristochastic array q.
2. The affine hull of Ω_0, should have codimension one[13] in $\mathbb{R}^{3n}$.
3. The origin is in the relative interior of $\operatorname{conv}(\Omega_0)$. Note that the origin is already in $\operatorname{conv}(\Omega_0)$ by the tristochasticity of q.
4. The vector $\omega \in \Omega_{n,3}$ should be at a very small, but positive, distance η from $\operatorname{Aff}(\Omega_0)$. Note that this already implies that the facet gap of $\Omega_0 \cup \omega$ is small.

Finally, we define the entries of p by $p|_{\Omega_0} = \frac{1}{2}q$, $p_\omega = \frac{1}{2}$, and $p_\omega = 0$ elsewhere. Assuming we have found p according to this process, we now give intuition for the diameter bound.

Let v be the projection of ω to the orthogonal complement of $\operatorname{Aff}(\Omega_0)$. Intuitively, the capacity is only approximately attained by vectors very far in the $-v$ direction. Indeed, first note that $\operatorname{cap}(p) = 1/2$, because $\operatorname{cap}(q) = 1$ by tristochasticity, $\operatorname{cap}(p) \geqslant \frac{1}{2}\operatorname{cap}(q) = \frac{1}{2}$, and $f_p(-tv/\|v\|) = \frac{1}{2} + e^{-\eta t}$ so $f_p(-tv/\|v\|)$ tends to $\frac{1}{2}$. However, $f_p(-tv/\|v\|)$ tends to $\frac{1}{2}$ slowly if η is small. Indeed, $f_p(-tv/\|v\|) \leqslant \frac{1}{2}(1+\varepsilon)$ only if $t \geqslant \frac{1}{\eta}\log(1/\varepsilon)$.

To conclude rigorously that the capacity is only approached by vectors very far in the $-v$ direction, we must rule out directions with nonzero components in $\operatorname{Aff}(\Omega_0)$. For this, we must use the assumption that 0 is rather deep in the relative interior of $\operatorname{conv}(\Omega_0)$. If this is the case, then any ε-approximate minimizer must have a bounded component in $\operatorname{Aff}(\Omega_0)$, for otherwise the contribution to f_p from the elements of Ω_0 alone will be larger than $\frac{1}{2} + \varepsilon$.

The remainder of the section will be concerned with the construction of a subset Ω_0, an array q, and an element ω with these properties.

3.2 The construction

We construct the subset Ω_0 from a directed graph D on $[n]$, which we will determine later. If i, j is an edge in D, then Ω_0 includes the elements $(\varepsilon_i, \varepsilon_i, \varepsilon_j)$ as well as the three cyclic permutations of it. That is,

$$\Omega_0 = \{(\varepsilon_j, \varepsilon_i, \varepsilon_i), (\varepsilon_i, \varepsilon_j, \varepsilon_i), (\varepsilon_i, \varepsilon_i, \varepsilon_j) : ij \in E(D)\}.$$

We now describe the graph, as seen in Figure 2.

[13] This will not quite apply in our setting, because $\operatorname{Aff}(\Omega_{n,3})$ is not full-dimensional. Instead, $\operatorname{Aff}(\Omega_0)$ will be codimension one in $\operatorname{Aff}(\Omega_{n,3})$.

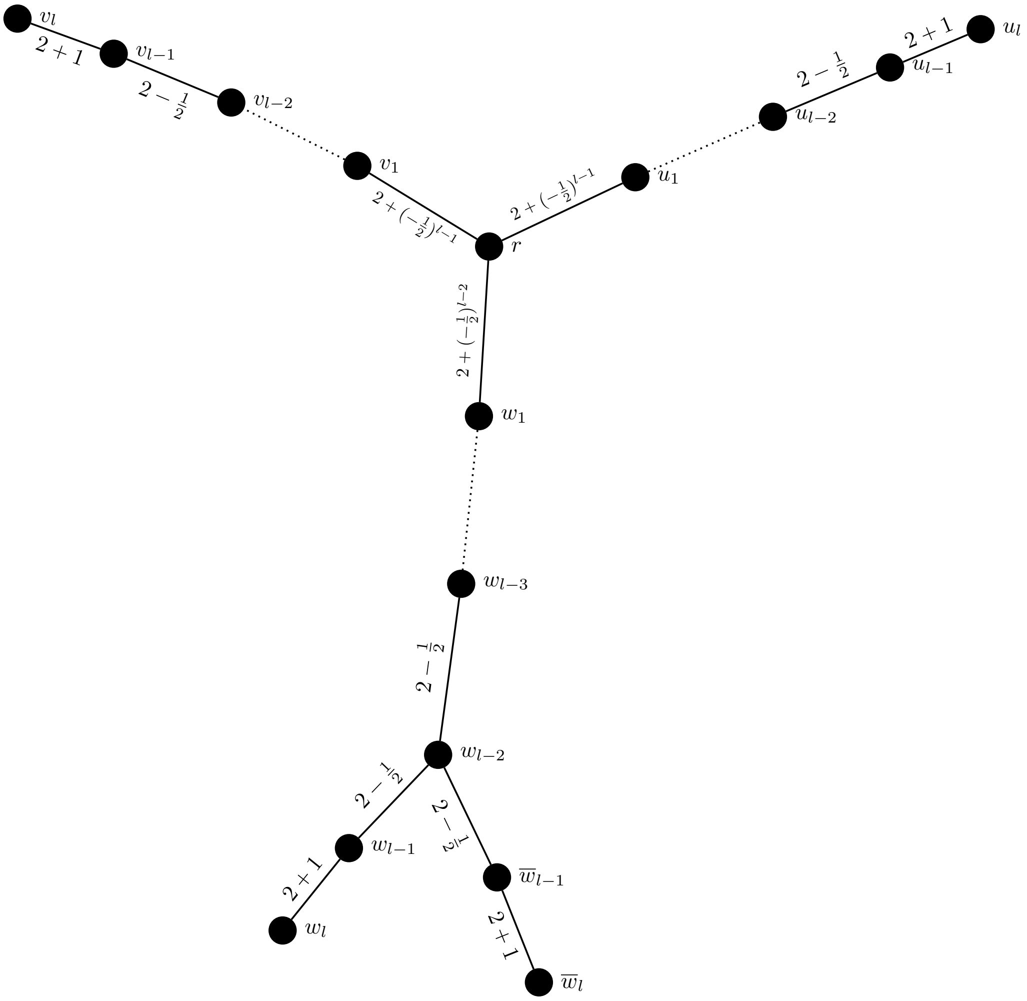


Figure 2 The graph D_l from Definition 3.2 with the edge labels proportional to the edge labeling q in Item 1 of Lemma 3.3 (the constant factor $1/6n$ is omitted for readability). We have also omitted the directions, which are all towards the root r.

▶ **Definition 3.2.** *The graph $D_l = (W, E)$ is a directed tree with $l + 1$ levels, where the root is on the 0^{th} level and the leaves are on the l^{th} level. The tree is constructed as follows.*

- *All the edges are directed towards the root and are between adjacent levels.*
- *The root has three children, and on the $l - 1$ levels below the root every node has one child.*
- *Additionally, one of the vertices on level $l - 2$ has an additional child which has its own child.*

Explicitly, the vertices W and edges E are given by

$$W = \{u_i, v_i, w_i : i \in [l]\} \cup \{w_0 := u_0 := v_0 := r, \bar{w}_{l-1}, \bar{w}_l\}.$$
$$E = \{u_i u_{i-1}, v_i v_{i-1}, w_i w_{i-1} : i \in [l]\} \cup \{\bar{w}_{l-1} w_{l-2}, \bar{w}_l \bar{w}_{l-1}\}.$$

Note that D_l has $3(l + 1)$ vertices so we set $n = 3(l + 1)$. Thus D_l has $3l + 2$ edges and so $|\Omega_0| = 3(3l + 2) = 3n - 3$. It is helpful to construct the matrix M whose set of rows is Ω_0. To make the matrix sparser, first replace ε_i by e_i by restricting the minimization to the

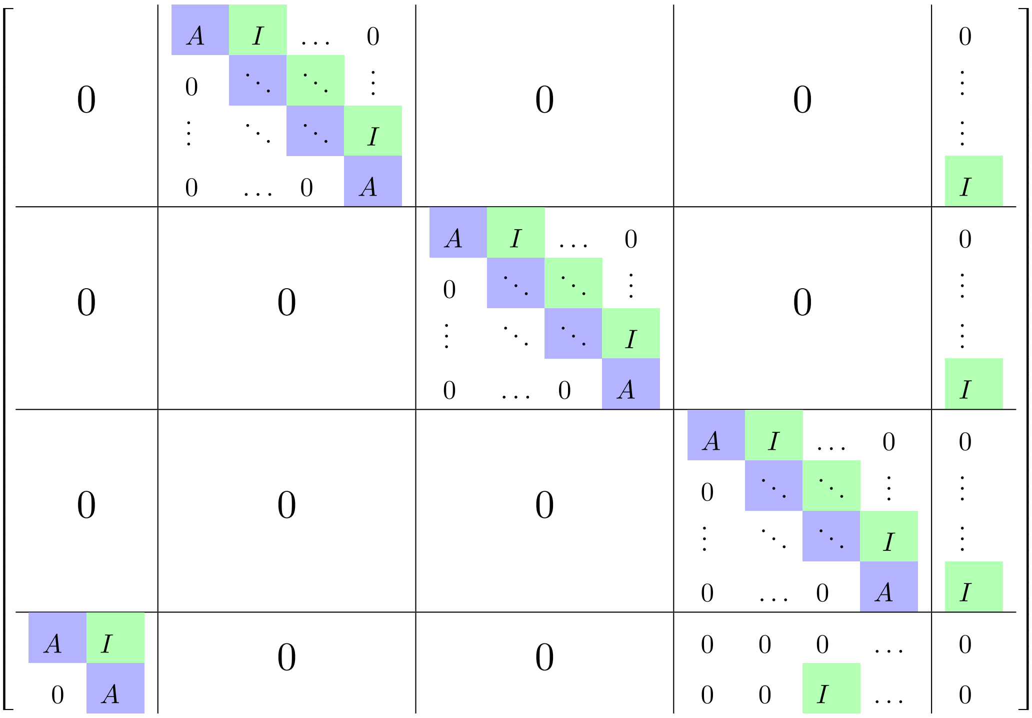


Figure 3 The matrix M written in the reordered basis described before Lemma 3.3. From the left, the five groups of columns correspond to the $\overline{w}'s$, the $u's$, the $v's$, the $w's$, and r among the vertices of D_l. As such the dimensions of the five column groups, from left, are $3 \cdot 2, 3(l-1), 3(l-1), 3(l-1), 3$, and the dimensions of the four groups of rows from top are $3(l-1), 3(l-1), 3(l-1), 3 \cdot 2$. A is as in Equation (28) and I is the 3×3 identity matrix.

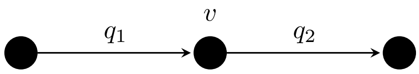


Figure 4 If v is a vertex of D_l with edges weighted q_1 and q_2 incident to it, then the column v, i of M for $i \in [3]$ sums to $q_1 + 2q_2$. That is, the incoming edge contributes its weight and the outgoing edge contributes twice its weight.

subspace $\sum x_i = \sum y_i = \sum z_i = 0$, which is without loss of generality. We define $\Omega_0' \subseteq \mathbb{R}^{3n}$ to be Ω_0 but with each $(\varepsilon_i, \varepsilon_j, \varepsilon_k)$ replaced by (e_i, e_j, e_k); define $\Omega_{n,3}'$ similarly and define $p_{(e_i,e_j,e_k)} := p_{(\varepsilon_i,\varepsilon_j,\varepsilon_k)}$. Then

$$\inf_{x \in \mathbb{R}^{3n}} \sum_{\omega \in \Omega_{n,3}} p_\omega e^{(\varepsilon_i,\varepsilon_j,\varepsilon_k)\cdot x} = \inf_{\substack{x,y,z \in \mathbb{R}^n \\ \sum x_i = \sum y_i = \sum z_i = 0}} \sum_{\omega \in \Omega_{n,3}'} p_\omega e^{(e_i,e_j,e_k)\cdot(x,y,z)}.$$

Moreover, when we write the matrix M, it is easier to write the vector (x, y, z) in the order $(x_1, y_1, z_1, x_2, y_2, z_2, \ldots)$ instead of the order $(x_1, \ldots, x_n, y_1, \ldots, y_n, z_1, \ldots, z_n)$. With this ordering, the matrix M with rows in Ω_0' is a block matrix M with blocks of size 3, with $n-1$ block rows, and with n block columns. Each block row corresponds to an edge in the directed graph $D_l = (W, E)$ on $n = 3(l+1)$ vertices. If $e \in E$ is an edge from $i \to j$, then the e^{th} row of M has the matrix

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \tag{28}$$

in the i^{th} block entry and

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

in the j^{th} block entry and zeroes elsewhere. See Figure 3 for a portrayal of the whole matrix M.

The first three properties for Ω_0 in the proof plan translate to the following three claims about M. The first relates to the tristochasticity of q, the second to the codimension of $\mathrm{Aff}(\Omega_0')$ in the subspace $(\mathbb{1}_n^\perp)^3$, and the third to the depth of the point $\frac{1}{n}\mathbb{1}_{3n}$ in $\mathrm{conv}(\Omega_0')$.

▶ **Lemma 3.3.** *Let $n = 3(l+1)$.*

1. *The probability distribution q on $E \times [3]$ defined (for $i \in [3]$) by*

$$\textit{for } j \in [l], \quad q_{u_j u_{j-1},i} = q_{v_j v_{j-1},i} = \frac{1}{6n}\left(2 + (-2)^{-(l-j)}\right)$$
$$\textit{for } j \in [l-2], \quad q_{w_j w_{j-1},i} = \frac{1}{6n}\left(2 + (-2)^{-(l-j-1)}\right)$$
$$q_{w_{l-1} w_{l-2},i} = q_{\bar{w}_{l-1}\bar{w}_{l-2},i} = \frac{1}{2} q_{w_l w_{l-1},i} = \frac{1}{2} q_{\bar{w}_l \bar{w}_{l-1},i} = \frac{1}{6n}\left(\frac{3}{2}\right)$$

   on the rows of M has expectation $\frac{1}{n}\mathbb{1}_{3n}$. That is, if the rows of M are scaled by the values of q, each column sums to $1/n$. Note that the entries of q are $\Theta(\frac{1}{n})$. Ignoring the index i in $q_{uv,i}$ allows us to view q as a labeling of the edges of the graph D_l; see Figures 2 and 4.

2. $\ker M = \mathrm{span}(\Omega_0')^\perp$ *is spanned by the 2 dimensional space $S \subseteq \mathbb{R}^{W \times [3]}$ given by*

$$S = \{s : s(v,1) = \alpha, s(v,2) = \beta, s(v,3) = \gamma \textit{ for all } v \in W,\ \alpha + \beta + \gamma = 0\}$$

   and the function $f \in \mathbb{R}^{W \times [3]}$ which for all $i \in [3]$ assigns

$$f(u_j, i) = f(v_j, i) = f(w_j, i) = (-2)^{-j} \textit{ for } j \in [l] \cup \{0\}$$
$$\textit{and } f(\bar{w}_{l-k}, i) = f(w_{l-k}, i) \textit{ for } k \in \{0,1\}. \tag{29}$$

   Note that $f \in (\mathbb{1}_n^\perp)^3 \subseteq S^\perp$. Thus we have the orthogonal decomposition $\mathrm{span}(\Omega_0')^\perp = S \oplus \mathrm{span}\, f$.

3. *Apart from the three zero singular values, all singular values of M are $\Omega(1/n)$.*

Given the lemma, let us prove that the diameter bound holds according to the proof outline at the beginning of the section.

Proof of Theorem 1.1. We first show the claim for n of the form $n = 3(l+1)$; the bound follows for $3(l+1) < n < 3(l+2)$ by applying Proposition 3.5 with $t = 3(l+1)$, using that the array we construct has capacity 1/2 and $t/n \geqslant 2/3$.

We now show the diameter lower bound for $n = 3(l+1)$. It is enough to exhibit a constant $C > 0$, and a probability distribution p on $\Omega_{n,3}' = \{e_i : i \in [n]\}^3$ such that for for all $N \geqslant Cn^2 \log n$ and all $x, y, z \in \mathbb{1}_n^\perp$,

$$\sum_{\omega \in \Omega_{n,d}'} p_\omega e^{\omega \cdot (x,y,z)} \leqslant e^{-N} + \inf_{x',y',z' \in \mathbb{1}_n^\perp} \sum_{\omega \in \Omega_{n,d}'} p_\omega e^{\omega \cdot (x,y,z)}$$

only if $\|(x, y, z)\|_2 = \Omega(2^{n/3}N)$. Note that the space $(\mathbb{1}_n^{\perp})^3$ over which we are infimizing is a subspace of $S^{\perp}$ where S is as in Lemma 3.3, and that $\Omega'_{n,d} \subseteq S^{\perp}$. The proof will follow the outline in Section 3.1; namely, we will consider a subset $\Omega'_0 \subseteq \Omega'_{n,d}$ and an element $\omega' \in \Omega'_{n,d}$ very close to, but outside of, $\mathrm{Aff}(\Omega'_0)$.

Consider the set $\Omega'_0 \subseteq \Omega'_{n,d}$ of rows of M in Lemma 3.3 and the probability distribution q on Ω'_0 from Lemma 3.3. Let $\omega' = (e_{u_l}, e_{v_l}, e_{w_l})$ for the vertices $u_l, v_l, w_l \in D_l$. Let $\Omega = \Omega'_0 \cup \{\omega'\}$, and define the probability distribution p on Ω by $p_{\omega'} = \frac{1}{2}$ and $p_\omega = \frac{1}{2}q_\omega$ for $\omega \in \Omega'_0$. Recall from Lemma 3.3 the orthogonal decomposition $\mathrm{span}(\Omega'_0)^{\perp} = S \oplus \mathrm{span}\, f$. As $\mathbb{R}^{3n} = \mathrm{span}(\Omega'_0) \oplus \mathrm{span}(\Omega'_0)^{\perp}$, we have the orthogonal decomposition $S^{\perp} = \mathrm{span}(\Omega'_0) \oplus \mathrm{span}\, f$. Observe that $\omega' \notin \mathrm{span}(\Omega'_0)$, because by Lemma 3.3 we have $\mathrm{span}(\Omega'_0)^{\perp} = \ker M = S + \mathrm{span}\, f$ and clearly $f \cdot \omega' \neq 0$.

By Item 1 of Lemma 3.3 we have $\sum_{\omega\in\Omega'_0} q_\omega \omega = \frac{1}{n}(\mathbb{1}_n, \mathbb{1}_n, \mathbb{1}_n)$ and thus $\mathrm{cap}(q) = 1$. Therefore, $\omega' \notin \mathrm{span}(\Omega'_0)$ implies that the infimum is $1/2$ for this choice of Ω and p. We claim that the infimum can only be approximately attained by $h \in (\mathbb{1}_n^{\perp})^3$ with a very large component in the one-dimensional space $\mathrm{span}\, f = \mathrm{span}(\Omega'_0)^{\perp} \cap (\mathbb{1}_n^{\perp})^3$. As in the proof outline, we must bound the components in $\mathrm{span}(\Omega'_0)$ of the approximate minimizer h. For $h \in (\mathbb{1}_n^{\perp})^3$ write $h = h_0 + af$ and $\omega' = \omega_0 + bf$ where $h_0, \omega_0 \in \mathrm{span}\, \Omega'_0$. Note that $|b| = \frac{|f\cdot\omega'|}{\|f\|^2} = O(2^{-l}) = O(2^{-n/3})$ and that $h_0 \in (\mathbb{1}_n^{\perp})^3$, because h and f are. Suppose

$$\sum_{\omega\in\Omega} p_\omega e^{\omega\cdot h} \leqslant \frac{1}{2}e^{-N} + \frac{1}{2}.$$

Equivalently,

$$\sum_{\omega\in\Omega'_0} q_\omega e^{\omega\cdot h_0} + e^{h_0\cdot\omega_0 + ab\|f\|^2} \leqslant e^{-N} + 1. \tag{30}$$

Suppose $\|h_0\|$ is bounded by L. If $e^{h_0\cdot\omega_0+ab\|f\|^2} \leqslant e^{-N}$, then $|ab| = \Omega(N - L)$. In particular, $\|h\| \geqslant \|af\| = |ab|\|f\|/|b| = \Omega((N-L)2^{n/3})$ because of the previous bounds on $|ab|, |b|$, and the fact that $\|f\| = \Theta(1)$. It remains to prove a bound L for $\|h_0\|$. We will do this by showing that if $\|h_0\|$ were too large, then the first term of the left-hand side of Equation (30) would be too large. This amounts to $\frac{1}{n}\mathbb{1}_{3n}$ being in the relative interior of $\mathrm{conv}(\Omega'_0)$, but will be proved using lower bounds on the singular values of M.

Let α denote the least nonzero singular value of M; by Item 3 of Lemma 3.3 $\alpha = \Omega(1/n)$. As $h_0 \in \mathrm{span}(\Omega'_0) = \mathrm{rowspan}(M)$, we have $\|Mh_0\| \geqslant \alpha\|h_0\|$ by the singular value bound. We claim that there is some $\omega \in \Omega'_0$ satisfying $\omega \cdot h_0 = \Omega(\alpha\|h_0\|/n)$. To prove this, first note that the $\sum_{\omega\in\Omega'_0} q_\omega \omega \cdot h_0 = \frac{1}{n}(\mathbb{1}_n, \mathbb{1}_n, \mathbb{1}_n) \cdot h_0 = 0$ because $h_0 \in (\mathbb{1}_n^{\perp})^3$. Moreover, by Lemma 3.3 we have $q_\omega = \Theta(1/n)$. The claim follows from Lemma 3.4 below applied to the sequence $(\omega \cdot h_0 : \omega \in \Omega'_0)$.

Because $q_\omega = \Theta(\frac{1}{n})$, we must have that $\omega \cdot h_0 = O(\log n)$ for all $w \in \Omega'_0$. Else, the contribution from the term $q_\omega e^{\omega\cdot h_0}$ alone is larger than 1, in which case x cannot be an e^{-N}-approximate minimizer. Finally, $\|h_0\| = O(n(\log n)/\alpha) = O(n^2 \log n)$, and so we may take $L = O(n^2 \log n)$ and $N \geqslant 2L$. ◀

In the above proof, we used the following simple lemma.

▶ **Lemma 3.4.** *Let $0 < \beta < \gamma$. Suppose $z \in \mathbb{R}^m$ is such that $\sum_{i=1}^m q_i z_i = 0$ for $q_i \in (\beta/m, \gamma/m)$. Then there exists $i \in [m]$ such that $z_i \geqslant \frac{\beta}{2\gamma m}\|z\|_2$.*

Proof. Because $\sum q_i z_i = 0$,

$$\sum_{i:z_i<0} q_i|z_i| = \sum_{i:z_i\geqslant 0} q_i z_i,$$

and

$$\sum_{i:z_i<0} q_i|z_i| + \sum_{i:z_i\geqslant 0} q_i z_i \geqslant (\beta/m)\|z\|_1 \geqslant (\beta/m)\|z\|_2.$$

Thus $\sum_{i:z_i\geqslant 0} q_i|z_i| \geqslant \frac{\beta}{2m}\|z\|_2$, so there is some i such that $q_i z_i > \frac{1}{m}\frac{\beta}{2m}\|z\|_2$. Thus $z_i > \frac{\beta}{2\gamma m}\|z\|_2$. ◀

To show that our diameter lower bound holds for all values of n, we need the following proposition, which is proved in Appendix E. The idea is to prove diameter bounds for larger arrays from diameter bounds for smaller ones by embedding the smaller array in a "corner" of the larger array.

▶ **Proposition 3.5.** *Suppose $1 \leqslant t \leqslant n$. Let p be a d-dimensional array in $(\mathbb{R}^t_{\geqslant 0})^{\otimes d}$ with unit sum; in particular $\operatorname{cap}(p) \leqslant 1$. Let q be the d-dimensional array in $(\mathbb{R}^n_{\geqslant 0})^{\otimes d}$ array such that $q_{i_1,\dots,i_d} = \frac{t}{n} p_{i_1,\dots,i_d}$ for $i_1,\dots,i_d \in [t]$, $q_{iii} = 1/n$ for $t+1 \leqslant i \leqslant n$, and $q_{i_1,\dots,i_d} = 0$ otherwise. For $\varepsilon \leqslant 1 - \operatorname{cap}(p)$,*

$$D_{f_q}(\varepsilon) \geqslant D_{f_p}\left(\frac{(1-\operatorname{cap}(p))\varepsilon}{1-\operatorname{cap}(p)^{t/n}}\right).$$

In particular, the norm of any ε-approximate minimizer of f_q is at least the norm of some $\left(\frac{1-\operatorname{cap}(p)}{1-\operatorname{cap}(p)^{t/n}}\right)\varepsilon$-approximate minimizer of f_p.

As a corollary of the proof of Theorem 1.1, we have a bound on the *facet gap* of [14]. The facet gap of a finite set Ω is defined to be the least distance of an element of Ω to the affine hull of a facet of $\operatorname{conv}(\Omega)$. We have shown that the distance between $\operatorname{Aff}(\Omega_0')$ and ω' is $O(2^{-l})$, or $O(2^{-n/3})$.

▶ **Corollary 3.6** (Facet gap of array scaling). *There is a subset $\Omega_1 \subseteq \Omega_{n,3}$ with facet gap $O(2^{-n/3})$.*

Analogously to what is done for the margin in Proposition C.1, we may also embed this array inside a larger array to obtain a diameter bound for $d \geqslant 3$. For $d \geqslant 3$, take $q(i,j,k,l,l,\dots,l) = \frac{1}{n} p_{ijk}$ for all $i,j,k,l \in [n]$. Then for $(x_1,\dots,x_d) \in (\mathbb{1}^{\perp}_n)^d$ we have

$$f_q(x_1,\dots,x_d) = \frac{1}{n} f_p(x_1,x_2,x_3) \sum_{l=1}^n e^{\sum_{j=4}^d (x_j)_l}.$$

For fixed x_1,x_2,x_3, by Jensen's inequality f_q is minimized when $x_j = 0_n$ for $j \geqslant 4$ and takes value $f_p(x_1,x_2,x_3)$, and thus f_q has the same diameter bound as f_p.

▶ **Corollary 3.7** (Diameter bound for $d \geqslant 3$). *There is an absolute constant $C > 0$ such that the following holds. For all $d \geqslant 3$, there is a family of arrays $q \in (\mathbb{R}^n_{\geqslant 0})^{\otimes d}$ with $O(n^2)$ nonzero entries, each of bit-complexity $O(n)$, that satisfies the following property. For all $0 < \varepsilon \leqslant \exp(-Cn^2 \log n)$ and $x \in \mathbb{R}^{dn}$, if*

$$f_q(x) \leqslant \operatorname{cap}(p) + \varepsilon$$

then $\|x\|_2 = \Omega\left(2^{n/3}\log(1/\varepsilon)\right)$.

3.3 Proof of the properties of the construction

We now prove Lemma 3.3.

Proof of Lemma 3.3. It is first helpful to change basis on each copy of $\mathbb{R}^3$ so that the A blocks are diagonalized. Let $U \in \mathrm{Mat}(3)$ be an orthogonal matrix such that

$$U^\dagger A U = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

This is possible because $2, -1, -1$ are the eigenvalues of the symmetric matrix A. In particular, the first column of U is $(1,1,1)/\sqrt{3}$, and the second two columns span the space of vectors with sum zero. Then $M' = (U^{\oplus n})^\dagger M U^{\oplus n}$ is of the form $P \oplus L \oplus L$ where $P_{e,v} = M'_{(e,1),(v,1)}$ for $e, v \in E \times V$ and $L_{e,v} = M'_{(e,2),(v,2)}$. Note that L is the edge-vertex incidence matrix of the directed graph D_l, the row corresponding to the edge (u,v) of D_l has a -1 in the column indexed by the vertex u and a $+1$ in the column indexed by v. Moreover, P is the matrix obtained from L by replacing every -1 entry by a 2.

To prove Item 2, observe that $\ker M$ is $(U^{\oplus n}) \ker M' = (U^{\oplus n}) \ker P \oplus \ker L \oplus \ker L$. Because D_l is connected, $\ker L = \operatorname{span} \mathbb{1}_n$. As the second two columns of U span the subspace of $\mathbb{R}^3$ of vectors with sum 0, the two-dimensional space S is given by $(U^{\oplus n})0 \oplus \operatorname{span} \mathbb{1}_n \oplus \operatorname{span} \mathbb{1}_n = (U^{\oplus n})0 \oplus \ker L \oplus \ker L$. We next reason for $\ker P$, the other summand of the orthogonal decomposition of $\ker M'$. The graph D_l is a connected tree, so $\ker P$ is one dimensional. This is because every choice of $g(w_0) \in \mathbb{R}$ determines a unique function $g : V \to \mathbb{R}$ in $\ker P$. We claim that the function $g(v) = f(v,1)$ for f as in Equation (29) is in $\ker P$, and hence spans it. To check this, one must check that for every edge $(v,w) \in E$ we have $2g(v) + g(w) = 0$. It is instructive to look at Figure 2. Observe that this property holds for the edges $u_{k,k-1}$ if the sequence $g(u_k)$ obeys the recurrence relation $g(u_{k-1}) = -2g(u_k)$ for $k \in [l]$, which is indeed true by the definition of f. Checking the condition for v and w is similar. As the first column of U is proportional to $\mathbb{1}_3$, $(U^{\oplus n}) \ker P \oplus 0 \oplus 0$ is spanned by the function f. This proves Item 2.

To show Item 3, it is enough to argue that the singular values of P, L, L obey the desired bound. For L this follows straightforwardly from the fact that L is an incidence matrix of a connected, directed tree and so is totally unimodular with linearly independent rows. The singular value bound follows by Lemma 3.8. Rather than arguing spectrally for P, we make an ad-hoc argument using the structure of D_l. We first show that $\|x^t P\|_\infty = \Omega(\|x\|_\infty)$ for all $x \in \mathbb{R}^{n-1}$, which suffices because $\|x^t P\|_2 \geqslant \|x^t P\|_\infty$ and $\|x\|_\infty \geqslant \frac{1}{\sqrt{n}} \|x\|_2$.

Let $x \in \mathbb{R}^{n-1} \backslash \{0\}$ and e be an edge in D_l such that $|x(e)| = \|x\|_\infty$. If $e = u_i u_{i-1}$ for $i \in [l]$, then $|x^t P(u_i)| \geqslant \|x\|_\infty$ because either $i < l$ in which case

$$|x^t P(u_i)| = |2x(u_i u_{i-1}) + x(u_{i+1} u_i)| \geqslant 2\|x\|_\infty - |x(u_{i+1} u_i)| \geqslant \|x\|_\infty$$

or $i = l$ and so $|x^t P(u_i)| = |2x(e)| = 2\|x\|_\infty$. The same argument applies to all other edges except $e = w_{l-2} w_{l-3}$. In the latter case we are done if $x^t P(w_{l-2}) \geqslant 1/3\|x\|_\infty$. Otherwise we necessarily have $|x(w_{l-1} w_{l-2})| + |x(\bar{w}_{l-1} w_{l-2})| \geqslant 5/3\|x\|_\infty$, since $x^t P(w_{l-2}) = 2x(e) + x(w_{l-1} w_{l-2}) + x(\bar{w}_{l-1} w_{l-2})$. It follows that $|x(w_{l-1} w_{l-2})| \geqslant 5/3\|x\|_\infty - |x(\bar{w}_{l-1} w_{l-2})| \geqslant 5/3\|x\|_\infty - \|x\|_\infty \geqslant 2/3\|x\|_\infty$. As $|x^t P(w_{l-1})| = 2x(w_{l-1} w_{l-2}) + x(w_l w_{l-1})$, we have

$$|x^t P(w_{l-1})| \geqslant 2|x(w_{l-1} w_{l-2})| - |x(w_l w_{l-1})| \geqslant \frac{4}{3}\|x\|_\infty - \|x\|_\infty \geqslant \frac{1}{3}\|x\|_\infty.$$

In any case, there is some value of $x^t P$ with absolute value greater or equal $1/3\|x\|_\infty$.

Finally, for Item 1 we note that the probability distribution q on the rows of M has expectation equal to the all $1/n$ function if and only if the probability distribution q' defined by $q'_e = 3q_{e,1}$ on the rows of P has expectation equal to the all $3/n$ function on the vertices of D_l. Recall that P is obtained from the edge-vertex incidence matrix of D_l by replacing every -1 with a 2. Thus the expectation of the rows under q' at a vertex v is $\sum_{w:(w,v)\in D_l} q'_{(w,v)} + \sum_{w:(v,w)\in D_l} 2q'_{(v,w)}$; see Figure 4. We now check that this is equal to $3/n$ for each vertex of D_l; it is helpful to look at Figure 2. The leaves u_l, v_l, w_l, and $\overline{w}_l$ all have outdegree one and indegree zero, and q' takes the value $3 \cdot 3/6n = 3/2n$ on the outgoing edges. The expectation under q' thus takes value $3/n$ on these vertices. On vertices of indegree one and outdegree one, q' takes the value $\frac{1}{2n}\left(2 + (-2)^{-k}\right)$ on the incoming edge and $\frac{1}{2n}\left(2 + (-2)^{-(k+1)}\right)$ on the outgoing edge. Thus the expectation takes the value $\frac{1}{2n}\left(2 + (-2)^{-k}\right) + \frac{1}{2n}\left(4 - (-2)^{-k}\right) = 3/n$. The remaining vertices to check, those of total degree three, are r and w_{l-2}. For r, which has only incoming edges, the expectation under q' is $2 \cdot \frac{1}{2n}\left(2 + (-2)^{-(l-1)}\right) + \frac{1}{2n}\left(2 + (-2)^{-(l-2)}\right)$, which is again $3/n$. For w the expectation is $2 \cdot \frac{1}{2n}\left(2 - \frac{1}{2}\right) + 2 \cdot \frac{1}{2n}\left(2 - \frac{1}{2}\right) = 3/n$. This completes the proof. ◀

▶ **Lemma 3.8.** *If A is an $n \times k$ totally unimodular matrix with linearly independent columns, then the eigenvalues of $A^T A$ are all at least $1/n^2$.*

Proof. First note that $k \leqslant n$ by the linear independence of the columns of A. The least eigenvalue of $A^T A$ is $\min_{x\in\mathbb{R}^k\setminus\{0\}}(x^T A^T Ax)/\|x\|^2 = \min_{x\in\mathbb{R}^k\setminus\{0\}} \|Ax\|^2/\|x\|^2$, so it suffices to show that for all $x \in \mathbb{R}^k$, Ax has norm at least $\|x\|/n$. Indeed, if $Ax = y$, then there is some invertible $k \times k$ submatrix A' of A and $k \times 1$ submatrix y' of y such that $A'x = y'$. By Cramer's rule and unimodularity of A' we have that, for $i \in [k]$,

$$x_i = \frac{\det(B_i)}{\det(A')} = \pm \det(B_i)$$

where B_i is simply the matrix that one obtains by replacing the i^{th} column of A' with the vector y'. By performing the Laplace expansion with respect to the i^{th} column, and by unimodularity of the minors, we have that $x_i \leqslant \|y\|_1$, and so $\|x\|_2 \leqslant \sqrt{k}\|y\|_1 \leqslant n\|Ax\|_2$ (using $k \leqslant n$). ◀

4 The noncommutative case

In this section we extend the results from the commutative to the noncommutative case. For this, we recall in the first subsection necessary concepts such as moment maps and moment polytopes, and we define the weight margin and the gap of a representation. The second subsection introduces the key concept of a *free* subset of weights, see [24]. This concept dates at least back to [18, Proposition 1.2], where it is called *strong orthogonality*. Freeness will be used to transfer results from the commutative to the noncommutative case.[14] The latter is done in the following three subsections, where we prove bounds on the tensor gap, on the gap for homogeneous polynomials and on the diameter for the natural $\mathrm{SL}(n)^3$ action on 3-tensors. Finally, we show a bound for the weight margin of certain quiver representations. This provides an example, where the constructed set of weights is *not* free, compare Remark 4.28. Still, after adding enough arrows to the considered quiver, we are able to ensure the same bound for the gap.

[14] Actually all presented concepts in the first two subsections work in the very general setting of reductive groups and their rational representations. For the sake of clarity and concreteness we stick to the special case needed in this paper, i.e. the reductive group $\mathrm{SL}(n)^d := \mathrm{SL}(n) \times \cdots \times \mathrm{SL}(n)$ with $d \geqslant 1$ many copies of $\mathrm{SL}(n)$.

4.1 Moment maps and moment polytopes

In the following we introduce the null-cone problem and its dual characterization via moment maps and moment polytopes. This allows us to rigorously introduce the weight margin and the gap of a rational representation. Thereby we establish precise meaning and interpretation of our results regarding these two notions (in view of the null-cone problem). We stick to the notation of [12], where the gap (implicitly) and the weight margin have been introduced. A reader unfamiliar with representation theory is referred to Appendix B.

Let $G = \mathrm{SL}(n)^d$, $K = \mathrm{SU}(n)^d$, $\mathrm{T} = \mathrm{ST}(n)^d$ and $\mathrm{T}_K = K \cap \mathrm{T}$ be matrix Lie subgroups of $\mathrm{GL}(dn)$ via block-diagonal embedding. Then we can think of their Lie algebras $\mathrm{Lie}(G)$ etc. as being block diagonally embedded into $\mathbb{C}^{dn \times dn}$. For a rational representation $\pi \colon G \to \mathrm{GL}(V)$ we write $g \cdot v := \pi(g)v$ for the induced action, where $g \in G$ and $v \in V$. Moreover, we denote the set of weights of π by $\Omega(\pi) \subseteq i\,\mathrm{Lie}(\mathrm{T}_K)$ and the induced representation on Lie algebras by $\Pi \colon \mathrm{Lie}(G) \to \mathrm{End}(V)$. We remark that we usually identify $i\,\mathrm{Lie}(\mathrm{T}_K) \cong (\mathbb{1}_n^{\perp})^d \subseteq (\mathbb{R}^n)^d$, where $\mathbb{1}_n^{\perp}$ denotes the orthogonal complement of the all-ones vector $\mathbb{1}_n$ in $\mathbb{R}^n$.

The *orbit* of $v \in V$ is $G \cdot v := \{g \cdot v \mid g \in G\}$ and we denote its closure[15] by $\overline{G \cdot v}$. A vector v is called *G-unstable*, if $0 \in \overline{G \cdot v}$, and otherwise v is G-semistable. Equivalently, a vector $v \in V$ is G-unstable if and only if its *capacity*

$$\mathrm{cap}_G(v) := \inf_{g \in G} \|g \cdot v\|^2$$

equals zero. The G-unstable vectors form an affine subvariety of V - the *null-cone* (with respect to G). Orbit, stability, and capacity can also be defined for T by replacing G by T in the definitions.

As discussed in Section 1.2, the null-cone problem has many applications in different fields of computer science, mathematics and physics.

Next, we introduce the moment map. Given a rational representation $\pi \colon G \to \mathrm{GL}(V)$ there exists an Hermitian inner product $\langle \cdot, \cdot \rangle$ on V, by convention linear in the second argument, such that $\langle k \cdot v, k \cdot w \rangle = \langle v, w \rangle$ holds for all $k \in K$ and all $v, w \in V$.[16]

▶ **Definition 4.1.** *For $v \in V \backslash \{0\}$ we define $\mu_G(v) \in i\,\mathrm{Lie}(K)$ as the unique element of the real vector space $i\,\mathrm{Lie}(K)$, which satisfies for all $A \in i\,\mathrm{Lie}(K)$*

$$\mathrm{tr}\left(\mu_G(v)A\right) = \frac{\langle v, \Pi(A)v \rangle}{\langle v, v \rangle}.$$

This defines the moment map $\mu_G \colon V \backslash \{0\} \to i\,\mathrm{Lie}(\mathrm{T})$ *of G. Replacing G by T and K by T_K we derive the moment map $\mu_{\mathrm{T}} \colon V \backslash \{0\} \to i\,\mathrm{Lie}(\mathrm{T}_K)$ of T.*

The maps μ_G and μ_{T} are indeed moment maps in the sense of symplectic geometry; namely for the induced action of K and, respectively, T_K on the projective space $\mathbb{P}(V)$. Recall $i\,\mathrm{Lie}(K) \subseteq \mathbb{C}^{dn \times dn}$ so we can consider $\|\mu_G(v)\|_F$ and $\|\mu_{\mathrm{T}}(v)\|_F$.

An important application of these moment maps is due to the Kempf-Ness theorem [36], which provides a duality for the null-cone membership problem:

$$\mathrm{cap}_G(v) = 0 \qquad \Leftrightarrow \qquad 0 < \inf_{g \in G} \|\mu_G(g \cdot v)\|_F = \min_{0 \neq w \in \overline{G \cdot v}} \|\mu_G(w)\|_F \tag{31}$$

and similarly for T, replacing G by T in the above equation. The two moment maps are related as follows.

[15] The Euclidean- and the Zariski-closure of $G \cdot v$ coincide.

[16] In our concrete representations later on this will be the standard inner product.

▶ **Proposition 4.2.** *Let $p\colon i\,\mathrm{Lie}(K) \to i\,\mathrm{Lie}(\mathrm{T}_K)$ be the orthogonal projection. Then $\mu_{\mathrm{T}} = p \circ \mu_G$ and $\|\mu_{\mathrm{T}}(v)\|_F \leqslant \|\mu_G(v)\|_F$ for all $v \in V\backslash\{0\}$.*

Proof. Since $i\,\mathrm{Lie}(\mathrm{T}_K) \subseteq i\,\mathrm{Lie}(K)$ the definition of the moment maps gives $\mathrm{tr}[\mu_{\mathrm{T}}(v)H] = \mathrm{tr}[\mu_G(v)H]$ for all $H \in i\,\mathrm{Lie}(\mathrm{T}_K)$. But $\mu_{\mathrm{T}}(v) \in i\,\mathrm{Lie}(\mathrm{T}_K)$ is the unique element with this property, hence $p(\mu_G(v)) = \mu_{\mathrm{T}}(v)$. The inequality $\|\mu_{\mathrm{T}}(v)\|_F \leqslant \|\mu_G(v)\|_F$ follows directly from the first part. ◀

Now, we explain how the moment maps induce certain polytopes, which can also be used to express the duality in (31). Moreover, the combinatorics of these polytopes captures the important complexity measures *(weight) margin* and *gap*. Indeed, one of our main contributions is to analyze parts of this combinatorics, thereby deducing complexity barriers for certain computational problems.

Since the action of T via π is completely determined by the weight space decomposition $V = \bigoplus_{\omega\in\Omega(\pi)} V_\omega$ of V, one can compute $\mu_{\mathrm{T}}(v)$ in terms of this decomposition. For this, write $v = \sum_\omega v_\omega$ with $v_\omega \in V_\omega$ and define the support of v with respect to π as

$$\mathrm{supp}(v) := \{\omega \in \Omega(\pi) \mid v_\omega \neq 0\}.$$

Using that distinct weight spaces are orthogonal, one computes

$$\mu_{\mathrm{T}}(v) = \sum_\omega \frac{\langle v_\omega, v_\omega\rangle}{\langle v, v\rangle}\,\omega,$$

which is a convex combination of the weights in $\mathrm{supp}(v)$. Noting that $\mathrm{supp}(v) = \mathrm{supp}(t \cdot v)$ for $t \in \mathrm{T}$ also $\mu_{\mathrm{T}}(t\cdot v) \in \Delta_{\mathrm{T}}(v) := \mathrm{conv}\{\omega \mid \omega \in \mathrm{supp}(v)\}$. In fact,

$$\Delta_{\mathrm{T}}(v) = \overline{\{\mu_{\mathrm{T}}(t\cdot v) \mid t \in \mathrm{T}\}} = \big\{\mu_{\mathrm{T}}(w) \mid w \in \overline{\mathrm{T}\cdot v}, w \neq 0\big\} \subseteq i\,\mathrm{Lie}(\mathrm{T}_K)$$

and $\Delta_{\mathrm{T}}(v)$ is called the *weight polytope* of v.

It is an astonishing result that for fixed $v \in V\backslash\{0\}$, the set $\{\mu_G(g\cdot v) : g \in G\}$ gives rise to a polytope as follows. Let $\mathrm{spec}\colon \mathrm{Herm}(n) \to \mathbb{R}^n$ be the function sending a Hermitian matrix to its eigenvalues in decreasing order. Recalling that $i\,\mathrm{Lie}(K) \subseteq \mathrm{Herm}(n)^d$ is block-diagonally embedded in $\mathbb{C}^{dn\times dn}$, we set

$$s\colon i\,\mathrm{Lie}(K) \to (\mathbb{R}^n)^d, \quad \mathrm{diag}(A_1,\dots,A_d) \mapsto \big(\mathrm{spec}(A_1),\dots,\mathrm{spec}(A_d)\big).$$

Then for $v \in V\backslash\{0\}$ the set

$$\Delta_G(v) := \big\{s\big(\mu_G(w)\big) \mid w \in \overline{G\cdot v}, w \neq 0\big\}$$

is a rational convex polytope, see e.g. [28] or [45, Appendix] by Mumford. We call $\Delta_G(v)$ the *moment polytope* of v. Noting that $\|A\|_F = \|\mathrm{spec}(A)\|_2$ for any $A \in \mathrm{Herm}(n)$ we have $\|\mu_G(v)\|_F = \|s(\mu_G(v))\|_2$ for all $v \in V\backslash\{0\}$. Thus, we can formulate the duality from (31) also as follows:

$$\mathrm{cap}_G(v) = 0 \quad\Leftrightarrow\quad \mathrm{dist}\big(0, \Delta_G(v)\big) > 0 \quad\Leftrightarrow\quad 0 \notin \Delta_G(v),$$

and similarly for T. This motivates the following two definitions.

▶ **Definition 4.3.** *Let $\pi\colon G \to \mathrm{GL}(V)$ be a rational representation. We define the* gap *of π as*[17]

$$\gamma_G(\pi) := \min\left\{\|\mu_G(v)\|_F \mid v \neq 0 \textit{ is } G\textit{-unstable}\right\} = \min\left\{\operatorname{dist}\left(0, \Delta_G(v)\right) \mid v \neq 0 \textit{ is } G\textit{-unstable}\right\},$$

and the weight margin *of π as*

$$\gamma_{\mathrm{T}}(\pi) := \min\left\{\|\mu_{\mathrm{T}}(v)\|_F \mid v \neq 0 \textit{ is } \mathrm{T}\textit{-unstable}\right\} = \min\left\{\operatorname{dist}\left(0, \Delta_{\mathrm{T}}(v)\right) \mid v \neq 0 \textit{ is } \mathrm{T}\textit{-unstable}\right\}.$$

Equivalently, $\gamma_{\mathrm{T}}(\pi)$ is the margin of the set of weights $\Omega(\pi)$, i.e. $\gamma_{\mathrm{T}}(\pi) = \gamma(\Omega(\pi))$.

Thus, the gap $\gamma_G(\pi)$ is the largest constant $C > 0$ with the following property: If $\|\mu_G(v)\|_F < C$ for some vector $v \in V$, then v is G-semistable. The same statement holds for the weight margin $\gamma_{\mathrm{T}}(\pi)$ replacing G by T. Therefore, these notions capture how small $\mu_G(g \cdot v)$ (respectively $\mu_{\mathrm{T}}(t \cdot v)$) must be to certify null-cone non-membership. The next remark connects the gap to the classical notion of *instability* due to Mumford [44].

▶ **Remark 4.4.** *The gap is twice the minimum value of all positive instabilities. Indeed, let $M(v)$ denote the instability of a non-zero vector v, see e.g. [45, eq. (9)]. Then* $\operatorname{dist}(0, \Delta_G(v)) \geqslant 2M(v)$ *and [45, Theorem 6.1] implies*

$$\gamma_G(\pi) = \inf\{2M(v)\colon v \neq 0, v \textit{ is } G\textit{-unstable}\}.$$

▶ **Example 4.5.** *Recall the tensor scaling action, in which the group $G = \mathrm{SL}(n)^d$ acts on $(\mathbb{C}^n)^{\otimes d}$ via the representation*

$$\pi_{n,d}\colon \mathrm{SL}(n)^d \to \mathrm{GL}\left((\mathbb{C}^n)^{\otimes d}\right), \ (g_1, \ldots, g_d) \mapsto g_1 \otimes \cdots \otimes g_d\,.$$

Similar computations to those in Example B.2 show that the set of weights of $\pi_{n,d}$ is

$$\Omega(\pi_{n,d}) = \Omega_{n,d} = \left\{\varepsilon_i \mid i \in [n]\right\}^d \subseteq (\mathbb{R}^n)^d.$$

Therefore, the weight margin $\gamma_{\mathrm{T}}(\pi_{n,d})$ is the margin $\gamma(\Omega_{n,d})$ for the array scaling problem from Theorem 1.3 and Theorem 2.1. Moreover, the moment map μ_G for $\pi_{n,d}$ can be computed in terms of the quantum marginals as described in the introduction, i.e. $\gamma_G(\pi_{n,d})$ is indeed the tensor gap.

The weight margin and the gap satisfy the following inequality.

▶ **Proposition 4.6.** *It holds that $\gamma_{\mathrm{T}}(\pi) \leqslant \gamma_G(\pi)$.*

Proof. Let $v \neq 0$ be G-unstable. Then there exists $k \in K$ such that $k \cdot v$ is T-unstable; see [51, Theorem 3.25]. By Proposition 4.2 we obtain

$$\|\mu_G(v)\|_F = \|\mu_G(k \cdot v)\|_F \geqslant \|\mu_{\mathrm{T}}(k \cdot v)\|_F \geqslant \gamma_{\mathrm{T}}(\pi)$$

where we used in the first equality that $\mu_G(k \cdot v) = k\mu_G(v)k^\dagger$. Therefore $\gamma_G(\pi) \geqslant \gamma_{\mathrm{T}}(\pi)$. ◀

This inequality motivates the next subsection.

[17] Gap and weight margin are well-defined, i.e. the minimum is attained. Indeed, the moment maps give rise to continuous maps on $\mathbb{P}(V)$ and the non-zero G-unstable (respectively non-zero T-unstable) vectors form a projective subvariety of $\mathbb{P}(V)$; in particular they form a compact set.

4.2 Free sets of weights

Proposition 4.6 from the preceding subsection shows us that an upper bound for the weight margin $\gamma_{\mathrm{T}}(\pi)$ need not necessarily apply to the gap $\gamma_G(\pi)$. Still, many of our bounds in the commutative case (weight margin and diameter) transfer to the noncommutative case (gap and diameter). We use crucially the notion of a *free* subset of weights (or [24]). Freeness is also known as *strong orthogonality* [18].

▶ **Definition 4.7.** *Let $\pi\colon G \to \mathrm{GL}(V)$ be a rational representation with set of weights $\Omega(\pi)$.*

A subset $\Gamma \subseteq \Omega(\pi)$ is called free *if no two distinct elements of Γ differ by a root of G. In other words, $\Gamma \cap (\Gamma + \alpha) = \varnothing$ holds for all roots α of G.*

Furthermore, a vector $v \in V\backslash\{0\}$ is called free *if its support* $\mathrm{supp}(v) \subseteq \Omega(\pi)$ *is free.*

We transfer the results from the commutative to the noncommutative case with the upcoming Proposition 4.8. It is known that for vectors v with free support one has $\mu_G(v) = \mu_{\mathrm{T}}(v)$. This appears implicitly in [49, Lemma 7.1] and [24, Proposition 2.2], but we prove it below for completeness. We thank Visu Makam for pointing out to us that this equality still holds under a weaker condition on v, when the representation decomposes into orthogonal subrepresentations. This can be used to turn our weight margin upper bound for quivers into a gap upper bound (Theorem 4.25). This weaker condition also appears in [21, Theorem 6.5].

▶ **Proposition 4.8.** *Let $\pi\colon G \to \mathrm{GL}(V)$ be a rational representation and suppose $V = \bigoplus_{i=1}^k V_i$ is an orthogonal decomposition into G-subrepresentations with respect to the K-invariant inner product, that is used to define μ_{T} and μ_G. Let $v = (v_1, \ldots, v_k) \in V\backslash\{0\}$, $v_i \in V_i$ be such that all supports $\Gamma_i := \mathrm{supp}(v_i) \subseteq \Omega(\pi)$ are free. Then for all $t \in \mathrm{T}$ it holds that $\mu_G(t \cdot v) \in i\,\mathrm{Lie}(\mathrm{T}_K)$ and $\mu_G(t \cdot v) = \mu_{\mathrm{T}}(t \cdot v)$.*

If additionally $0 \notin \Delta_{\mathrm{T}}(v) = \mathrm{conv}(\Gamma)$, where $\Gamma = \bigcup_i \Gamma_i$, then the upper bound $\mathrm{dist}(0, \mathrm{conv}(\Gamma))$ for the weight margin $\gamma_{\mathrm{T}}(\pi)$ also applies to the gap, i.e. $\gamma_G(\pi) \leqslant \mathrm{dist}(0, \mathrm{conv}(\Gamma))$.

Proof. The action of T preserves the supports Γ_i, and in particular preserves their freeness. Hence, it suffices to show $\mu_G(v) \in i\,\mathrm{Lie}(\mathrm{T}_K)$, which immediately yields $\mu_G(v) = \mu_{\mathrm{T}}(v)$ by Proposition 4.2. Moreover, the orthogonality with respect to the K-invariant inner product shows $\mu_G(v) = H_1 \oplus \cdots \oplus H_k$, where $H_i = \mu_G^{(i)}(v_i)$ is given by the moment map $\mu_G^{(i)}$ of the G-module V_i if $v_i \neq 0$ and otherwise $H_i = 0$. The latter holds similarly for μ_{T}.

Therefore, we may assume $k = 1$, i.e. $v \neq 0$ has free support Γ. We write $v = \sum_{\omega\in\Gamma} v_\omega$ for $v_\omega \in V_\omega$. Then, for any root α of G and all $A \in i\,\mathrm{Lie}(K) \cap \mathrm{Lie}(G)_\alpha$ we have $\Pi(A)v_\omega = 0$ by $\Gamma \cap (\Gamma + \alpha) = \varnothing$ (i.e., freeness) and Proposition B.4. Thus, $\Pi(A)v = 0$ and $\mathrm{tr}\big(\mu_G(v)A\big) = 0$ for all roots α and all $A \in i\,\mathrm{Lie}(K) \cap \mathrm{Lie}(G)_\alpha$. With the root space decomposition $\mathrm{Lie}(G) = \mathrm{Lie}(\mathrm{T}) \oplus \bigoplus_\alpha \mathrm{Lie}(G)_\alpha$ (see also Example B.3) we conclude $\mu_G(v) \in i\,\mathrm{Lie}(\mathrm{T}_K)$. The first statement is proven.

For the second claim we note that indeed $\bigcup_i \Gamma_i = \mathrm{supp}(v)$. If additionally $0 \notin \mathrm{conv}(\Gamma) = \Delta_{\mathrm{T}}(v)$, then v is T-unstable. In particular, v is G-unstable and thus

$$\gamma_G(\pi) \leqslant \mathrm{dist}\big(0, \Delta_G(v)\big).$$

On the other hand, we have

$$\mathrm{dist}\big(0, \Delta_G(v)\big) = \inf_{g\in G} \|\mu_G(g\cdot v)\|_F \leqslant \inf_{t\in\mathrm{T}} \|\mu_G(t\cdot v)\|_F \overset{(*)}{=} \mathrm{dist}\big(0, \mathrm{conv}(\Gamma)\big),$$

where we used $\mu_G(t\cdot v) = \mu_{\mathrm{T}}(t\cdot v)$ in $(*)$. We conclude by combining the two inequalities. ◀

▶ **Remark 4.9.** *It is well-known that any rational representation $\pi\colon G \to \mathrm{GL}(V)$ can be decomposed into G-irreducible subrepresentations that are pairwise orthogonal with respect to the fixed K-invariant inner product. Proposition 4.8 shows that ensuring freeness on the irreducible subrepresentations suffices.*

We end the section with an interesting connection between the weight margin and the gap.

▶ **Proposition 4.10.** *Let $\pi\colon G \to \mathrm{GL}(V)$ be a rational representation and denote its m-fold direct sum by π^m.*

1. *The weight margin satisfies $\gamma_{\mathrm{T}}(\pi) = \gamma_{\mathrm{T}}(\pi^m)$ for all $m \geqslant 1$.*
2. *The gap satisfies $\gamma_G(\pi^m) \geqslant \gamma_G(\pi^{m+1})$ for all $m \geqslant 1$.*
3. *There exists some $m \leqslant \dim(V)$ such that $\gamma_G(\pi^m) = \gamma_{\mathrm{T}}(\pi^m) = \gamma_{\mathrm{T}}(\pi)$.*

Proof. We note that π^m is given by the action $g \cdot (v_1, \ldots, v_m) = (g \cdot v_1, \ldots, g \cdot v_m)$ on V^m. Furthermore, the K-invariant inner product $\langle \cdot, \cdot \rangle$ of V induces naturally a K-invariant product on V^m by

$$\langle (v_1, \ldots, v_m), (w_1, \ldots, w_m) \rangle_{V^m} := \sum_{i=1}^{m} \langle v_i, w_i \rangle.$$

For the first claim just note that the weight space decomposition for π^m is $V^m = \bigoplus_{\omega \in \Omega(\pi)} V^m_\omega$ and hence $\Omega(\pi^m) = \Omega(\pi)$.

For the second claim, let $(v_1, \ldots, v_m) \in V^m \backslash \{0\}$ be G-unstable such that $\|\mu_G(v_1, \ldots, v_m)\|_F = \gamma_G(\pi^m)$. Then $(v_1, \ldots, v_m, 0) \in V^{m+1} \backslash \{0\}$ is G-unstable as well, so $\|\mu_G(v_1, \ldots, v_m, 0)\|_F \geqslant \gamma_G(\pi^{m+1})$. Moreover, under the inner product $\langle \cdot, \cdot \rangle_{V^{m+1}}$ the first m copies of V are orthogonal to the last copy. Thus, $\mu_G(v_1, \ldots, v_m, 0)$ is the 2×2 block-diagonal matrix $\mathrm{diag}(\mu_G(v_1, \ldots, v_m), 0)$ and hence $\|\mu_G(v_1, \ldots, v_m, 0)\|_F = \|\mu_G(v_1, \ldots, v_m)\|_F = \gamma_G(\pi^m)$.

Finally, let $\Gamma = \{\omega_1, \ldots, \omega_m\} \subseteq \Omega(\pi)$ be such that $0 \notin \mathrm{conv}(\Gamma)$ and $\mathrm{dist}(0, \mathrm{conv}(\Gamma)) = \gamma_{\mathrm{T}}(\pi)$. We have $m \leqslant |\Omega(\pi)| \leqslant \dim(V)$ by the weight space decomposition $V = \bigoplus_{\omega \in \Omega(\pi)} V_\omega$. Now, for each $\omega_i \in \Gamma$ fix some weight vector $v_i \in V_{\omega_i} \backslash \{0\}$. Then $v := (v_1, \ldots, v_m) \in V^m$ satisfies the assumptions of Proposition 4.8, because $\Gamma_i = \{\omega_i\}$ is free and the distinct copies of V are orthogonal under $\langle \cdot, \cdot \rangle_{V^m}$. Thus, we obtain

$$\gamma_G(\pi^m) \leqslant \mathrm{dist}\big(0, \mathrm{conv}(\Gamma)\big) = \gamma_{\mathrm{T}}(\pi) = \gamma_{\mathrm{T}}(\pi^m),$$

but on the other hand $\gamma_G(\pi^m) \geqslant \gamma_{\mathrm{T}}(\pi^m)$ by Proposition 4.6. ◀

4.3 Freeness for tensors

We recall from Example 4.5 that $\pi_{n,d}$ denotes the natural representation of $G = \mathrm{SL}(n)^d$ on $(\mathbb{C}^n)^{\otimes d}$ and that the weight margin $\gamma_{\mathrm{T}}(\pi_{n,d})$ is the margin $\gamma(\Omega_{n,d})$ for the array scaling problem from Theorem 1.3 and Theorem 2.1. The purpose of this subsection is to prove the bounds for $\gamma_{\mathrm{T}}(\pi_{n,d})$ from Theorem 2.1 also for the gap $\gamma_G(\pi_{n,d})$.

▶ **Theorem 4.11.** *Let $\pi_{n,d}$ be the representation induced by the natural action of $G := \mathrm{SL}(n)^d$ on $(\mathbb{C}^n)^{\otimes d}$. Then the weight margin $\gamma_{\mathrm{T}}(\pi_{n,d})$ and the gap $\gamma_G(\pi_{n,d})$ can be bounded as follows:*

(a) *If $n = 2$ and $d \geqslant 3$, then $\gamma_{\mathrm{T}}(\pi_{2,d}) \leqslant \gamma_G(\pi_{2,d}) \leqslant 2^{-\frac{d}{2}+1}$.*

(b) *If $n \geqslant 3$ and $d = 3$, then $\gamma_{\mathrm{T}}(\pi_{n,3}) \leqslant \gamma_G(\pi_{n,3}) \leqslant 2^{-n+1}$.*

(c) *If $n \geqslant 3$ and $d = 6r - 3$ for some integer $r \geqslant 2$, then*

$$\gamma_{\mathrm{T}}(\pi_{n,d}) \leqslant \gamma_G(\pi_{n,d}) \leqslant \frac{\sqrt{6}}{(n-1)\sqrt{r}}\, 2^{-r(n-1)+1} \leqslant 2^{-r(n-1)+1} = 2^{-\frac{(d+3)(n-1)}{6}+1}.$$

Though the above theorem only applies to certain d, we can "pad" the tensors to obtain similar results for all $d \geqslant 3$. This is because bounds for $\gamma_G(\pi_{n,d})$ via free subsets of weights also hold for $\gamma_G(\pi_{n,d+2})$ and $\gamma_G(\pi_{n,d+3})$, see Proposition C.1. The missing case $n \geqslant 3$ and $d = 4$ is treated in Proposition C.2. Therefore, we can conclude Theorem 1.6 from the above Theorem 4.11.

Our main method for transfering the bounds from the commutative case (Theorem 2.1) to the noncommutative case is to use the concept of freeness in conjunction with Proposition 4.8. The following definition will be convenient for proving freeness of tensors.

▶ **Definition 4.12** (Free sets). *A set $M \subseteq [n]^d$ is called* free*, if $i = (i_1, \dots, i_d), j = (j_1, \dots, j_d) \in M$ with $i \neq j$ always implies $|\{i_l \neq j_l \mid l = 1, \dots, d\}| \geqslant 2$.*

▶ **Proposition 4.13.** *Let $M \subseteq [n]^d$ and denote the induced subset of weights by*

$$\Gamma_M := \{(\varepsilon_{i_1}, \dots, \varepsilon_{i_d}) \mid (i_1, \dots, i_d) \in M\} \subseteq (\mathbb{R}^n)^d.$$

Then M is a free set if and only if the set of weights $\Gamma_M \subseteq \Omega(\pi_{n,d})$ is free as in Definition 4.7.

Proof. We recall that Γ_M is free if and only if no two distinct elements of Γ_M differ by a root of $G = \mathrm{SL}(n)^d$, see Definition 4.7. Furthermore, remember that the roots of G are

$$(e_i - e_j, 0_n, \dots, 0_n), (0_n, e_i - e_j, 0_n, \dots, 0_n), \dots\dots, (0_n, \dots, 0_n, e_i - e_j) \in (\mathbb{R}^n)^d$$

for $i, j \in [n]$ with $i \neq j$; see also Example B.3. Now, if $M \subseteq [n]^d$ is not free, then there exist $i = (i_1, \dots, i_d), j = (j_1, \dots, j_d) \in M$ with $i \neq j$ such that they exactly differ one component. Without loss of generality we assume $i_1 \neq j_1$ and $i_l = j_l$ for $l = 2, \dots, n$. But then

$$(\varepsilon_{i_1}, \dots, \varepsilon_{i_d}) = (\varepsilon_{j_1}, \dots, \varepsilon_{j_d}) + (e_{i_1} - e_{j_1}, 0_n, \dots, 0_n),$$

and hence Γ_M is not free. Clearly, the argument can be inverted to show that if Γ_M is not free, then M is not free. ◀

The above proposition shows how the equality $\mu_G(t \cdot v) = \mu_{\mathrm{T}}(t \cdot v)$ of Proposition 4.8 can be verified directly for tensors. For tensors, the moment map components are the quantum marginals, and the equality $\mu_G(t \cdot v) = \mu_{\mathrm{T}}(t \cdot v)$ simply says that the quantum marginals are diagonal. Each off-diagonal entry of a quantum marginal is the inner product between distinct $d-1$-dimensional slices of a tensor, and if the support of the tensor is free then the supports of such slices are entirely disjoint - thus the quantum marginals are diagonal.

In the following two Propositions we show, that the subsets of weights, which witness the upper bounds for the (weight) margin in Theorem 2.1, are all free. Thereby, we will implicitly use Proposition 4.13.

▶ **Proposition 4.14.** *For $r \geqslant 2$ the rows of A_{2r} form a free subset of $[2]^{2r}$, i.e. $\Gamma_{2,2r}$ is free. Moreover, for $r \geqslant 1$ the set of weights $\Gamma_{2,2r+1}$ is free.*

Proof. Clearly, $\Gamma_{2,3} = \{\varepsilon_{1,1,1}, \varepsilon_{2,1,2}\}$ is free. Recall the constructions of $\Gamma_{2,2r}$ and $\Gamma_{2,2r+1}$ from Section 2.1. If $\Gamma_{2,2r}$ is free, then $\Gamma_{2,2r+1}$ is clearly also free. Thus, we are left to prove the former.

Consider A_{2r} as defined in Equation (6). We must show that distinct rows of A_{2r} differ in at least two entries for all $r \geqslant 2$. The claim is proven by induction on $r \geqslant 3$. For $r = 3$, we verify the claim by inspection of A_6. Let a_i be the i^{th} row of A_6; its definition is recalled in the left-hand table below. The right-hand table lists for each pair a_i, a_j with $i < j$ two distinct entries in which a_i and a_j differ, which shows the claim for $r = 3$.

entry	1	2	3	4	5	6
a_1	1	1	1	1	1	1
a_2	2	1	2	2	2	2
a_3	1	2	2	1	1	1
a_4	2	2	1	1	2	2
a_5	1	2	1	2	2	1
a_6	2	2	2	2	1	1

	a_2	a_3	a_4	a_5	a_6
a_1	1,3	2,3	1, 2	2,4	1,2
a_2		1,2	2,3	1,2	5,6
a_3			1,3	3,4	1, 4
a_4				1,4	3,4
a_5					1,3

In fact, the table also proves the claim for $r = 2$, since $a_1, \dots, a_4$ already pairwise differ in at least two of the first four entries.

Now assume that the claim holds for some fixed $r \geqslant 3$. Let a_i, a_j be distinct rows of A_{2r+2}; we will show they differ in at least two entries. If $1 \leqslant i < j \leqslant 2r$, then by our inductive hypothesis there is nothing to prove because the first $2r$ rows of A_{2r+2} contain A_{2r} as a submatrix.

To complete the proof, it is enough to show that the $4 \times (2r + 2)$ submatrix formed by restricting to the m^{th} block row, $m \in [r]$, and the last block row of A_{2r+2} satisfies the hypothesis, i.e. any two distinct rows of this submatrix differ in at least two entries. This is the case as restricting to its 1^{st}, m^{th} and last block columns yields a 4×6 submatrix of A_6 if $m \neq 1$, namely

$$\begin{pmatrix} B_2 & B_3 & B_1 \\ B_2 & B_2 & B_3 \end{pmatrix},$$

and a 4×4 submatrix equal to A_4 if $m = 1$. ◀

▶ **Proposition 4.15.** *For $n \geqslant 3$ the set $\mathfrak{W}_n \subseteq [n]^3$ is free, i.e. $\Gamma_{n,3} \subseteq \Omega(\pi_{n,3})$ is free. Furthermore, for $n \geqslant 3$ and $r \geqslant 2$ the set of weights $\Gamma_{n,6r-3} \subseteq \Omega(\pi_{n,6r-3})$ is free.*

Proof. We remind the reader that

$$\mathfrak{W}_n = \big\{(s, 1, s), (s, s, 1), (s-1, s, s) \mid s = 2, 3, \dots, n\big\}.$$

Let $x = (x_1, x_2, x_3), y = (y_1, y_2, y_3) \in \mathfrak{W}_n$ be such that $x \neq y$. We prove by a distinction of cases that x and y differ in at least two entries. First, we assume $x_1 = y_1$. Then $a := x_1 = y_1 \geqslant 2$, otherwise $x = (1, 2, 2) = y$ contradicts $x \neq y$. Thus $x, y \in \{(a, 1, a), (a, a, 1), (a, a+1, a+1)\}$ and we conclude that x and y differ in at least two entries as $x \neq y$. Second, we assume $x_1 \neq y_1$. There is nothing to show if $x_2 \neq y_2$, so we additionally assume $b := x_2 = y_2$. If $b = 1$, then we are done by $x = (x_1, 1, x_1)$ and $y = (y_1, 1, y_1)$. On the other hand, $b \geqslant 2$ yields $x, y \in \{(b, b, 1), (b-1, b, b)\}$ and as $x \neq y$ they differ in the first and third entry. This proves the first statement.

For the second claim, recall that

$$\Gamma_{n,6r-3} = \{\varepsilon_{\sigma(i),\sigma(j),\sigma(k)} \mid (i, j, k) \in \mathfrak{W}_{rn} \backslash \mathfrak{J}_r\},$$

where $\sigma \colon [rn] \to [n]^{2r-1}$ is injective, compare Remark 2.10. By the first part $\mathfrak{W}_{rn}$ is free and so is its subset $\mathfrak{W}_{rn} \backslash \mathfrak{J}_r$. Hence $\Gamma_{n,6r-3}$ is free as σ is injective. ◀

We are now ready to deduce Theorem 4.11.

Proof of Theorem 4.11. Recall that all the bounds in Theorem 4.11 hold for the weight margin $\gamma_{\mathrm{T}}(\pi)$ by Theorem 2.1. This was proven by exhibiting witness sets $\Gamma_{n,d} \subseteq \Omega(\pi_{n,d})$ such that $0 \notin \operatorname{conv}(\Gamma_{n,d})$, which gives the bound $\gamma_{\mathrm{T}}(\pi_{n,d}) \leqslant \operatorname{dist}(0, \operatorname{conv}(\Gamma_{n,d}))$. But if $\Gamma_{n,d}$ is free, then we even have

$$\gamma_G(\pi_{n,d}) \leqslant \operatorname{dist}\big(0, \operatorname{conv}(\Gamma_{n,d})\big)$$

by Proposition 4.8. By Proposition 4.14 the witness sets $\Gamma_{2,3}$ and $\Gamma_{2,2r}$, $\Gamma_{2,2r+1}$, $r \geqslant 2$ for Theorem 2.1(a) are free, which proves Theorem 4.11(a). Similarly, we conclude parts (b) and (c) with Proposition 4.15, which shows that for $n \geqslant 3$ and $r \geqslant 2$ the witness sets $\Gamma_{n,3}$ and $\Gamma_{n,6r-3}$ are free. ◀

4.4 Freeness for homogeneous polynomials

In the following we transfer the result from d-tensors to the natural $\mathrm{SL}(n)$ action on homogeneous d-forms in n variables. This representation is given by

$$\varrho_{n,d} \colon \mathrm{SL}(n) \to \mathrm{GL}\left(\mathbb{C}[x_1, \ldots, x_n]_d\right), \; g \mapsto \left(p(x) \mapsto p(g^{-1}x)\right).$$

Each monomial $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, given by a multi-index $\alpha = (\alpha_1, \ldots, \alpha_n) \in (\mathbb{Z}_{\geqslant 0})^n$ with $|\alpha| := \sum_i \alpha_i = d$, is a weight vector for $\varrho_{n,d}$ with weight $-\alpha + \frac{d}{n}\mathbb{1}_n$. Therefore

$$\Omega(\varrho_{n,d}) = \left\{ -\alpha + \frac{d}{n}\mathbb{1}_n \;\middle|\; \alpha \in (\mathbb{Z}_{\geqslant 0})^n \text{ with } |\alpha| = d \right\},$$

i.e. $\Omega(\varrho_{n,d}) = \Omega'$ from Equation (27) and the bounds from Corollary 2.13 apply to $\gamma_{\mathrm{ST}(n)}(\varrho_{n,d}) = \gamma(\Omega')$. If $n = dm$ for some integer $m \geqslant 1$, then we have $-\Omega(\pi_{m,d}) \subseteq \Omega(\varrho_{n,d})$.

▶ **Proposition 4.16.** *Let $n = dm$ for some integer $m \geqslant 1$. If $\Gamma \subseteq \Omega(\pi_{m,d})$ is free, then $-\Gamma \subseteq \Omega(\varrho_{n,d})$ is free.*

Proof. We prove the statement by contraposition. Assume that $-\Gamma \subseteq \Omega(\varrho_{n,d})$ is not free. Then there exists a root $\alpha = e_i - e_j \in \mathbb{R}^n$ of $\mathrm{SL}(n)$, where $i, j \in [n]$ with $i \neq j$, and two distinct weights $\omega, \omega' \in -\Gamma$ such that $\omega = \omega' + e_i - e_j$, equivalently $-\omega = -\omega' - e_i + e_j$. The latter equation enforces $-\alpha$ to be of the form

$$(0_m, \ldots, 0_m, e_k - e_l, 0_m, \ldots, 0_m) \in (\mathbb{R}^m)^d \cong \mathbb{R}^n \qquad \text{for some } k, l \in [m] \text{ with } k \neq l,$$

because $-\omega, -\omega' \in \Omega(\pi_{m,d})$. Thus, $-\alpha$ is a root of $\mathrm{SL}(m)^d$ and hence $\Gamma \subseteq \Omega(\pi_{m,d})$ is not free. ◀

As a consequence of the preceding Proposition we obtain bounds for the gap $\gamma_{\mathrm{SL}(n)}(\varrho_{n,d})$.

▶ **Theorem 4.17** (Gap for Polynomial scaling). *Let $d \geqslant 3$ and let $n = dm$ for some integer $m \geqslant 2$. Then there exists a constant $C > 0$, independent of n and d such that*

$$\gamma_{\mathrm{SL}(n)}(\varrho_{n,d}) \leqslant 2^{-Cdm} = 2^{-Cn}.$$

More concretely, for $d = 3$ and $m \geqslant 3$ it holds that

$$\gamma_{\mathrm{SL}(n)}(\varrho_{n,d}) \leqslant \operatorname{dist}\left(0, \Gamma_{m,3}\right) \leqslant 2^{-m+1} = 2^{-\frac{n}{3}+1},$$

and if $m \geqslant 3$ and $d = 6r - 3$ for some $r \geqslant 2$, we have

$$\gamma_{\mathrm{SL}(n)}(\varrho_{n,d}) \leqslant \operatorname{dist}\left(0, \Gamma_{m,6r-3}\right) \leqslant 2^{-r(m-1)+1} = 2^{-\frac{(d+3)(m-1)}{6}+1} \approx 2^{-\frac{n}{6}}.$$

Proof. We recall that Theorem 1.6 was proven by padding the results from Theorem 4.11. Thus, for each $m \geqslant 2$ and $d \geqslant 3$ the bound $\gamma_{\mathrm{SL}(m)^d}(\pi_{m,d}) \leqslant 2^{-Cmd}$ from Theorem 1.6 is witnessed by a free set of weights $\Gamma_{m,d} \subseteq \Omega(\pi_{m,d})$, i.e. $0 < \operatorname{dist}(0, \operatorname{conv}(\Gamma_{m,d})) \leqslant 2^{-Cdm}$. But then $0 \notin \operatorname{conv}(-\Gamma_{m,d})$ and $-\Gamma_{m,d} \subseteq \Omega(\varrho_{n,d})$ is free by Proposition 4.16. Therefore, Proposition 4.8 yields

$$\gamma_{\mathrm{SL}(n)}(\varrho_{n,d}) \leqslant \operatorname{dist}\left(0, \operatorname{conv}(-\Gamma_{m,d})\right) = \operatorname{dist}\left(0, \operatorname{conv}(\Gamma_{m,d})\right) \leqslant 2^{-Cdm}.$$

Similarly, we get the other bounds by using freeness of $\Gamma_{m,3}$ and, respectively, $\Gamma_{m,6r-3}$ (see Proposition 4.15) combined with the distance bounds Lemma 2.7 and Lemma 2.12, respectively. ◀

4.5 Freeness and diameter bound

In this section we show that the diameter lower bound of Theorem 1.1 generalizes to diameter bounds for the capacity Equation (4) over the noncommutative group $G = \mathrm{SL}(n)^d$. Many algorithms for computing the capacity have resorted to geodesically convex optimization - G can be viewed as a manifold on which $g \mapsto \|g \cdot v\|^2$ is geodesically convex. The distance between an element of g and the identity in this geometry is closely related to the condition number of the matrix g. The diameter bound question is the following: *given an input v and $\varepsilon > 0$, how large a ball in G about the identity must we optimize over to find an approximate minimizer $g \in G$ such that $\|g \cdot v\|^2 - \mathrm{cap}(v) \leqslant \varepsilon$?* In other words, how well-conditioned can we expect approximate minimizers to Equation (4) to be? This matters because all the algorithms we know start at the origin and take small steps in the manifold, and if all the high-precision solutions are far from the origin then such algorithms cannot reach any of them quickly.

Before tackling this question we must make our notions of distance more precise. The manifold we use is actually not G but rather the manifold P of Hermitian, positive-definite matrices in G. Indeed, we can write

$$\inf_{g \in G} \|g \cdot v\|^2 = \inf_{g \in G} \langle v, g^\dagger g \cdot v\rangle = \inf_{x \in P} \langle v, x \cdot v\rangle.$$

Thus we may instead optimize the function $f_v : g \mapsto \langle v, g \cdot v\rangle$ over P. The manifold P is a prototypical example of a *Hadamard manifold*, a complete, simply connected Riemannian manifold of non-positive sectional curvature [8]. For us, $G = \mathrm{SL}(n)^d$ for some d, and so P is just the set of d-tuples of positive-definite matrices of determinant 1. Even for $d = 1$, P contains a totally geodesic submanifold isometric to the hyperbolic plane; as such the volumes of balls grow exponentially in their radius.[18] The function $f_v : g \mapsto \|g \cdot v\|^2$ is convex along geodesics in this manifold [12][19]. The geodesics through a point $X \in P$ are given by $\gamma(t) = \sqrt{X} e^{Ht} \sqrt{X}$ for Hermitian H. The Riemannian gradient $\nabla \log f_v(g)$ of $\log f_v$ at $g \in P$ is given by the moment map $\mu_G(g \cdot v)$. The geodesic ball of radius R in P about the identity is given by

$$B_R := \{e^A : A \text{ traceless, Hermitian, } \|A\|_F \leqslant R\} \subseteq P.$$

In a slight abuse of notation, we define the geodesic ball in G (rather than P) to be KB_R, as in the introduction. The values taken by f_v over B_{2R} are the same as the values taken by $g \mapsto \|g \cdot v\|^2$ on KB_R. We now define diameter bounds.

▶ **Definition 4.18.** *The diameter bound $D_f(\varepsilon)$ for a function f on P and a real number $\varepsilon > 0$ is defined as the infimum over $R > 0$ such that*

$$\inf_{g \in B_R} f(g) \leqslant \varepsilon + \inf_{g \in P} f(g).$$

We will show that the diameter bound for the norm-squared function can grow faster than $\mathrm{poly}(n, \log(1/\varepsilon))$ for $d = 3$. Firstly, we need to review how diameter bounds for tensors in $(\mathbb{R}^n_{\geqslant 0})^d$ like that in Theorem 1.1 relate to diameter bounds for tensors in $(\mathbb{C}^{\otimes n})^d$ over $\mathrm{SL}(n)^d$

[18] The volume of a ball can be computed exactly [27], but the very crude bound of volume $\Omega(e^{\Theta(r) - O(n \log n)})$ for the geodesic ball of radius r can be proved elementarily. The manifold $\mathrm{PD}(n) \cap \mathrm{SL}(n)$ contains the hyperbolic plane as a totally geodesic submanifold, in which the ball of radius r has area $e^{\Theta(r)}$ [15]. This shows the ball of radius r in $\mathrm{PD}(n) \cap \mathrm{SL}(n)$ contains $\Omega(e^{\Theta(r)})$ balls of radius 1, which themselves have volume at least $e^{-O(n \log n)}$ by comparison with the Euclidean ball.

[19] This was implicitly shown much earlier in [36].

and $\mathrm{ST}(n)^d$. Infimizing $f_v(g)$ over the subset $P \cap \mathrm{ST}(n)^d \subseteq P$, or the tuples of positive-definite diagonal matrices within $\mathrm{SL}(n)^d$, results in a program of the form Equation (2). For $d = 3$, for example,

$$\inf_{g \in P \cap \mathrm{ST}(n)^3} \langle v, g \cdot v \rangle = \mathrm{cap}(p) = \inf_{x \in (\mathbb{R}^n)^3} \sum_{\omega \in \Omega_{n,3}} p_\omega e^{\omega \cdot x} = \inf_{x \in (\mathbb{1}_n^\perp)^3} \sum_{\omega \in \Omega_{n,3}} p_\omega e^{\omega \cdot x} \tag{32}$$

where $\Omega_{n,3} = \{(\varepsilon_i, \varepsilon_j, \varepsilon_k) : i, j, k \in [n]\}$ and $p_{(\varepsilon_i,\varepsilon_j,\varepsilon_k)} = |v_{ijk}|^2$. The correspondence is exactly $g = e^{\mathrm{diag}(x)}$ for $x \in (\mathbb{1}_n^\perp)^3$, which implies the following.

▶ **Lemma 4.19.** *For all $\varepsilon > 0$, the diameter bound $D_f(\varepsilon)$ for the function $f_v : g \mapsto \langle v, g \cdot v \rangle$ on $\mathrm{ST}(n)^3$ is equal to the diameter bound $D_h(\varepsilon)$ of the function f_p where $p_{ijk} = |v_{ijk}|^2$, or*

$$f_p \colon (\mathbb{R}^n)^3 \to \mathbb{R}, \quad x \mapsto \sum_{i,j,k \in [n]} |v_{ijk}|^2 e^{(\varepsilon_i, \varepsilon_j, \varepsilon_k) \cdot x}.$$

Of course, there's nothing special about $d = 3$ here, and the lemma generalizes straightforwardly to other d. For instance, applying Lemma 4.19 for $d = 2$ shows that restricting operator scaling to diagonal matrices yields an instance of matrix scaling. We have shown how diameter bounds over $\mathrm{ST}(n)^d$ relate to those over $(\mathbb{R}^n)^d$. Now we complete the chain by showing how to relate diameter bounds over $\mathrm{SL}(n)^d$ to those over $\mathrm{ST}(n)^d$. We will show that tensors with free support (defined in Definition 4.12) have the same diameter bound over $\mathrm{SL}(n)^d$ as they do over $\mathrm{ST}(n)^d$, which by Theorem 1.1 and Lemma 4.19 we have shown can be superpolynomial. We then show that the construction from Section 3.2 is free.

▶ **Theorem 4.20.** *Let G denote $\mathrm{SL}(n)^d$, and let T denote $\mathrm{ST}(n)^d$. Suppose $\mu_{\mathrm{T}}(t \cdot v) = \mu_G(t \cdot v)$ for all $t \in \mathrm{T}$ (which holds if v has free support). Then for any $R > 0$ we have*

$$\inf_{g \in B_R} f_v(g) = \inf_{g \in \mathrm{T} \cap B_R} f_v(g),$$

where B_R denotes the geodesic ball of radius R about the identity in G.

Proof. Define $B := B_R$ and recall that P denotes the positive-definite matrices in G. Let $f : P \to \mathbb{R}$ be given by $f : g \to \langle v, g \cdot v \rangle$. Clearly $\inf_{g \in B} f(g) \leqslant \inf_{g \in \mathrm{T} \cap B} f(g)$. We must show the converse inequality. Let $g^* := \arg\min_{g \in B} f(g)$. Recall that P is a Hadamard manifold. Define T_+ to be $\mathrm{T} \cap P$. Let πg^* denote the projection of g^* to T_+, that is, the closest point in T_+ to g^*. As T_+ is a geodesically convex set, projections to T_+ are unique and distances decrease under the projection [8, Theorem 2.1.12]. Thus, $\pi g^* \in B$. If we can show that $f(\pi g^*) \leqslant f(g^*)$ then the proof is complete.

Let $g^* = \exp_{\pi g^*}(x)$ for some x in the tangent space $T_{\pi g^*} P$ to P at πg^*. That is, $\gamma \colon [0, 1] \to P$, $t \mapsto \exp_{\pi g^*}(tx)$ is the geodesic between πg^* and g^*. Then, in the local inner product $\langle \cdot, \cdot \rangle_{\pi g^*}$ at πg^*, x is orthogonal to the tangent space $T_{\pi g^*} \mathrm{T}_+ \subseteq T_{\pi g^*} P$ of T_+ at πg^*, because πg^* is a local minimum of the geodesically convex function $d(g^*, \cdot)^2$ on T_+ and x is proportional to the gradient of $d(g^*, \cdot)^2$ at πg^*.

The function f is geodesically convex, and its gradient $\nabla f(\pi g^*)$ is proportional to the moment map $\mu_G(\pi g^* \cdot v)$. By the assumption that $\mu_{\mathrm{T}}(t \cdot v) = \mu_G(t \cdot v)$ for all $t \in \mathrm{T}$, $\mu_G(\pi g^* \cdot v)$ is in $i\,\mathrm{Lie}(\mathrm{T}_K)$, which is precisely the tangent space of T_+ at πg^*. Thus

$$f(g^*) = f(\exp_{\pi g^*}(x)) \geqslant f(\pi g^*) + \langle x, \nabla f(\pi g^*) \rangle_{\pi g^*} = f(\pi g^*),$$

which completes the proof. ◀

▶ **Lemma 4.21.** *The support of the tensor p from Theorem 1.1 is free.*

Proof. Recall that a tensor in $(\mathbb{C}^n)^{\otimes 3}$ is free if and only if the supports of distinct rows of its weight matrix intersect in at most one element. The construction in Proposition 3.5 preserves freeness, so we can consider the case $n = 3(l+1)$ treated in the proof of Theorem 1.1. Recall that, in this case, the support of p is $\Omega_0' \cup \omega'$ where Ω_0' is the rows of a matrix M defined from the directed graph D_l. Each row in the matrix M corresponds to some edge D_l. Let us first verify that Ω_0' is free. Assuming the rows correspond to the same edge, they can be verified to have intersection in at most one element, because the nonzero entries of the three rows corresponding to an edge are contained in a 3×6 submatrix with the following form:

$$\begin{bmatrix} A & I \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Here the cells containing 1 are colored for readability. Now consider the case that the rows belong to two different edges. If the two edges share no vertices, then clearly the corresponding edges do not intersect. Because the graph is a directed tree, edges may only share a vertex which is the sink of at least one of the edges. If the vertex is a sink for both edges, then the nonzero entries in the 6 rows belonging to either edge (after permutation) take the form

$$\begin{bmatrix} 0 & A & I \\ A & 0 & I \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

If the shared vertex is a sink for only one edge, then the rows are

$$\begin{bmatrix} 0 & A & I \\ A & I & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

In all these cases it can be verified that supports of distinct rows intersect in at most one element. Lastly, we need to make sure that the intersection of the support of ω' with the support of any element of Ω_0' is at most one. Recall that ω' is defined to have entry one in each block corresponding to the leaves u_l, v_l, w_l in D_l. However, there are no edges between the leaves, so the support of no row can intersect that of ω' in more than one element. ◀

We are now nearly ready to prove Theorem 1.4. We would simply use the array p from the proof of Theorem 1.1, but setting $|v_{ijk}|^2 = p_{ijk}$ would not be solvable over the rationals. Therefore we must round $\sqrt{p_{ijk}}$, which requires some additional technical lemmas proven in Appendix E.

▶ **Lemma 4.22** (Rounding and diameter bounds). *Let $p, q : \Omega \to \mathbb{R}_{\geqslant 0}$ be positive functions on a finite set $\Omega \subseteq \mathbb{R}^m$. Suppose there is a set B such that*

$$\inf_{x \in B} f_p(x) \geqslant (1+\varepsilon) \operatorname{cap} p,$$

and let $M = \max\{1/q_\omega, 1/p_\omega : \omega \in \Omega\}$. Then

$$\inf_{x \in B} f_q(x) \geqslant ((1+\varepsilon)(1 - M\|p-q\|_\infty) - M\|p-q\|_1) \operatorname{cap} q.$$

▶ **Lemma 4.23** (Rounding and capacity). *Let $\Omega \subseteq \mathbb{R}^m$ be finite and let $p, q : \Omega \to \mathbb{R}_{\geqslant 0}$ be positive functions on Ω. Let $M_0 = \max_{\omega \in \Omega} 1/q_\omega$. Then*

$$\log \operatorname{cap} q \geqslant \log \operatorname{cap} p - M_0 \|p - q\|_\infty.$$

Proof of Theorem 1.4. First recall that the values taken by $g \mapsto \|g \cdot v\|^2$ on the geodesic ball KB_R in G are the same as the values taken by $f_v : g \mapsto \langle v, g \cdot v \rangle$ on B_{2R} in P. Thus it is enough to show that $f := f_v$ has diameter bound $D_f(\varepsilon) = \Omega(2^{n/3} \log(1/\varepsilon))$ for $\varepsilon \leqslant e^{-Cn^2 \log n}$.

We will apply Lemma 4.22 with p as in the proof of Theorem 1.1 and $q_{ijk} = |v_{ijk}|^2$, with v_{ijk} chosen so that v has the same support as p and $p_{ijk} - \delta < |v_{ijk}|^2 \leqslant p_{ijk}$ for δ small. Because v is free, by Theorem 4.20 the diameter bound for f_v is the same as the diameter bound for f_v over $\mathrm{ST}(n)^3$. By Lemma 4.19, this is the same as the diameter bound for f_q. It remains to show that $D_{f_q}(\varepsilon) = \Omega(2^{n/3} \log(1/\varepsilon))$. We will do this by relating $D_{f_q}(\varepsilon)$ to $D_{f_p}(\varepsilon)$; in particular we will show $D_{f_q}(\Omega(\varepsilon)) \geqslant D_{f_p}(\varepsilon)$.

Let $R = D_{f_p}(\varepsilon)$. We have $\inf_{x \in (\mathbb{R}^n)^3, \|x\| \leqslant R} f_p(x) \geqslant \operatorname{cap}(p) + \varepsilon = (1 + 2\varepsilon) \operatorname{cap}(p)$, recalling that $\operatorname{cap}(p) = 1/2$. By Lemma 4.22,

$$\inf_{x \in (\mathbb{R}^n)^3, \|x\| \leqslant R} f_q(x) \geqslant ((1 + 2\varepsilon)(1 - M\|p - q\|_\infty) - M\|p - q\|_1) \operatorname{cap}(q).$$

As $\operatorname{cap} q \leqslant 1/2$, if $M\|p - q\|_\infty \leqslant M\|p - q\|_1 \leqslant c\varepsilon$ for c a small enough constant, then we have $((1 + 2\varepsilon)(1 - M\|p - q\|_\infty) - M\|p - q\|_1) \operatorname{cap} q = \operatorname{cap} q + \Omega(\varepsilon)$, so

$$\inf_{x \in (\mathbb{R}^n)^3, \|x\| \leqslant R} f_q(x) \geqslant \operatorname{cap} q + \Omega(\varepsilon).$$

Thus $D_{f_q}(\Omega(\varepsilon)) \geqslant D_{f_p}(\varepsilon)$ assuming $M\|p - q\|_1 \leqslant c\varepsilon$. To ensure that this constraint is satisfied, choose v of bit complexity $O(\log n + \log(1/\varepsilon))$ such that $\|p - q\|_1 = \frac{c}{n}\varepsilon$. Because $p_{ijk} = \Omega(1/n)$ for i, j, k in the support of p by construction, we have $q_{ijk} = \Omega(1/n)$ for i, j, k in the support of q and hence $M = O(n)$. Thus $M\|p - q\|_1 \leqslant c\varepsilon$. Applying Lemma 4.23 together with our assumptions about the size of $p - q$ and the fact that $\operatorname{cap}(q) = \operatorname{cap}(v)$ implies the final claim that $\operatorname{cap}(v) \geqslant 1/4$ and that $1 \geqslant \|v\| \geqslant 1/2$. ◀

Finally, we remark that the same diameter bound holds for $d \geqslant 3$ for *tuples* of tensors. We note that if $v \in (\mathbb{C}^n)^{\otimes 3}$ has free support, then so does the tensor $v \otimes e_l \otimes \ldots \otimes e_l \subset (\mathbb{C}^n)^{\otimes d}$ for $d \geqslant 3$. By Proposition 4.8, the tuple $w \in ((\mathbb{C}^n)^{\otimes d})^n$ given by

$$w_l = \frac{1}{n}\, v \otimes e_l \otimes \ldots \otimes e_l \text{ for } l \in [n]$$

has $\mu_T(t \cdot v) = \mu_G(t \cdot v)$ for all $t \in \mathrm{ST}(n)^d$. The commutative problem obtained by restricting to $\mathrm{SL}(n)^d$ as in Lemma 4.19 is precisely f_q as in Corollary 3.7. As in the proof of Theorem 1.4, by Theorem 4.20, Lemma 4.19 and Corollary 3.7, we have the following.

▶ **Corollary 4.24.** *There is a constant $C > 0$ such that the following holds for all $d \geqslant 3$. For all $\varepsilon \leqslant \exp(-Cn^2 \log n)$, there is a tuple of tensors $w = w(\varepsilon) \in ((\mathbb{C}^n)^{\otimes d})^n$ with $O(n^2)$ nonzero entries of bit complexity $O(\log n + \log(1/\varepsilon))$, and a geodesic ball $B = B(\varepsilon)$ of radius $\Omega\left(2^{n/3} \log(1/\varepsilon)\right)$ about the identity in $\mathrm{SL}(n)^d$, such that*

$$\inf_{g \in B} \|g \cdot w\|^2 \geqslant \operatorname{cap}(v) + \varepsilon.$$

Moreover, it holds that $1/4 \leqslant \operatorname{cap}(w) \leqslant 1$ and $1/2 \leqslant \|w\| \leqslant 1$.

4.6 A bound on weight margin and gap for quivers

For $d \geqslant 2$ let Q_d be the quiver

$$1 \longleftarrow 2 \longrightarrow 3 \cdots\cdots\cdots d-2 \longrightarrow d-1 \longleftarrow d \qquad \text{if } d \text{ even}$$

$$1 \longrightarrow 2 \longleftarrow 3 \cdots\cdots\cdots d-2 \longrightarrow d-1 \longleftarrow d \qquad \text{if } d \text{ odd.}$$

and let $Q_d^{(k)}$ be the quiver one obtains from Q_d by adding $k-1$ additional copies of each arrow in Q_d. As before, let $G = \mathrm{SL}(n)^d$ and $\mathrm{T} = \mathrm{ST}(n)^d$. Then G acts on the quiver Q_d with dimension vector $(n, \ldots, n)$ as described in the introduction. We denote the corresponding representation by π_d. Note that the action of G on $Q_d^{(k)}$ with dimension vector $(n, \ldots, n)$ is given by π_d^k. In this subsection we prove a bound on the weight margin of π_d and on the gap of π_d^n. The bound on $\gamma_G(\pi_d^n)$ is thanks to the refinement of freeness in Proposition 4.8 pointed out by Visu Makam.

▶ **Theorem 4.25.** *Let $n, d \geqslant 2$ and denote the natural action of $G = \mathrm{SL}(n)^d$ on the quiver Q_d with dimension vector $(n, \ldots, n)$ by $\pi_d \colon \mathrm{SL}(n)^d \to \mathrm{GL}(V_d)$, where $V_d = (\mathbb{C}^{n \times n})^{d-1}$. The representation π_d^n corresponds to the G-action on the quiver $Q_d^{(n)}$ with dimension vector $(n, \ldots, n)$. It holds that*

$$\gamma_{\mathrm{T}}(\pi_d) \leqslant (n-1)^{-d+1} \qquad \textit{and} \qquad \gamma_G(\pi_d^n) \leqslant (n-1)^{-d+1}.$$

▶ **Remark 4.26.** *Before proving the theorem, we point out a few consequences.*

1. *Theorem 4.25 shows that $\gamma_{\mathrm{T}}(\pi_d)^{-1}$ and $\gamma_G(\pi_d^n)^{-1}$ are not polynomially bounded with respect to $\dim V_d = (d-1)n^2$ and $\dim \mathrm{SL}(n)^d = d(n^2-1)$. Instead we see for fixed n and $d \to \infty$ an exponential behaviour in the number of vertices d. Thus, our bound shows that the exponential behaviour in d cannot be avoided in general lower bounds for quiver actions like [12, Theorem 6.21 Item 4]. The latter applied to π_d shows $\gamma_{\mathrm{T}}(\pi_d) \geqslant n^{-d^2-(3/2)d}(dn+1)^{-d}$.*
2. *The proof of Theorem 4.25 below shows that for the bound on the gap it is enough to consider the quiver $Q_d^{(n-1)}$ with an additional n^{th} arrow from d to $d-1$.*
3. *The ideas presented below can be adjusted to prove similar bounds for other dimension vectors. For example, one can show that the gap for the* SL*-action on $Q_d^{(2)}$ with dimension vector $(1, 3, 3, \ldots, 3, 2)$ is inverse exponential in d. This aligns with an algebraic barrier for this action; the invariants that cut out the null cone for this action have exponential degree [19, Proposition 1.5].*
4. *The quiver Q_d is of finite representation type and has no oriented cycles. Therefore, the null-cone membership problem for π_d can be solved in polynomial-time by algebraic algorithms.[20] This means Q_d is an example where the weight margin is very small but there still exist efficient algorithms. Can the existence of efficient algorithms still be explained by a large gap in this case? This leads to the following interesting open question.*

▶ **Problem 4.27.** *Is the gap $\gamma_G(\pi_d)$ inverse polynomial in n* and *d?*

A positive answer would provide an interesting example, since in this case the weight margin of π_d would be *significantly* smaller than the gap of π_d.

We now introduce several lemmas needed to prove Theorem 4.25. Note that the set of weights of π_d viewed as a subset of $(\mathbb{R}^n)^d$ is

$$\Big\{ \big((-1)^d \varepsilon_i, (-1)^{d-1} \varepsilon_j, 0, \ldots, 0\big), \big(0, (-1)^{d-1} \varepsilon_i, (-1)^{d-2} \varepsilon_j, 0, \ldots, 0\big), \ldots, \big(0, \ldots, 0, \varepsilon_i, -\varepsilon_j\big) \mid i, j \in [n] \Big\}.$$

[20] Personal communication with Visu Makam. There does not seem to be an explicit reference in the literature.

We define recursively the subsets of weights

$$\Gamma_2 := \{(\varepsilon_i, -\varepsilon_j) \mid i \in [n-1],\, j \in [n]\} \subseteq \Omega(\pi_2) \subseteq \mathbb{R}^{2n}$$

$$\text{for } d \geqslant 3,\ \Gamma_d := \left\{\left((-1)^d \varepsilon_i, (-1)^{d-1}\varepsilon_n, 0_n, \ldots, 0_n\right) \mid i \in [n-1]\right\} \cup (\{0_n\} \times \Gamma_{d-1}) \subseteq \Omega(\pi_d) \subseteq \mathbb{R}^{dn}\,.$$

▶ **Remark 4.28.** *We note that for $d \geqslant 2$, Γ_d is not* not *free. For instance, we can always write*

$$(0_n, \ldots, 0_n, \varepsilon_1, -\varepsilon_1) = (0_n, \ldots, 0_n, \varepsilon_1, -\varepsilon_2) + (0_n, \ldots, 0_n, 0_n, e_2 - e_1),$$

i.e. the weights $(0_n, \ldots, 0_n, \varepsilon_1, -\varepsilon_1)$, $(0_n, \ldots, 0_n, \varepsilon_1, -\varepsilon_2) \in \Gamma_d$ differ by the root $(0_n, \ldots, 0_n, 0_n, e_2 - e_1)$ of $\mathrm{SL}(n)^d$. Therefore, we cannot *deduce a bound on the gap $\gamma_G(\pi_d)$ via Proposition 4.8. However, the latter allows us to deduce at least a bound on the gap of π_d^n.*

In the next two lemmas we show that Γ_d witnesses the bound on $\gamma_{\mathrm{T}}(\pi_d)$ and afterwards we use Proposition 4.8 to transfer this bound to $\gamma_G(\pi_d^n)$.

▶ **Lemma 4.29.** *For all $d \geqslant 2$ it holds that $0 \notin \operatorname{conv}(\Gamma_d)$.*

Proof. We prove the statement by induction on $d \geqslant 2$. For $d = 2$, just note that any element in $\operatorname{conv}(\Gamma_2) \subseteq \mathbb{R}^{2n}$ has value $-1/n$ in the n-th entry. In particular, $0 \notin \operatorname{conv}(\Gamma_2)$. For $d \geqslant 3$ let

$$x = \sum_{\omega \in \Gamma_d} \lambda_\omega\, \omega\,, \quad \lambda_\omega \geqslant 0$$

be a convex combination of the elements in Γ_d. Assume there is an $i \in [n-1]$ such that for

$$\omega_i := \left((-1)^d \varepsilon_i, (-1)^{d-1}\varepsilon_n, 0_n, \ldots, 0_n\right)$$

one has $\lambda_{\omega_i} > 0$. Then the n-th entry of x is non-zero, since ω_i has n-th entry $(-1)^{d+1}/n$ and all (other) $\omega \in \Gamma_d$ have $(-1)^{d+1}/n$ or zero as n-th entry. On the other hand, if $\lambda_{\omega_i} = 0$ for all $i \in [n-1]$, then $x \in \{0_n\} \times \operatorname{conv}(\Gamma_{d-1})$. By induction hypothesis on $d-1$ we necessarily have $x \neq 0$. ◀

▶ **Lemma 4.30.** *For $d \geqslant 2$ it holds that $x_d := \lambda_d\left((-1)^{d-1}\varepsilon_n, 0_n, \ldots, 0_n\right) \in \operatorname{conv}(\Gamma_d)$, where*

$$\lambda_d := \left(\sum_{i=1}^{d-1} (n-1)^i\right)^{-1}.$$

In particular, $\|x_d\|_2 < |\lambda_d| \leqslant (n-1)^{-d+1}$.

Proof. We proceed by induction on $d \geqslant 2$. In the case $d = 2$, consider the convex combination

$$\sum_{i=1}^{n-1} \sum_{j=1}^{n} \frac{1}{(n-1)n} (\varepsilon_i, -\varepsilon_j) = \frac{1}{n-1}(-\varepsilon_n, 0_n) = x_2\,,$$

where we used (5). Now assume the claim is proven for some $d \geqslant 2$, hence

$$\lambda_d\left(0_n, (-1)^{d-1}\varepsilon_n, 0_n, \ldots, 0_n\right) \in \{0_n\} \times \operatorname{conv}(\Gamma_d) \subseteq \operatorname{conv}(\Gamma_{d+1}). \tag{33}$$

Setting $\mu := (n-1)\lambda_{d+1}\lambda_d^{-1}$ we have $\mu\lambda_d = (n-1)\lambda_{d+1}$ and $\mu + (n-1)\lambda_{d+1} = 1$. Together with (5) and (33) we deduce $x_{d+1} \in \operatorname{conv}(\Gamma_{d+1})$ via

$$\mu\,\lambda_d\left(0_n, (-1)^{d-1}\varepsilon_n, 0_n, \ldots, 0_n\right) + \lambda_{d+1} \sum_{i=1}^{n-1} \left((-1)^{d+1}\varepsilon_i, (-1)^d \varepsilon_n, 0_n, \ldots, 0_n\right) = x_{d+1}.$$

This ends the induction. Finally, $\|x_d\|_2 < |\lambda_d|$ follows from $\|\varepsilon_n\|_2 < 1$. ◀

Proof of Theorem 4.25. By Lemma 4.29 and Lemma 4.30 we have

$$\gamma_{\mathrm{T}}(\pi_d) \leqslant (n-1)^{-d+1}.$$

With the fact $\Omega(\pi_d) = \Omega(\pi_d^n)$ and with Proposition 4.8 we transfer this bound to the gap of π_d^n. To do so, we note that the natural inner product on $V_d^n = (\mathbb{C}^{n\times n})^{n(d-1)}$, given by the trace inner product on each $\mathbb{C}^{n\times n}$ copy, is invariant under the action of $K = \mathrm{SU}(n)^d$. Clearly, distinct $\mathbb{C}^{n\times n}$ copies are orthogonal under this inner product. Thus, to be able to apply Proposition 4.8 it is enough to assign to each $\mathbb{C}^{n\times n}$ copy, i.e. to each arrow of $Q_d^{(n)}$, a matrix M_i such that $\operatorname{supp}(M_i)$ is free and $\Gamma_d = \bigcup_i \operatorname{supp}(M_i)$.

For this, we consider the $n \times n$ matrices

$$M := \begin{pmatrix} I_{n-1} & 0 \\ 0 & 0 \end{pmatrix} \qquad \text{and} \qquad P := \begin{pmatrix} 0 & I_{n-1} \\ 1 & 0 \end{pmatrix},$$

and $E_{i,j}$ is the matrix with (i,j)-entry one and all other entries zero. Then $E_{i,i}P = E_{i,\sigma(i)}$, where $\sigma\colon [n] \to [n]$ is the cycle $(1\ 2\ \ldots\ n)$. Therefore, for $k \in [n]$ we have

$$\operatorname{supp}\left(MP^{k-1}\right) = \left\{\left(0_{n(d-2)}, \varepsilon_i, -\varepsilon_{\sigma^{k-1}(i)}\right) \mid i \in [n-1]\right\} \text{ and } \{0_{n(d-2)}\} \times \Gamma_2 = \bigcup_{k\in[n]} \operatorname{supp}\left(MP^{k-1}\right).$$

For fixed k, $i_1 \neq i_2$ implies $\sigma^{k-1}(i_1) \neq \sigma^{k-1}(i_2)$, so any distinct elements of $\operatorname{supp}(MP^{k-1})$ differ in the last two $\mathbb{R}^n$-components. Hence, each $\operatorname{supp}(MP^{k-1})$ is free and we assign $M, MP, \ldots, MP^{n-1}$ to the n arrows that go from vertex d to vertex $d-1$. For $l \in [d-2]$, we assign to the n arrows between the vertices l and $l+1$ each of the matrices $E_{1,n}, E_{2,n}, \ldots, E_{n-1,n}$ at least once. (Exactly one of the latter matrices is assigned to two of these arrows.) Clearly, the support of $E_{i,n}$, $i \in [n-1]$ is free as it contains just one weight. By construction, this assignment does the job. Moreover, the argument shows that $n-1$ arrows between the vertices l and $l+1$, $l \in [d-2]$, suffice. ◀

References

1 P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, Princeton, NJ, 2008. With a foreword by Paul Van Dooren. doi:10.1515/9781400830244.

2 Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 172–181, 2018.

3 Zeyuan Allen-Zhu, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Much faster algorithms for matrix scaling. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 890–901. IEEE, 2017.

4 Noga Alon and Văn H. Vũ. Anti-Hadamard matrices, coin weighing, threshold gates, and indecomposable hypergraphs. *Journal of Combinatorial Theory, Series A*, 79(1):133–160, 1997.

5 Jason M. Altschuler and Enric Boix-Adsera. Polynomial-time algorithms for Multimarginal Optimal Transport problems with structure, 2020. arXiv:2008.03006.

6 Carlos Améndola, Kathlén Kohn, Philipp Reichenbach, and Anna Seigal. Invariant theory and scaling algorithms for maximum likelihood estimation, 2020. arXiv:2003.13662.

7 Nima Anari, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials, entropy, and a deterministic approximation algorithm for counting bases of matroids. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 35–46. IEEE, 2018.

8 Miroslav Bacák. *Convex analysis and optimization in Hadamard spaces*, volume 22. Walter de Gruyter GmbH & Co KG, 2014.

9 Rajendra Bhatia. *Positive definite matrices*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2007.

10 Peter Bürgisser, Matthias Christandl, Ketan D. Mulmuley, and Michael Walter. Membership in moment polytopes is in NP and coNP. *SIAM J. Comput.*, 46(3):972–991, 2017. `doi:10.1137/15M1048859`.

11 Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 883–897. IEEE, 2018.

12 Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Towards a theory of non-commutative optimization: geodesic first and second order methods for moment maps and polytopes, 2019. `arXiv:1910.12375`.

13 Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Alternating Minimization, Scaling Algorithms, and the Null-Cone Problem from Invariant Theory. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:20, 2018. `doi:10.4230/LIPIcs.ITCS.2018.24`.

14 Peter Bürgisser, Yinan Li, Harold Nieuwboer, and Michael Walter. Interior-point methods for unconstrained geometric programming and scaling problems, 2020. `arXiv:2008.12110`.

15 James W. Cannon, William J. Floyd, Richard Kenyon, Walter R. Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31:59–115, 1997.

16 Michael B. Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained Newton's method and interior point methods. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 902–913. IEEE, 2017.

17 Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

18 Jiri Dadok and Victor Kac. Polar representations. *J. Algebra*, 92(2):504–524, 1985. `doi:10.1016/0021-8693(85)90136-X`.

19 Harm Derksen and Visu Makam. Degree bounds for semi-invariant rings of quivers. *J. Pure Appl. Algebra*, 222(10):3282–3292, 2018. `doi:10.1016/j.jpaa.2017.12.007`.

20 Harm Derksen and Visu Makam. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *Algebra Number Theory*, 14(10):2791–2813, 2020. `doi:10.2140/ant.2020.14.2791`.

21 Harm Derksen and Visu Makam. An exponential lower bound for the degrees of invariants of cubic forms and tensor actions. *Adv. Math.*, 368:107136, 25, 2020. `doi:10.1016/j.aim.2020.107136`.

22 Michael A Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 527–542. Springer, 2013.

23 Cole Franks and Ankur Moitra. Rigorous Guarantees for Tyler's M-estimator via quantum expansion, 2020. `arXiv:2002.00071`.

24 Matthias Franz. Moment polytopes of projective G-varieties and tensor products of symmetric group representations. *J. Lie Theory*, 12(2):539–549, 2002.

25 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 109–117. IEEE, 2016.

26 Ankit Garg, Christian Ikenmeyer, Visu Makam, Rafael Oliveira, Michael Walter, and Avi Wigderson. Search Problems in Algebraic Complexity, GCT, and Hardness of Generators for Invariant Rings. In *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:17, 2020. `doi:10.4230/LIPIcs.CCC.2020.12`.

27 X. Gual-Arnau and A. M. Naveira. Volume of tubes in noncompact symmetric spaces. *Publ. Math. Debrecen*, 54(3-4):313–320, 1999.

28 V. Guillemin and S. Sternberg. Convexity properties of the moment mapping. II. *Invent. Math.*, 77(3):533–546, 1984. doi:10.1007/BF01388837.

29 Leonid Gurvits. Classical complexity and quantum entanglement. *Journal of Computer and System Sciences*, 69(3):448–484, 2004.

30 Leonid Gurvits. Combinatorial and algorithmic aspects of hyperbolic polynomials, 2004. arXiv:math/0404474.

31 Brian C. Hall. *Lie groups, Lie algebras, and representations*, volume 222 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2003. An elementary introduction. doi:10.1007/978-0-387-21554-9.

32 Linus Hamilton and Ankur Moitra. No-go Theorem for Acceleration in the Hyperbolic Plane, 2021. arXiv:2101.05657.

33 Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *Conference on Learning Theory*, pages 354–375, 2013.

34 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Comput. Complexity*, 27(4):561–593, 2018. doi:10.1007/s00037-018-0165-7.

35 Bahman Kalantari and Leonid Khachiyan. On the complexity of nonnegative-matrix scaling. *Linear Algebra and its applications*, 240:87–103, 1996.

36 George Kempf and Linda Ness. The length of vectors in representation spaces. In *Algebraic geometry (Proc. Summer Meeting, Univ. Copenhagen, Copenhagen, 1978)*, volume 732 of *Lecture Notes in Math.*, pages 233–243. Springer, Berlin, 1979.

37 M. K. Kravtsov and V. E. Lukshin. On some properties of noninteger vertices of a three-index axial transportation polytope. *Tr. Inst. Matematiki NAN Belarusi*, 13(2):31–36, 2005.

38 V. M. Kravtsov. Combinatorial properties of noninteger vertices of a polytope in a three-index axial assignment problem. *Kibernet. Sistem. Anal.*, 43(1):33–44, 189, 2007. doi:10.1007/s10559-007-0023-0.

39 Tianyi Lin, Nhat Ho, Marco Cuturi, and Michael I. Jordan. On the complexity of approximating multimarginal optimal transport, 2019. arXiv:1910.00152.

40 Nathan Linial and Zur Luria. On the vertices of the d-dimensional Birkhoff polytope. *Discrete & Computational Geometry*, 51(1):161–170, 2014.

41 Tomasz Maciążek and Adam Sawicki. Critical points of the linear entropy for pure L-qubit states. *Journal of Physics A: Mathematical and Theoretical*, 48(4):045305, January 2015. doi:10.1088/1751-8113/48/4/045305.

42 Tomasz Maciążek and Adam Sawicki. Asymptotic properties of entanglement polytopes for large number of qubits. *Journal of Physics A: Mathematical and Theoretical*, 51(7):07LT01, January 2018. doi:10.1088/1751-8121/aaa4d7.

43 Ketan Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

44 David Mumford. *Geometric Invariant Theory.* Ergebnisse der Mathematik und ihrer Grenzgebiete, Neue Folge, Band 34. Springer-Verlag, Berlin-New York, 1965.

45 Linda Ness. A stratification of the null cone via the moment map. *Amer. J. Math.*, 106(6):1281–1329, 1984. With an appendix by David Mumford. doi:10.2307/2374395.

46 Beresford N. Parlett and Christian Reinsch. Balancing a matrix for calculation of eigenvalues and eigenvectors. In *Handbook for Automatic Computation*, pages 315–326. Springer, 1971.

47 Alexander Rusciano. A Riemannian Corollary of Helly's theorem. *J. Convex Anal.*, 27(4):1261–1275, 2020.

48 Mohit Singh and Nisheeth K. Vishnoi. Entropy, optimization and counting. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 50–59, 2014.

49 Reyer Sjamaar. Convexity properties of the moment mapping re-examined. *Adv. Math.*, 138(1):46–91, 1998. doi:10.1006/aima.1998.1739.

50 Damian Straszak and Nisheeth K. Vishnoi. Maximum entropy distributions: Bit complexity and stability. In *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2861–2891. PMLR, 25–28 June 2019. arXiv:1711.02036.

51 Nolan R. Wallach. *Geometric Invariant Theory: Over the real and complex numbers.* Universitext. Springer, Cham, 2017. doi:10.1007/978-3-319-65907-7.

52 Hermann Weyl. *The classical groups: their invariants and representations*, volume 45. Princeton university press, 1946.

53 Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638. PMLR, 2016.

A Notation

f_p	the function $\mathbb{R}^m \to \mathbb{R}_{\geqslant 0}, x \mapsto \sum_{\omega\in\Omega} p_\omega e^{\omega\cdot x}$, see Equation (2)
$\operatorname{cap}(p)$	the capacity of a non-negative function p on a finite set $\Omega \subseteq \mathbb{R}^m$, see Equation (2)
$\operatorname{cap}(v)$	the capacity of a vector v under a group action, see Equation (4)
$[n]$	the set $\{1, 2, \ldots, n\}$
0_n	the zero vector in $\mathbb{R}^n$
e_i	the i^{th} canonical unit vector in $\mathbb{R}^n$
$\mathbb{1}_n$	the all-ones vector in $\mathbb{R}^n$
$\mathbb{1}_n^\perp$	the orthogonal complement of $\mathbb{1}_n$ in $\mathbb{R}^n$, i.e. $\left\{(v_1, \ldots, v_n) \in \mathbb{R}^n \colon \sum_i v_i = 0\right\}$
ε_i	the vector $e_i - \frac{1}{n}\mathbb{1}_n$
I_n	the $n \times n$ identity matrix
$\operatorname{dist}(0, S)$	the distance from the origin to the set S
$\operatorname{conv}(S)$	the convex hull of S in $\mathbb{R}^n$
$\operatorname{Aff}(S)$	the affine hull of S in $\mathbb{R}^n$
$\pi_{n,d}$	the representation for d-dimensional tensor scaling
$\Omega(\pi)$	the set of weights of a representation π
$\Omega_{n,d} = \Omega(\pi_{n,d})$	the set $\{\varepsilon_i \colon i \in [n]\}^d$ corresponding to d-dimensional array scaling; equal to the set of weights of the tensor scaling representation $\pi_{n,d}$, see Example 4.5
$\gamma(\Omega)$	the *margin* of the finite set $\Omega \subseteq \mathbb{R}^m$, see Definition 1.2
$\gamma_{\mathrm{T}}(\pi)$	the *weight margin* of a representation π, i.e. $\gamma(\Omega(\pi))$, see Definition 4.3
$\gamma_G(\pi)$	the *gap* of a representation π, see Definition 4.3
$\operatorname{tr}(A)$	the trace of a square matrix A
$D_f(\varepsilon)$	the diameter bound of a function f for $\varepsilon > 0$, see Definition 3.1 respectively Definition 4.18
$\lVert A\rVert_F$	the Frobenius norm of a square matrix A
e^A	the exponential of a square matrix A
$\operatorname{Lie}(G)$	the Lie algebra of a matrix Lie group G
$\operatorname{GL}(n)$	the group of invertible *complex* $n \times n$ matrices
$\operatorname{SL}(n)$	the group of invertible *complex* $n \times n$ matrices with determinant one
$\operatorname{ST}(n)$	the group of diagonal invertible *complex* $n \times n$ matrices with determinant one
$\operatorname{SU}(n)$	the group of unitary matrices of size $n \times n$ and determinant one
$\operatorname{Herm}(n)$	the set of complex Hermitian $n \times n$ matrices
$\operatorname{GL}(V)$	the group of $\mathbb{C}$-linear, bijective maps $V \to V$, where V is a $\mathbb{C}$-vector space

B Representation theory background

In this section we briefly recall some representation theory. All the concepts we present here actually work in the very general setting of reductive groups and their rational representations, see e.g. [12, section 2]. For the sake of clarity and concreteness we stick to the special case needed in this paper, i.e. the reductive group $\operatorname{SL}(n)^d := \operatorname{SL}(n) \times \cdots \times \operatorname{SL}(n)$ with $d \geqslant 1$ many copies of $\operatorname{SL}(n)$.

We call a Euclidean-closed subgroup $H \subseteq \operatorname{GL}(n)$ a *matrix Lie group*. Indeed, such an H is naturally a Lie group (c.f. [31, Theorem 1.19]) with real *Lie algebra*

$$\operatorname{Lie}(H) := \left\{ A \in \mathbb{C}^{n\times n} \mid \forall\, t \in \mathbb{R} \colon e^{tA} \in H \right\}.$$

The Lie bracket for $\mathrm{Lie}(H)$ is the commutator $[A, B] := AB - BA$. Moreover, for $d \geqslant 1$ the product $H^d := H \times \cdots \times H$ becomes a matrix Lie group via block-diagonal embedding into $\mathrm{GL}(dn)$, i.e.

$$H^d \hookrightarrow \mathrm{GL}(dn), \quad (h_1, \ldots, h_d) \mapsto \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_d \end{pmatrix}$$

Then the Lie algebra of H^d is $\mathrm{Lie}(H)^d = \mathrm{Lie}(H) \times \cdots \times \mathrm{Lie}(H)$ block-diagonally embedded into $\mathbb{C}^{dn \times dn}$. If $G \subseteq \mathrm{GL}(n)$ is another matrix Lie group, then $G \cap H$ is again a matrix Lie group with Lie algebra $\mathrm{Lie}(G \cap H) = \mathrm{Lie}(G) \cap \mathrm{Lie}(H)$.

▶ **Example B.1.** *The groups* $\mathrm{GL}(n)$, $\mathrm{SL}(n)$, $\mathrm{U}(n)$ *and* $\mathrm{GT}(n)$ *are matrix Lie groups with Lie algebras*

$$\mathrm{Lie}(\mathrm{GL}(n)) = \mathbb{C}^{n \times n} \qquad \mathrm{Lie}(\mathrm{U}(n)) = \{A \in \mathbb{C}^{n \times n} \mid A^\dagger = -A\} = i \,\mathrm{Herm}(n)$$
$$\mathrm{Lie}(\mathrm{SL}(n)) = \{A \in \mathbb{C}^{n \times n} \mid \mathrm{tr}(A) = 0\} \qquad \mathrm{Lie}(\mathrm{GT}(n)) = \{A \in \mathbb{C}^{n \times n} \mid A \textit{ diagonal matrix}\}.$$

Therefore, also $\mathrm{SU}(n)$, $\mathrm{ST}(n)$ *and* $\mathrm{U}(n) \cap \mathrm{ST}(n)$ *are matrix Lie groups and their Lie algebras are obtained by corresponding intersections of the above Lie algebras. In particular, we have*

$$\mathrm{Lie}(\mathrm{U}(n) \cap \mathrm{ST}(n)) = \big\{ i \,\mathrm{diag}(x_1, \ldots, x_n) \mid x_j \in \mathbb{R}, x_1 + \ldots + x_n = 0 \big\}.$$

Thus, we can identify $i \,\mathrm{Lie}(\mathrm{U}(n) \cap \mathrm{ST}(n))$ *with the orthogonal complement* $(\mathbb{1}_n)^\perp \subseteq \mathbb{R}^n$ *of the all-ones vector* $\mathbb{1}_n$.

In the following, let $G := \mathrm{SL}(n)^d$ for some $d \geqslant 1$. Then $K := \mathrm{SU}(n)^d$ is a maximal compact subgroup of G, and $\mathrm{T} := \mathrm{ST}(n)^d$ and $\mathrm{T}_K := K \cap \mathrm{T}$ are maximal tori of G and K, respectively. As explained above, we think of all these groups as matrix Lie subgroups of $\mathrm{GL}(dn)$, and hence of their Lie algebras as subsets of $\mathbb{C}^{dn \times dn}$.

A *rational representation* of $G = \mathrm{SL}(n)^d$ is a group morphism $\pi \colon G \to \mathrm{GL}(V)$, such that in some basis of V the matrix entries of $\pi(g) \in \mathrm{GL}(V)$ are polynomials in the matrix entries of g.[21] Such a rational representation of G induces a representation of the Lie algebras by

$$\Pi \colon \mathrm{Lie}(G) \to \mathrm{End}(V), \quad A \mapsto \left.\frac{d}{dt}\right|_{t=0} \pi\left(e^{tA}\right)$$

with the property $\pi(e^A) = e^{\Pi(A)}$ for all $A \in \mathrm{Lie}(G)$. Restricting π to the commutative subgroup T induces a so-called *weight space decomposition* of V. That is, there is some finite set $\Omega(\pi) \subseteq i \,\mathrm{Lie}(\mathrm{T}_K)$ and a decomposition $V = \bigoplus_{\omega \in \Omega(\pi)} V_\omega$ into non-zero subspaces such that each $\omega \in \Omega(\pi)$ and any $v_\omega \in V_\omega$ satisfy

$$\forall A \in \mathrm{Lie}(\mathrm{T}) \colon \quad \pi\left(e^A\right) v_\omega = e^{\mathrm{tr}(A\omega)} v_\omega$$

or, equivalently,

$$\forall A \in \mathrm{Lie}(\mathrm{T}) \colon \quad \Pi\left(A\right) v_\omega = \mathrm{tr}(A\omega) v_\omega.$$

The elements $\omega \in \Omega(\pi)$ are called *weights* of π and the $v_\omega \in V_\omega$ are called *weight vectors*. Considering Example B.1 we frequently use the identification $i \,\mathrm{Lie}(\mathrm{T}_K) \cong (\mathbb{1}_n^\perp)^d$, where $\mathbb{1}_n^\perp$ is the orthogonal complement of $\mathbb{1}_n$ in $\mathbb{R}^n$. We note that for $\omega \in i \,\mathrm{Lie}(\mathrm{T}_K) \subseteq \mathbb{C}^{dn \times dn}$ the Frobenius norm $\|\omega\|_F$ becomes under this identification the 2-norm $\|\omega\|_2$ in $(\mathbb{R}^n)^d$.

[21] In other words, π is a morphism of affine algebraic groups.

▶ **Example B.2.** *Let $d = 1$. The group $G = \mathrm{SL}(n)$ acts on $\mathbb{C}^n$ by left-multiplication, which induces the rational representation $\pi\colon \mathrm{SL}(n) \to \mathrm{GL}(n), g \mapsto g$ with corresponding Lie algebra representation $\Pi\colon \mathrm{Lie}(\mathrm{SL}(n)) \to \mathbb{C}^{n\times n}, A \mapsto A$. For $i \in [n]$ we set*

$$\varepsilon_i := e_i - \frac{1}{n}\mathbb{1}_n \in \mathbb{1}_n^{\perp} \subseteq \mathbb{R}^n.$$

For all $A = \mathrm{diag}(a_1, \ldots, a_n) \in \mathrm{Lie}(\mathrm{T})$ and all $i \in [n]$

$$\pi\left(e^A\right) e_i = \mathrm{diag}(e^{a_1}, \ldots, e^{a_n}) e_i = e^{a_i} e_i \overset{(*)}{=} e^{\mathrm{tr}(A\,\mathrm{diag}(\varepsilon_i))} e_i$$

where we used $a_1 + \ldots + a_n = 0$ in $(*)$. Thus, $\varepsilon_i \in \mathbb{1}_n^{\perp} \cong i\,\mathrm{Lie}(\mathrm{T}_K)$ is a weight of π with weight vector e_i. Since $\mathbb{C}^n = \bigoplus_i \mathbb{C}e_i$, we deduce $\Omega(\pi) = \{\varepsilon_i \mid i \in [n]\}$.

▶ **Example B.3.** *Of particular importance in representation theory is the* adjoint representation. *That is, $G = \mathrm{SL}(n)^d$ acts on its Lie algebra by conjugation $\mathrm{Ad}\colon G \to \mathrm{GL}(\mathrm{Lie}(G)), g \mapsto (A \mapsto gAg^{-1})$, which induces the representation of Lie algebras $\mathrm{ad}\colon \mathrm{Lie}(G) \mapsto \mathrm{End}(\mathrm{Lie}(G)), A \mapsto (B \mapsto [A, B])$. The non-zero weights $\alpha \in \Omega(\mathrm{Ad})$ are called* roots *of G and the weight spaces $\mathrm{Lie}(G)_\alpha$ are called* root spaces.

Let $d = 1$ and for $i, j \in [n]$ denote by $E_{i,j}$ the matrix with entry one at position i, j and all other entries being zero. Then for $i, j \in [n]$ with $i \neq j$ and for all $A = \mathrm{diag}(a_1, \ldots, a_n), B \in \mathrm{Lie}(\mathrm{T})$ we compute

$$\mathrm{ad}(A)E_{i,j} = [A, E_{i,j}] = (a_i - a_j)E_{i,j} = \mathrm{tr}\left(A\,\mathrm{diag}(e_i - e_j)\right)E_{i,j},$$
$$\mathrm{ad}(A)(B) = [A, B] = 0.$$

Since $0_n, e_i - e_j \in \mathbb{1}_n^{\perp} \cong i\,\mathrm{Lie}(\mathrm{T}_K)$, we deduce $e_i - e_j \in \Omega(\mathrm{Ad})$ with weight vector $E_{i,j}$ and $0_n \in \Omega(\mathrm{Ad})$ with weight vector $B \in \mathrm{Lie}(\mathrm{T})$. Therefore, the set of roots of $\mathrm{SL}(n)$ is $\{e_i - e_j \mid i, j \in [n], i \neq j\}$, because $\mathrm{Lie}(G) = \mathrm{Lie}(\mathrm{T}) \oplus \bigoplus_{i\neq j} \mathbb{C}E_{i,j}$.

More generally, one can deduce that the roots of $G = \mathrm{SL}(n)^d$ are the

$$(e_i - e_j, 0_n, \ldots, 0_n), (0_n, e_i - e_j, 0_n, \ldots, 0_n), \ldots\ldots, (0_n, \ldots, 0_n, e_i - e_j) \in (\mathbb{R}^n)^d$$

for $i, j \in [n]$ with $i \neq j$ and that $\mathrm{Lie}(G) = \mathrm{Lie}(\mathrm{T}) \oplus \bigoplus_\alpha \mathrm{Lie}(G)_\alpha$.

We need the following property of roots, see e.g. [31, Lemma 7.11].

▶ **Proposition B.4.** *Let α be a root of $G = \mathrm{SL}(n)^d$ and let $\pi\colon G \to \mathrm{GL}(V)$ be a rational representation of G. If V_ω is the weight space of some weight $\omega \in \Omega(\pi)$, then*

$$\Pi\big(\mathrm{Lie}(G)_\alpha\big)(V_\omega) \subseteq V_{\omega+\alpha},$$

where $V_{\omega+\alpha} := \{0\}$, if $\omega + \alpha \notin \Omega(\pi)$.

C Padding for tensor margin and tensor gap

The Theorems 2.1 and 4.11 only give for all $n \geqslant 2$ bounds for certain sub-families of $\{(n, d) \mid d \geqslant 3\}$. Still, we can deduce Theorems 1.3 and 1.6 via some padding on the number of tensor factors d; that padding is provided in Proposition C.1 below. Recall the representation for tensor scaling

$$\pi_{n,d}\colon \mathrm{SL}(n)^d \to \mathrm{GL}\left((\mathbb{C}^n)^{\otimes d}\right), \ (g_1, \ldots, g_d) \mapsto g_1 \otimes \cdots \otimes g_d,$$

which set of weights is $\Omega(\pi_{n,d}) = \Omega_{n,d} = \{\varepsilon_i \mid i \in [n]\}^d \subseteq (\mathbb{R}^n)^d$.

▶ **Proposition C.1.** *Let* $G := \mathrm{SL}(n)^d$ *and* $n, d \geqslant 1$. *Consider a set of weights* $\Gamma_{n,d} \subseteq \Omega_{n,d}$ *such that* $0 \notin \mathrm{conv}(\Gamma_{n,d})$, *i.e.* $\Gamma_{n,d}$ *witnesses the inequality* $\gamma(\Omega_{n,d}) = \gamma_{\mathrm{T}}(\pi_{n,d}) \leqslant \mathrm{dist}(0, \mathrm{conv}(\Gamma_{n,d}))$.

1. *Then* $\gamma(\Omega_{n,d+1}) \leqslant \mathrm{dist}\left(0, \mathrm{conv}(\Gamma_{n,d})\right)$. *Consequently,* $\gamma(\Omega_{n,d+1}) \leqslant \gamma(\Omega_{n,d})$.
2. *If additionally* $\Gamma_{n,d}$ *is free, then* $\gamma_G(\pi_{n,d+r}) \leqslant \mathrm{dist}\left(0, \mathrm{conv}(\Gamma_{n,d})\right)$ *for all* $r \geqslant 2$.

Proof. To prove the statement we set for $r \geqslant 1$

$$\Delta_r := \{(\varepsilon_i, \ldots, \varepsilon_i) \mid i \in [n]\} \subseteq (\mathbb{R}^n)^r \qquad \text{and} \qquad \Gamma_{n,d+r} := \Gamma_{n,d} \times \Delta_r \subseteq \Omega(\pi_{n,d+r}).$$

By Equation (5) we have $0 \in \mathrm{conv}(\Delta_r)$ and therefore

$$\mathrm{conv}(\Gamma_{n,d+r}) = \mathrm{conv}(\Gamma_{n,d}) \times \mathrm{conv}(\Delta_r) \supseteq \mathrm{conv}(\Gamma_{n,d}) \times \{0\}.$$

The latter implies

$$\mathrm{dist}\left(0, \mathrm{conv}(\Gamma_{n,d+r})\right) \leqslant \mathrm{dist}\left(0, \mathrm{conv}(\Gamma_{n,d})\right). \tag{34}$$

Clearly, $0 \in \mathrm{conv}(\Gamma_{n,d+r})$ implies $0 \in \mathrm{conv}(\Gamma_{n,d})$ or, by contraposition, the assumption $0 \notin \mathrm{conv}(\Gamma_{n,d})$ yields $0 \notin \mathrm{conv}(\Gamma_{n,d+r})$. The latter for $r = 1$ shows $\gamma_{\mathrm{T}}(\pi_{n,d+1}) \leqslant \mathrm{dist}\left(0, \mathrm{conv}(\Gamma_{n,d+1})\right)$ and we conclude the first assertion with Equation (34).

Assume in addition that $\Gamma_{n,d}$ is free and let $r \geqslant 2$. Considering Definition 4.12 and Proposition 4.13 we prove that also $\Gamma_{n,d+r}$ is free. For this, let $M \subseteq [n]^d$ be such that $\Gamma_M = \Gamma_{n,d}$ and consider $(x, i, \ldots, i), (y, j, \ldots, j) \in M \times [n]^r$ with $(x, i, \ldots, i) \neq (y, j, \ldots, j)$. If $x \neq y$, then x and y differ in at least two components by freeness of M. If $x = y$, then we have $i \neq j$ and so $(x, i, \ldots, i)$ and $(y, j, \ldots, j)$ differ in at least two components, using $r \geqslant 2$. This shows that $\Gamma_{n,d+r}$ is free for $r \geqslant 2$. Since also $0 \notin \mathrm{conv}(\Gamma_{n,d+r})$ we obtain with Proposition 4.8 that $\gamma_G(\pi_{n,d+r}) \leqslant \mathrm{dist}\left(0, \mathrm{conv}(\Gamma_{n,d+r})\right)$ holds for all $r \geqslant 2$. Finally, we deduce the second statement using Equation (34). ◀

▶ **Proposition C.2.** *For* $n \geqslant 3$ *it holds that* $\gamma_{\mathrm{T}}(\pi_{n,4}) \leqslant \gamma_G(\pi_{n,4}) \leqslant 2^{-n+1}$.

Proof. This result can be obtained by imitating the proof of Theorem 2.1(b) in subsection 2.2 by using

$$\Gamma_{n,4} := \{(\varepsilon_i, \varepsilon_j, \varepsilon_k, \varepsilon_i) \mid (i, j, k) \in \mathfrak{W}_n\} \subseteq \Omega(\pi_{n,4}).$$

Clearly, $0 \notin \mathrm{conv}(\Gamma_{n,4})$ as $0 \notin \mathrm{conv}(\Gamma_{n,3})$ by Lemma 2.8. Moreover, one can show with Lemma 2.5 (similar to the proof of Lemma 2.7) that

$$x := -\frac{1}{c\, 2^{n-1}}(\varepsilon_1, \varepsilon_1, \varepsilon_1, \varepsilon_1) \in \mathrm{conv}(\Gamma_{n,4}), \quad \text{where} \quad c = n - 2^{-n+1} \geqslant 2.$$

Thus, $\|(\varepsilon_1, \varepsilon_1, \varepsilon_1, \varepsilon_1)\| \leqslant \sqrt{4}$ implies $\|x\| \leqslant c^{-1} 2^{-n+1}\sqrt{4} \leqslant 2^{-n+1}$. This proves $\gamma_{\mathrm{T}}(\pi_{n,4}) \leqslant 2^{-n+1}$.

Since $\mathfrak{W}_n$ is free by Proposition 4.15, the set $\{(i, j, k, i) \mid (i, j, k) \in \mathfrak{W}_n\}$ is free. Hence, we conclude $\gamma_G(\pi_{n,4}) \leqslant 2^{-n+1}$ with Proposition 4.13 and Proposition 4.8. ◀

D Proof of Lemma 2.11

Proof. For the sake of contradiction assume that $0 \in \mathrm{Aff}(\Gamma_{n,6r-3})$. Then there are coefficients $a_s, b_s, c_s \in \mathbb{R}$, where $2 \leqslant s \leqslant rn$, such that $a_2 = \ldots = a_r = b_2 = \ldots = b_r = 0$, $\sum_s (a_s + b_s + c_s) = 1$ and

$$\sum_{s=2}^{rn} \left(a_s\, \varepsilon_{\sigma(s),\sigma(1),\sigma(s)} + b_s\, \varepsilon_{\sigma(s),\sigma(s),\sigma(1)} + c_s\, \varepsilon_{\sigma(s-1),\sigma(s),\sigma(s)}\right) = 0 \in (\mathbb{R}^n)^{6r-3}. \tag{35}$$

The bulk of our work will consist of proving the equations

$$b_2 + c_2 = b_3 + c_3 = \ldots = b_{rn} + c_{rn} \tag{36}$$

$$a_2 + c_2 = a_3 + c_3 = \ldots = a_{rn} + c_{rn}. \tag{37}$$

From here we will derive a contradiction. We now set about proving Equations (36) and (37). Rewrite the left-hand-side of Equation (35) as the collection for $k \in [2r-1]$ of the following affine linear combinations of $\varepsilon_1, \ldots, \varepsilon_n$ in $\mathbb{R}^n$:

$$\sum_{s=2}^{rn} \big(a_s\, \varepsilon_{\sigma_k(s)} + b_s\, \varepsilon_{\sigma_k(s)} + c_s\, \varepsilon_{\sigma_k(s-1)}\big) = 0 \tag{38}$$

$$\sum_{s=2}^{rn} \big(a_s\, \varepsilon_{\sigma_k(1)} + b_s\, \varepsilon_{\sigma_k(s)} + c_s\, \varepsilon_{\sigma_k(s)}\big) = 0 \tag{39}$$

$$\sum_{s=2}^{rn} \big(a_s\, \varepsilon_{\sigma_k(s)} + b_s\, \varepsilon_{\sigma_k(1)} + c_s\, \varepsilon_{\sigma_k(s)}\big) = 0. \tag{40}$$

If we expand this expressions as affine linear combinations of the ε_l, then by Lemma 2.2 the coefficient of ε_l must be n^{-1} for all $l \in [n]$. Translating this for equations (38), (39) and (40) respectively with $2 \leqslant l \leqslant n$ and $k \in [r]$, and using for $j \in [r]$ that

$$\sigma_k\big(r(l-1) + j - k + 1\big) = \left\lceil \frac{(r(l-1) + j - k + 1) + (k-1)}{r} \right\rceil = l \tag{41}$$

we get

$$\forall\, k \in [r], l \in \{2, 3, \ldots, n\}: \qquad \sum_{j=1}^{r} \big(a_{r(l-1)+j-k+1} + b_{r(l-1)+j-k+1} + c_{r(l-1)+j-k+2}\big) = \frac{1}{n} \tag{42}$$

$$\forall\, k \in [r], l \in \{2, 3, \ldots, n\}: \qquad \sum_{j=1}^{r} \big(b_{r(l-1)+j-k+1} + c_{r(l-1)+j-k+1}\big) = \frac{1}{n} \tag{43}$$

$$\forall\, k \in [r], l \in \{2, 3, \ldots, n\}: \qquad \sum_{j=1}^{r} \big(a_{r(l-1)+j-k+1} + c_{r(l-1)+j-k+1}\big) = \frac{1}{n} \tag{44}$$

respectively, where we set $c_{rn+1} := 0$. Fixing some $l \geqslant 2$ and subtracting Equation (43) with $k = 1$ from Equation (43) for $k = 2$, we find a telescoping sum that reduces to $b_{r(l-1)} + c_{r(l-1)} = b_{rl} + c_{rl}$. Indeed, subtracting the two yields

$$\begin{aligned} 0 &= \sum_{j=1}^{r} \big(b_{r(l-1)+j-1} + c_{r(l-1)+j-1}\big) - \sum_{j=1}^{r} \big(b_{r(l-1)+j} + c_{r(l-1)+j}\big) \\ &= \sum_{j=0}^{r-1} \big(b_{r(l-1)+j} + c_{r(l-1)+j}\big) - \sum_{j=1}^{r} \big(b_{r(l-1)+j} + c_{r(l-1)+j}\big) \\ &= \big(b_{r(l-1)} + c_{r(l-1)}\big) - \big(b_{rl} + c_{rl}\big). \end{aligned}$$

More generally, for $k \in [r-1]$ combining (43) for k and $k \leftarrow k+1$, implies $b_{rl-k+1} + c_{rl-k+1} = b_{r(l-1)-k+1} + c_{r(l-1)-k+1}$ for all $l = 2, \ldots, n$, i.e. for every $k \in [r-1]$ we have

$$c_{r-k+1} = b_{r-k+1} + c_{r-k+1} = b_{2r-k+1} + c_{2r-k+1} = \ldots = b_{rn-k+1} + c_{rn-k+1}. \tag{45}$$

We are still missing the value $k = 0$, or the equations

$$b_{r+1} + c_{r+1} = b_{2r+1} + c_{2r+1} = \ldots = b_{r(n-1)+1} + c_{r(n-1)+1}. \tag{46}$$

We obtain this by subtracting, for $l = 2, \ldots, n$, (43) for $k = 1$ and l from (43) with $k = r$ and $l \leftarrow l + 1$. Indeed,

$$\begin{aligned} 0 &= \sum_{j=1}^{r} \big(b_{rl+j-r+1} + c_{rl+j-r+1}\big) - \sum_{j=1}^{r} \big(b_{r(l-1)+j} + c_{r(l-1)+j}\big) \\ &= \sum_{j=2}^{r+1} \big(b_{r(l-1)+j} + c_{r(l-1)+j}\big) - \sum_{j=1}^{r} \big(b_{r(l-1)+j} + c_{r(l-1)+j}\big) \\ &= \big(b_{rl+1} + c_{rl+1}\big) - \big(b_{r(l-1)+1} + c_{r(l-1)+1}\big). \end{aligned}$$

Lastly, we are missing the equations $b_2+c_2 = b_3+c_3 = \ldots = b_{r+1}+c_{r+1}$ for Equation (36). We have not yet used in Equation (39) the values $k = r + m$ with $m \in [r - 1]$. For this we note that

$$\sigma_{r+m}\big(j\big) = 2 \quad \text{for} \;\; j \in \{r - m + 1\} \cup \{r + 2, r + 3, \ldots, 2r\}.$$

We use this equation to apply Lemma 2.2 to (39) for ε_2 and $k = r + m$ with $m \in [r - 1]$ to obtain

$$b_{r-m+1} + c_{r-m+1} + \sum_{j=2}^{r} \big(b_{r+j} + c_{r+j}\big) = \frac{1}{n}.$$

We need one more equation to eliminate the right-hand term, so we use the following. Lemma 2.2 applied to equation (43) for $k = 1$ and $l = 2$ yields

$$\sum_{j=1}^{r} \big(b_{r+j} + c_{r+j}\big) = \frac{1}{n}.$$

Subtracting this equation from the previous one yields, $b_{r-m+1} + c_{r-m+1} = b_{r+1} + c_{r+1}$ for all $m = 1, \ldots, r - 1$. Together with the equations (45) and (46) we conclude Equation (36). Analogously, (40) and (44) can be used to obtain Equation (37).

To get a contradiction we show that $a_s = b_s = c_s = 0$ for all $s = 2, 3, \ldots, rn$. For this, we set $a := \sum_s a_s$ and $b := \sum_s b_s$. Equation (41) still applies for $l = 1, k = 1$, so Lemma 2.2 applied to the coefficient of ε_1 in (38), in (39) and in (40) respectively for $k = 1$ gives

$$\sum_{j=1}^{r} c_{j+1} = \frac{1}{n}, \qquad a + \sum_{j=1}^{r-1} c_{j+1} = \frac{1}{n} \qquad \text{and} \qquad b + \sum_{j=1}^{r-1} c_{j+1} = \frac{1}{n}$$

respectively. Subtracting the second equation from the first gives $a = c_{r+1}$, and reasoning analogously for the third yields $a = b = c_{r+1}$. Moreover, (43) with $k = r$ and $l = 2$ is $\sum_{j=1}^{r}(b_{j+1} + c_{j+1}) = n^{-1}$. Using the latter together with $b_2 = \ldots = b_r = 0$ and $\sum_{j=1}^{r} c_{j+1} = n^{-1}$ yields $b_{r+1} = 0$ and similarly $a_{r+1} = 0$ via (44) with $k = r$ and $l = 2$.

Since now also $a_{r+1} = b_{r+1} = 0$, the equation (42) with $k = r$ and $l = 2$ simplifies to $\sum_{j=1}^{r} c_{j+2} = n^{-1}$. In conjunction with $\sum_{j=1}^{r} c_{j+1} = n^{-1}$ we deduce $c_2 = c_{r+2}$ and hence $b_{r+2} = 0 = a_{r+2}$ by (36) and (37). But now (42) with $k = r-1$ and $l = 2$ is $\sum_{j=1}^{r} c_{j+3} = n^{-1}$ and together with $\sum_{j=1}^{r} c_{j+2} = n^{-1}$ we get $c_3 = c_{r+3}$. Continuing inductively we obtain

$$\forall\, j \in [r]\colon \quad c_{j+1} = c_{r+j+1} \quad \text{and} \quad a_{r+j+1} = b_{r+j+1} = 0$$

via (42) with $l = 2$, $k \in [r]$ and via (36), (37). Then (42) with $k = r$ and $l = 3$ simplifies to $\sum_{j=1}^{r} c_{r+j+2} = n^{-1}$ and together with $n^{-1} = \sum_{j=1}^{r} c_{j+1} = \sum_{j=1}^{r} c_{r+j+1}$ we have $c_{r+2} = c_{2r+2}$. Hence, $b_{2r+2} = 0 = a_{2r+2}$ via (36) respectively (37). Continuing inductively in the outlined manner with equation (42) for $k \in [r]$, $l = 3, \dots, n$ and with the equations (36) and (37) we conclude $a_s = b_s = 0$ for all $s = 2, 3 \dots, rn$, so $a = b = 0$. Finally, (36) implies $c_{r+1} = c_s$ for all $s = 2, \dots, rn$, but $c_{r+1} = b = 0$ giving the desired contradiction. ◀

E Padding and rounding for diameter bounds

We begin with the proof of Proposition 3.5. We prove it only for $d = 3$, but the proof goes through mutatis mutandis for all $d \geqslant 1$.

Proof of Proposition 3.5. Recall that q is the $n \times n \times n$ array such that $q_{ijk} = \frac{t}{n} p_{ijk}$ for $i, j, k \in [t]$, $q_{iii} = 1/n$ for $t + 1 \leqslant i \leqslant n$, and $q_{ijk} = 0$ otherwise. We may split the inputs $x, y, z \in \mathbb{1}_n^{\perp}$ into

$$\begin{aligned} x &= \left(x' + \alpha_1 \mathbb{1}_t, x'' - \frac{t}{n-t}\alpha_1 \mathbb{1}_{n-t}\right), \\ y &= \left(y' + \alpha_2 \mathbb{1}_t, y'' - \frac{t}{n-t}\alpha_2 \mathbb{1}_{n-t}\right), \\ z &= \left(z' + \alpha_3 \mathbb{1}_t, z'' - \frac{t}{n-t}\alpha_3 \mathbb{1}_{n-t}\right) \end{aligned}$$

where $x', y', z' \in \mathbb{R}^t, x'', y'', z'' \in \mathbb{R}^{n-t}$ each sum to zero; write $w = (x', y', z')$. As $\|(x, y, z)\|_2 \geqslant \|w\|_2$, it is enough to prove that $\|w\|_2$ is large for any approximate minimizer. By optimizing over α_i and x'', y'', z'' for fixed w, one computes that the optimum value for f_q for any fixed w is $f_p(w)^{t/n}$. To see this, write

$$f_q(x, y, z) = \frac{t e^{\alpha_1 + \alpha_2 + \alpha_3}}{n} f_p(w) + \frac{e^{-\frac{t}{n-t}(\alpha_1 + \alpha_2 + \alpha_3)}}{n} \sum_{i=t+1}^{n} e^{x_i'' + y_i'' + z_i''}.$$

First note that for fixed α_i's, the second term is minimized at $x'' = y'' = z'' = 0$ by Jensen's inequality. Furthermore, the value only depends on $\alpha := \alpha_1 + \alpha_2 + \alpha_3$. With $x'', y'', z'' = 0$, we have

$$f_q(x, y, z) = g(w, \alpha) := \frac{t e^{\alpha}}{n} f_p(w) + \frac{(n-t)}{n} e^{-\frac{t}{n-t}\alpha}.$$

Taking the derivative in α, we see that this is minimized when $f_p(w)e^{\alpha} = e^{-\frac{t}{n-t}\alpha}$, or $e^{\alpha} = f_p(w)^{-1/(1+\frac{t}{n-t})} = f_p(w)^{-\frac{n-t}{n}}$. Plugging this value in proves that the optimum is $f_p(w)^{t/n}$. By concavity of $x^{t/n}$, provided $f_p(w) \leqslant 1$ we have

$$f_p(w)^{t/n} - \operatorname{cap}(p)^{t/n} \geqslant \frac{1 - \operatorname{cap}(p)^{t/n}}{1 - \operatorname{cap}(p)} (f_p(w) - \operatorname{cap}(p)).$$

The first factor in the second term is the slope of the line from $(\operatorname{cap}(p), \operatorname{cap}(p)^{t/n})$ to $(1, 1)$. Thus for any $\varepsilon \leqslant 1 - \operatorname{cap}(p)$, any ε-approximate minimizer for f_q has norm at least that of some $\left(\frac{1-\operatorname{cap}(p)}{1-\operatorname{cap}(p)^{t/n}}\right)\varepsilon$-approximate minimizer for f_p. ◀

Proof of Lemma 4.23. We use the dual expression: $\log \operatorname{cap} q = -\inf_{\mathbb{E}_r \omega = 0} D_{KL}(r||q)$ where r ranges over probability distributions on Ω. In particular,

$$\log \operatorname{cap} q \geqslant -D_{KL}(r||q)$$

for any distribution r on Ω with $\mathbb{E}_r\omega = 0$. Let r be a probability distribution; calculate

$$\begin{aligned}\log\operatorname{cap} q \geqslant -D_{KL}(r||q) &= -D_{KL}(r||p) + D_{KL}(r||p) - D_{KL}(r||q)\\ &= -D_{KL}(r||p) + \sum_{\omega\in\Omega} r_\omega \log(r_\omega/p_\omega) - \sum_{\omega\in\Omega} r_\omega \log(r_\omega/q_\omega)\\ &= -D_{KL}(r||p) + \sum_{\omega\in\Omega} r_\omega(\log q_\omega - \log p_\omega).\end{aligned}$$

We lower bound $\log q_\omega - \log p_\omega \geqslant \frac{1}{q_\omega}(q_\omega - p_\omega)$ by applying the inequality $\log x \leqslant x - 1$ to $x = p_\omega/q_\omega$. Hence

$$\begin{aligned}\log\operatorname{cap} q &\geqslant -D_{KL}(r||p) + \sum_{\omega\in\Omega} r_\omega \frac{1}{q_\omega}(q_\omega - p_\omega)\\ &\geqslant -D_{KL}(r||p) - M_0\|p-q\|_\infty.\end{aligned}$$

Allowing $-D_{KL}(r||p)$ to tend to $\log\operatorname{cap} p$ completes the proof. ◀

Proof of Lemma 4.22. Applying Lemma 4.23 with the roles of p and q switched yields

$$\log\operatorname{cap} p \geqslant \log\operatorname{cap} q - M\|p-q\|_\infty.$$

Exponentiating both sides and applying the inequality $e^x \geqslant 1 + x$ yields $\operatorname{cap} p \geqslant (1 - M\|p - q\|_\infty)\operatorname{cap} q$. Thus

$$\inf_{x\in B} f_q(x) = \inf_{x\in S} f_q(x) \geqslant -\sup_{x\in S}|f_q(x) - f_p(x)| + \inf_{x\in S} f_p(x).$$

Note that the minimizer for f_q over B lies in the set $S := B \cap \{x :\ \forall\,\omega,\ q_\omega e^{x\cdot\omega} \leqslant f_q(0) = \|q\|_1\}$. For all $x \in S$, we have $e^{x\cdot\omega} \leqslant \operatorname{cap} q/p_\omega$ for all $\omega \in \Omega$, so

$$\begin{aligned}f_q(x) - f_p(x) &\leqslant \sum_{\omega\in\Omega}|p_\omega - q_\omega|e^{x\cdot\omega}\\ &\leqslant \sum_{\omega\in\Omega}|p_\omega - q_\omega|\|q\|_1/q_\omega)\\ &\leqslant \|p-q\|_1 M\|q\|_1.\end{aligned}$$

Combining the above inequality with the lower bound for $\operatorname{cap}(p)$,

$$\begin{aligned}\inf_{x\in B} f_q(x) &\geqslant -M\|q\|_1\|p-q\|_1 + (1+\varepsilon)\operatorname{cap} p\\ &\geqslant (1+\varepsilon)(1 - M\|p-q\|_\infty)\operatorname{cap} q - M\|p-q\|_1\|q\|_1.\end{aligned}$$ ◀

Communication Complexity with Defective Randomness

Marshall Ball
Computer Science Department, Columbia University, New York, NY, USA

Oded Goldreich
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel

Tal Malkin
Computer Science Department, Columbia University, New York, NY, USA

Abstract

Starting with the two standard model of randomized communication complexity, we study the communication complexity of functions when the protocol has access to a defective source of randomness. Specifically, we consider both the public-randomness and private-randomness cases, while replacing the commonly postulated perfect randomness with distributions over ℓ bit strings that have min-entropy at least $k \leq \ell$. We present general upper and lower bounds on the communication complexity in these cases, where the bounds are typically linear in $\ell - k$ and also depend on the size of the fooling set for the function being computed and on its standard randomized complexity.

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases Randomized Communication Complexity, Randomness Extraction, Min-Entropy

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.14

Funding This work was supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) via Contract No. 2019-1902070006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either express or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.
Marshall Ball: Partially supported by an IBM Research PhD Fellowship.
Oded Goldreich: Partially supported by an ISF grant number (Nr. 1146/18); research was conducted while he enjoyed the hospitality of the computer science department at Columbia University.

1 Introduction

While communication complexity is typically viewed as a tool for establishing lower bound on other models of computation, one may also view it as a study of (two-party) collaborations that can be carried out using a small amount of communication. The (two) parties participating in such a typical collaboration have a common goal, which is modeled as the computation of a function of their private inputs, and they wish to achieve it efficiently, which means using a small amount of communication (i.e., much smaller than required for communicating their entire input).

Given this perspective, one can ask whether randomness is helpful, and it is well-known that it is extremely helpful. For example, computing the equality function requires deterministic protocols that use a linear amount of communication (i.e., are not significantly better than the straightforward one), but can be performed by randomized protocols that use a constant amount of communication. The question addressed in this work is *what happens when the parties have at their disposal only defective sources of randomness?*

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 14; pp. 14:1–14:10
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 The Models

Our starting point is the two standard models of randomized communication complexity, which are closely related in the standard setting but may not be so in the current setting. In the standard public randomness model one postulates that the parties have access to a common (i.e., public) source of perfect randomness, whereas in the standard private randomness model one postulates that the each party has access to a private source of perfect randomness (which is uncorrelated to the other party's source). Indeed, in the standard setting, the public randomness model can easily emulate the private randomness model, and the opposite emulation is also possible at a very moderate cost [9].

We consider variants of these two models in which the postulated sources of perfect randomness are replaced by defective sources of randomness. In particular, we consider sources that output ℓ-bit long strings such that no string appears with probability exceeding 2^{-k}; that is, we consider sources of min-entropy k, with a focus on the case that $k \in [\Omega(\log \ell), \ell)$. A special case of interest is when the min-entropy rate (i.e., k/ℓ) is a constant smaller than 1 and the actual inputs are of length related to ℓ; yet, we shall consider the problem in almost full generality.

1.2 Our Results

We show that if the random sources available to the two parties are moderately defective in the sense that their min-entropy rate is a constant smaller than 1, then computing the equality function on strings of length comparable to the length of the random sources requires a linear amount of communication, just as in the case that one uses no randomness at all. More generally, we show that, when using defective sources of randomness, no improvement can be obtained over the lower bound on the communication complexity of deterministic protocol that follows by a "fooling set" argument (see Definition 2.2). The foregoing assertions refers to the case that $\min(m, \ell) = \Omega(n)$ and $k < (1 - \Omega(1)) \cdot \ell$.

▶ **Theorem 1.1** (general lower bounds). *Suppose that $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ has a fooling set of size 2^m, and let $k \leq \ell$.*

- (public randomness version): *If f is computed by a protocol whose only source of randomness is a public random string of length ℓ that has min-entropy k, then the protocol uses at least* $\min(m-1, \ell-k-1)/2$ *bits of communication.*
- (private randomness version): *If f is computed by a protocol in which the only source of randomness is provided by two independent random strings of length ℓ, each seen by one of the parties and having min-entropy k, then the protocol uses at least* $\min(m-1, \ell-k-1)/2$ *bits of communication.*

We stress that, in the current context, the two models (i.e., public-randomness and private-randomness) are not easily reducible to one another.[1] Recall that lower-bounding the size of a fooling set is one of the King's Roads for proving lower bounds on the deterministic communication complexity of functions.[2] In particular, equality has a fooling set of size 2^n. In general, the logarithm of the size of the fooling set seems a reasonable proxy for the deterministic communication complexity, but it is indeed interesting to ask whether a result as Theorem 1.1 holds with m replaced by the deterministic communication complexity of f.

[1] In particular, the fact that a random source of logarithmic length suffices does not apply here: we are given defective random sources of certain length, and cannot easily transform them to significantly shorter length.

[2] However, as shown by Dietzfelbinger, Hromkovic, and Schnitger [4], the deterministic complexity may be exponentially larger than the lower bound provided by any fooling set.

While Theorem 1.1 asserts that using a moderately defective random sources of length that is comparable to the input is useless, it does not rule out the benefit of sources that are either less defective or are shorter (i.e., have shorter length). It turns out that it is possible to benefit from the use of such sources.

▶ **Theorem 1.2** (generic upper bounds). *For $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and $k \geq 2\log_2 n + O(1)$, the following holds.*

- (public randomness version): *Suppose that the randomized communication complexity of f in the standard public-randomness model is C. Then, f can be computed by a protocol whose sole source of randomness is a public random string of length ℓ that has min-entropy k using $O(\ell - k + 1) \cdot C$ bits of communication.*
- (private randomness version): *Suppose that the randomized communication complexity of f in the standard private-randomness model is C. Then, f can be computed by a protocol whose sole source of randomness is provided by two independent random strings of length ℓ, each seen by one of the parties and having min-entropy k, using $2(\ell - k) + 3\log_2 n + O(C)$ bits of communication.*

The protocols use suitable methods of randomness extraction. Specifically, in the public-randomness case the two parties apply a seeded extractor to the only random string available to them, while using all possible seeds (of length $\log_2(\ell - k) + O(1)$). In the private-randomness case the parties apply a two-source extractor to the $2 \cdot (\ell - k + \log_2 n + O(1))$-bit long prefix of their sources, which requires them to only communicate this prefix.

Recall that in the case of perfect randomness, a common random source (i.e., public randomness) is preferable to private randomness, since the public randomness is known to both parties whereas uncorrelated private randomness require coordination (or communication). In contrast, in the context of defective randomness, two independent sources (even when each is only seen by one party) seem preferable to a single source of (defective) public randomness. We stress that our results only suggest that the communication complexity in the private (defective-randomness) case may be lower than in the public (defective-randomness) case, and establishing such a separation is left as an open problem. We mention that for some functions such separation does not exist (see Proposition 3.5).

We focus on the case of min-entropy that is at least logarithmic in the length of the input to the protocol (i.e., $k \geq \log_2 n$), because this is the minimal amount of perfect randomness that is required for constant-communication protocols for equality.[3] Still, one may study the case of sub-logarithmic min-entropy (and possibly integrate our results with those of [1]).

1.3 Remotely Related Works

Goldwasser, Sudan, and Vaikuntanathan [6] raised the general question of which distributed computing tasks that require randomness can be performed also when having access to defective sources of randomness.[4] Specifically, they showed that (Byzantine) agreement tasks fall into this category; that is, they can be performed quite well also in the case that each

[3] See [1, Thm. 3], which shows that computing the equality function when having access only to k bits of perfect public randomness requires communication complexity $\Omega(n/2^k)$.

[4] In a somewhat related vein, a body of work has investigated whether defective randomness suffices for cryptographic security in a variety of settings. McInnes and Pinkas [8] initiated this line of work by showing that information theoretic symmetric key cryptography is impossible without pure randomness. Dodis et al. [5] later extended this result to rule out the feasibility of a variety of cryptographic tasks from defective randomness, including computationally-secure symmetric key cryptography.

party has access to a (single) defective source of randomness. We stress that since the parties do not trust each other, the fact that their sources are independent of one another does not mean that they can extract almost perfect randomness by using some adequate extractor.

The following works that refer to different models of communication complexity are more related to the current study.

- Canonne et al. [2] considered a model that lies between the standard public and private randomness models (when the amount of randomness is sublogarithmic in the length of the inputs). Specifically, they considered two parties that are each given access to a private source of perfect randomness such that the two sources are tightly correlated (i.e., for a parameter $\rho \in [\pm 1]$, for each i, the i^{th} bit in the first source is ρ-correlated with the i^{th} bit in the second source). We mention that their motivation is not to study the usefulness of defective sources of randomness but rather to study the effect on uncertainty (about "contents") in communication complexity.
- Canetti and Goldreich [1] studied trade-offs between randomness and communication complexity. In particular, they showed that a logarithm amount of (perfect) randomness is sufficient for any communication protocol and that in some cases this upper bound is tight.
- Chor and Goldreich [3] studied the "distributional communication complexity" of functions when the protocol is only required to be correct with a specified probability $p > 1/2$, where the probability is taken over input pairs that are each chosen according to some distribution of specified min-entropy bound (i.e., min-entropy at least k). We stress that their study is fundamentally different from ours; they study the average-case (on inputs) behavior of protocols, where the inputs are drawn from a defective source of randomness, whereas we study the worst-case (on inputs) behavior of protocols that employ defective sources of randomness.

2 Preliminaries

We consider two-party randomized protocols for computing functions of the form $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, while using a defective source of randomness. Specifically, we consider sources of randomness that produce ℓ-bit long strings having min-entropy at least k; that is, each outcome occurs with probability at most 2^{-k}. Such sources are called (ℓ, k)-sources.

We consider both the public-randomness model in which the parties have access to common (public) randomness, and the private-randomness model in which each party has its private source of randomness, which is independent of the randomness of the other party. In our context (of defective random sources) it is important to stress that the postulated sources of randomness are the only ones available to the parties.

The results hold not only for "alternating protocols" (in which the parties alternatively exchange single bits), but directly for any protocol in which the sender of the next bit is determined by the communication so far; that is, no need to lose a factor of two in translation (from such general protocols to "alternating" ones).

2.1 Specific Background About Communication Complexity

We shall use the following basic result that refers to deterministic communication protocols.

▷ Claim 2.1 (the "corners lemma" (cf., e.g., [7, Prop. 1.13–1.14] or [10, Lem. 1.3–1.4])). Let Π' be a deterministic communication protocol and suppose that $\gamma \overset{\text{def}}{=} \Pi'(x_1, x_2) = \Pi'(y_1, y_2)$. Then, $\Pi'(x_1, y_2) = \Pi'(y_1, x_2) = \gamma$.

In addition, a basic notion of communication complexity that underlies many of its lower bound proofs is that of a fooling set, defdined as follows.

▶ **Definition 2.2** (fooling set (cf., e.g., [7, Sec. 1.3] or [10, Chap. 1])). *We say that $S \subseteq \{0,1\}^{n+n}$ is a* fooling set for *$f : \{0,1\}^{n+n} \to \{0,1\}$ if every f-monochromatic rectangle contains at most one point in S, where an f-*monochromatic rectangle *is a set $X \times Y$ such that $X, Y \subseteq \{0,1\}^n$ and f is constant on $X \times Y$* (i.e., $f(x,y) = f(x',y')$ for every $(x,y),(x',y') \in X \times Y$).

Note that a fooling set cannot contain two pre-images of $f^{-1}(0)$ (resp., $f^{-1}(1)$) that differ only on one coordinate; that is, if (x,y) and (x',y') are in a fooling set for f and $f(x,y) = f(x',y')$, then $x \neq x'$ and $y \neq y'$ (because two points that differ on a single coordinate constitute an f-monochromatic rectangle).

2.2 Specific Background About Randomness Extraction

As stated above, an (ℓ,k)-source is a distribution over ℓ-bit long strings having min-entropy at least k, where the min-entropy of a random variable X is $\min_{v \in \text{Supp}(X)}\{\log_2(1/\Pr[X=v])\}$. That is, X has min-entropy k if and only if for every v it holds that $\Pr[X=v] \leq 2^{-k}$.

We say that $\text{EXT} : \{0,1\}^d \times \{0,1\}^\ell \to \{0,1\}^m$ is a (seeded) (k,ϵ)-extractor if for every random variable X of min-entropy k the total variation distance between $\text{EXT}(U_d, X)$ and U_m is at most ϵ, where U_n denotes the uniform distribution on $\{0,1\}^n$. In this case ϵ is called the error of EXT, and d is its seed length.

We say that $\text{EXT} : \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^m$ is a two-source extractor for independent (ℓ,k)-sources if for every two independent random variables X and Y of min-entropy k the total variation distance between $\text{EXT}(X,Y)$ and U_m is at most ϵ, called its error. This definition is readily extended to independent sources of parameters (ℓ_1,k_1) and (ℓ_2,k_2) respectively.

3 The Public-Randomness Model

For a protocol Π in the public-randomness model, we denote by $\Pi(x,y;r)$ the transcript of the communication on input $(x,y) \in \{0,1\}^{n+n}$ and randomness $r \in \{0,1\}^\ell$. The output of such a protocol is determined by its transcript (e.g., it may be its last bit), and is denoted $\overline{\Pi}(x,y;r)$.

▶ **Definition 3.1** (communication complexity with a weak public source). *An (ℓ,k)-*public-randomness protocol for computing a function *$f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is a protocol that satisfies* $\Pr[\overline{\Pi}(x,y;\Xi) = f(x,y)] \geq 2/3$*, for every $(x,y) \in \{0,1\}^{n+n}$ and every (ℓ,k)-source Ξ.*

▶ **Theorem 3.2** (a general lower bound). *Suppose that $f : \{0,1\}^{n+n} \to \{0,1\}$ has a fooling get of size 2^m. Then, any (ℓ,k)-public-randomness protocol for computing f has communication complexity at least* $\min(m-1, \ell-k-1)/2$.

Proof. Suppose that f has a (ℓ,k)-public-randomness protocol, denoted Π, of communication complexity $t \leq (n-1)/2$. We first observe that there exists a dense set of possible source-outcomes R and two input pairs (x_1,y_1) and (x_2,y_2) that reside in the fooling set such that Π in constant on all triples (x_i,y_i,r), where $r \in R$ and $i \in \{1,2\}$. The theorem will follow by using the standard "corners lemma" (in a non-standard way) and defining a source that is uniform over R. Details follow.

The following technical claim has nothing to do with communication complexity; it holds for any function $F : [2^m] \times \{0,1\}^\ell \to \{0,1\}^t$, where in the current case $[2^m]$ represents the indices of the strings in the fooling set (for f), $\{0,1\}^\ell$ represents possible outcomes of the public source, and $\{0,1\}^t$ represents possible transcripts of Π.

▷ Claim 3.2.1 (a simple combinatorial claim). Let $F : [2^m] \times \{0,1\}^\ell \to \{0,1\}^t$. Then, for any $S = \{(x_1, y_1), ..., (x_{2^m}, y_{2^m})\}$ and $t \leq (m-1)/2$, there exist distinct $i, j \in [2^m]$, a string $\gamma \in \{0,1\}^t$, and a set $R \subseteq \{0,1\}^\ell$ of density at least 2^{-2t-1} such that for every $r \in R$ it holds that $F(i,r) = F(j,r) = \gamma$.

We will apply Claim 3.2.1 to the hitting set S and to the function $F(i,r) \stackrel{\text{def}}{=} \Pi(x_i, y_i; r)$. But let us prove the claim first.

Proof. A simple counting implies that, for every $i \in [2^m]$, there exist $\gamma_i \in \{0,1\}^t$ and a set $R_i \subseteq \{0,1\}^\ell$ of density 2^{-t} such that for every $r \in R_i$ it holds that $F(i,r) = \gamma_i$. Similarly, there exist $\gamma \in \{0,1\}^t$ and $G \subseteq [2^m]$ of density 2^{-t} such that $\gamma_i = \gamma$ for every $i \in G$.

The key observation is that if $t \leq (n-1)/2$, then there exist distinct $i, j \in G$ such that $|R_i \cap R_j| \geq 2^{\ell-2t-1}$. This is shown by fixing an arbitrary $G' \subseteq G$ of size 2^{t+1}, which is possible since $2^{t+1} \leq 2^{n-t}$, and assuming towards the contradiction that, for every distinct $i, j \in G'$, it holds that $|R_i \cap R_i| < 2^{\ell-2t-1}$. Then, we get

$$\begin{aligned}
\left|\sum_{i\in G'} R_i\right| &\geq \sum_{i\in G'} |R_i| - \sum_{i\neq j\in G'} |R_i \cap R_j| \\
&> 2^{t+1} \cdot 2^{\ell-t} - \binom{2^{t+1}}{2} \cdot 2^{\ell-2t-1} \\
&> 2^{\ell+1} - 2^{2t+1} \cdot 2^{\ell-2t-1} \\
&= 2^\ell
\end{aligned}$$

which is impossible. The claim follows by fixing $i \neq j$ such that $|R_i \cap R_j| \geq 2^{\ell-2t-1}$, and defining $R = R_i \cap R_j$. ◁

Applying Claim 3.2.1 to the hitting set $S = \{(x_1, y_1), ..., (x_{2^m}, y_{2^m})\}$ of the hypothesis, while letting $F(i,r) \stackrel{\text{def}}{=} \Pi(x_i, y_i; r)$ and using $t \leq (m-1)/2$, we infer that the fooling set contains two points (x_i, y_i) and (x_j, y_j) such that $\Pi(x_i, y_i; r) = \Pi(x_j, y_j; r) = \gamma$ holds for any $r \in R$, where $R \subseteq \{0,1\}^\ell$ has density at least 2^{-2t-1}.

Next, applying the "corners lemma" (i.e., Claim 2.1), we infer that $\Pi(x_i, y_i; r) = \Pi(x_i, y_j; r) = \Pi(x_i, y_j; r) = \Pi(x_j, y_j; r)$ for every $r \in R$. Note that this application of the "corners lemma" refers to the residual deterministic protocols $\Pi'_r(x,y) = \Pi(x,y;r)$, for all $r \in R$, and it implies that $\Pi'_r(x_i, y_j) = \Pi'_r(x_j, y_i) = \gamma$ for each $r \in R$.

Lastly, picking an $(\ell, \ell-2t-1)$-source that is uniform on R, we infer that, when fed with randomness from this source, the execution of Π does not distinguish these four input-pairs (i.e., (x_i, y_i), (x_i, y_j), (x_i, y_j) and (x_j, y_j)). On the other hand, by hypothesis that (x_i, y_i) and (x_j, y_j) belong to a fooling set, these four input-pairs cannot have the same f-value (i.e., it cannot be that $f(x_i, y_i) = f(x_i, y_j) = f(x_j, y_i) = f(x_j, y_j)$, since this would mean that (x_i, y_i) and (x_j, y_j) reside in the f-monochromatic rectangle $\{x_i, x_j\} \times \{y_i, y_j\}$). Hence, the hypothesis that Π is a (ℓ, k)-public-randomness protocol for f implies that the foregoing source has min-entropy below k; that is, $\ell - 2t - 1 < k$. The theorem follows, since we established $t > (\ell - k - 1)/2$, under the hypothesis $t \leq (m-1)/2$. ◀

An archetypical corollary

Recalling that equality has a fooling set of size 2^n and applying Theorem 3.2, we get

▶ **Corollary 3.3** (lower bound for equality). *Any (ℓ, k)-public-randomness protocol for computing equality of n-bit strings has communication complexity at least* $\min(n-1, \ell-k-1)/2$.

This lower bound is tight up to a constant factor, since equality has a constant communication protocol in the standard public-randomness model and the following generic result definitely applies to it.

▶ **Theorem 3.4** (a generic upper bound). *Suppose that $f : \{0,1\}^{n+n} \to \{0,1\}$ has communication complexity $\mathcal{C}^{\mathsf{pub}}(f)$ in the standard public-randomness model. Then, for every $k \leq \ell$ such that $k > \log_2 n + O(1)$, there exists an (ℓ, k)-public-randomness protocol for computing f with communication complexity $O(\ell - k) \cdot \mathcal{C}^{\mathsf{pub}}(f)$.*

Recall that equality has constant communication complexity in the standard public-randomness model.

Proof. Recall that the *randomness complexity* of any protocol for computing f can be reduced to $m \stackrel{\text{def}}{=} \log_2 n + O(1)$ (while possibly increasing its communication complexity by a constant factor).[5] The key observation is that the parties can *emulate the extraction* of m almost-random bits from the public (ℓ, k)-source, by trying all possible seeds for an adequate randomness extractor, and use the extracted bit to emulate the original randomized protocol. Specifically, for $k \geq m$, such extraction is possible using a (perfectly random) seed of length $d \stackrel{\text{def}}{=} \log(\ell - k) + O(1)$ (see, e.g., [11, Sec. 3.1]). Hence, the parties can emulate the randomized protocol by invoking it 2^d times using as randomness the "extracted outputs" under all possible seeds. Details follow.

Let $\text{EXT}(s, r)$ denote the output of the extractor $\text{EXT} : \{0,1\}^d \times \{0,1\}^\ell \to \{0,1\}^m$ on seed s and source outcome r. Then, given (defective) public-randomness $r \in \{0,1\}^\ell$, the parties emulate 2^d invocations of the standard randomized protocol such that in the i^{th} invocation they use public-randomness $\text{EXT}(i, r)$, where $i \in [2^d] \equiv \{0,1\}^d$, and rule by majority. Actually, we use a randomized protocol for the standard model that has error probability at most 0.1 (rather than at most 1/3), which can be obtained by a constant number of repetitions.

We claim that if EXT has error 0.05 on any (ℓ, k)-source R, then, for every fixed input pair, with probability at least 2/3 over the outcome of R, the majority of the extracted values (over all possible seeds) yield protocol executions with the correct output. This is the case because otherwise the statistical difference between $\text{EXT}(U_d, R)$ and U_m is at least $\frac{1}{3} \cdot \frac{1}{2} - 0.1 > 0.05$, where the first (resp., second) term represents a lower bound (resp., upper bound) on the probability that the protocol yields a wrong answer when run with randomness $\text{EXT}(U_d, R)$ (resp., U_m). This yields an (ℓ, k)-public-randomness protocol of communication complexity $2^d \cdot O(\mathcal{C}^{\mathsf{pub}}(f)) = O(\ell - k) \cdot \mathcal{C}^{\mathsf{pub}}(f)$. ◀

[5] See, e.g., [1, Thm. 5] and [9]. The basic argument leaves the communication complexity intact, but increases the error probability by an arbitrary small constant, where this constant effects the additive constant in m. To regain the original error bound, three repetitions suffice.

On the gap between the lower and upper bound

The bounds provided by Theorems 3.3 and 3.4 leave a gap of a factor $\Theta(\mathcal{C}^{\texttt{pub}}(f))$ in the non-trivial case (i.e., $\Omega(\ell-k)$) versus $O(\ell-k)\cdot\mathcal{C}^{\texttt{pub}}(f)$). The following example implies that the gap cannot be closed by increasing the lower bound.

▶ **Proposition 3.5** (improved upper bound). *For every $m<n$, there exists a function $f:\{0,1\}^{n+n}\to\{0,1\}$ that satisfies the following two conditions:*

1. *The function f has communication complexity $\mathcal{C}^{\texttt{pub}}(f)=\Theta(m)$ in the standard public-randomness model;*
2. *For every $k\le\ell$ such that $k>\log_2 n+O(1)$, there exists an (ℓ,k)-public-randomness protocol for computing f with communication complexity $O(\ell-k)+O(\mathcal{C}^{\texttt{pub}}(f))$.*

Proof. Consider the function $f(x'x'',y'y'')=\texttt{EQ}(x',y')\oplus\texttt{IP}_2(x'',y'')$, where $|x''|=m=n-|x'|$, $\texttt{EQ}$ denotes the equality function, and $\texttt{IP}_2$ denotes inner-product mod 2. Then, $\mathcal{C}^{\texttt{pub}}(f)\ge\mathcal{C}^{\texttt{pub}}(\texttt{IP}_2)=\Omega(m)$, where the first inequality follows by a straightforward reduction and the lower bound is proved in [3]. We obtain an (ℓ,k)-public-randomness protocol for computing f with communication complexity $O(\ell-k)\cdot\mathcal{C}^{\texttt{pub}}(\texttt{EQ})+m+1=O(\ell-k)+O(\mathcal{C}^{\texttt{pub}}(f))$, by combining the generic protocol for $\texttt{EQ}$ (see Theorem 3.4) with the straightforward deterministic protocol for $\texttt{IP}_2$. ◀

4 The Private-Randomness Model

For a protocol Π in the private-randomness model, we denote by $\Pi((x,r),(y,s))$ the transcript of the communication on input $(x,y)\in\{0,1\}^{n+n}$ with private randomness $r,s\in\{0,1\}^{\ell}$; that is, the first (resp., second) party gets input x (resp., y) and private randomness r (resp., s). The output of such a protocol is determined by its transcript (e.g., it may be its last bit), and is denoted $\overline{\Pi}((x,r),(y,s))$.

▶ **Definition 4.1** (communication complexity with weak private sources). *An (ℓ,k)-*private-randomness protocol for computing a function $f:\{0,1\}^n\times\{0,1\}^n\to\{0,1\}$ *is a protocol that satisfies $\Pr[\overline{\Pi}((x,\Xi'),(y,\Xi''))=f(x,y)]\ge 2/3$, for every $(x,y)\in\{0,1\}^{n+n}$ and every pair of independent (ℓ,k)-sources Ξ' and Ξ''.*

▶ **Theorem 4.2** (a general lower bound). *Suppose that $f:\{0,1\}^{n+n}\to\{0,1\}$ has a fooling get of size 2^m. Then, any (ℓ,k)-private-randomness protocol for computing f has communication complexity at least $\min(m-1,\ell-k-1)/2$.*

Proof. The proof is analogous to the proof of Theorem 3.2. We start with a hypothetical (ℓ,k)-private-randomness protocol, denoted Π, that computes f with communication complexity $t\le(n-1)/2$. Then, we apply Claim 3.2.1 to the (somewhat less natural) function $F:[2^m]\times\{0,1\}^{\ell}\to\{0,1\}^t$ defined by $F(i,r)\overset{\text{def}}{=}\Pi((x_i,r),(y_i,r))$, where $S=\{(x_1,y_1),...,(x_{2^m},y_{2^m})\}$ is a fooling set for f. Hence, we infer that *there exist distinct $i,j\in[2^m]$, a string $\gamma\in\{0,1\}^t$, and a set $R\subseteq\{0,1\}^{\ell}$ of density at least 2^{-2t-1} such that for every $r\in R$ it holds that* $\Pi((x_i,r),(y_i,r))=\Pi((x_j,r),(y_j,r))=\gamma$.

Now, applying Claim 2.1 thrice, we infer that $\Pi((x_a,r),(y_b,s))=\gamma$ for every $r,s\in R$ and $a,b\in\{i,j\}$. Specifically, for both $a\in\{i,j\}$ and every $r,s\in R$, considering the residual protocol $\Pi'_a(r,s)=\Pi((x_a,r),(y_a,s))$ and using $\Pi((x_a,r),(y_a,r))=\Pi((x_a,s),(y_a,s))=\gamma$, we infer that $\Pi((x_a,r),(y_a,s))=\gamma$. Hence, $\Pi((x_i,r),(y_i,s))=\gamma=\Pi((x_j,r),(y_j,s))$. Now, considering the residual protocol $\Pi'_{r,s}(x,y)=\Pi((x,r),(y,s))$ and using $\Pi((x_i,r),(y_i,s))=\Pi((x_j,r),(y_j,s))$, we get that $\Pi((x_i,r),(y_j,s))=\gamma=\Pi((x_j,r),(y_i,s))$.

Picking a pair of independent $(\ell, \ell - 2t - 1)$-sources that are each uniform on R, we infer that the execution of Π does not distinguish the four input-pairs (x_i, y_i), (x_i, y_j), (x_i, y_j) and (x_j, y_j). On the other hand, by hypothesis that (x_i, y_i) and (x_j, y_j) belong to a fooling set, and so these four input-pairs cannot have the same f-value. Hence, the hypothesis that Π is a (ℓ, k)-private-randomness protocol for f implies that $\ell - 2t - 1 < k$. The theorem follows, since we established $t > (\ell - k - 1)/2$, under the hypothesis $t \le (m - 1)/2$. ◀

▶ **Theorem 4.3** (a generic upper bound). *Suppose that $f : \{0,1\}^{n+n} \to \{0,1\}$ has communication complexity $\mathcal{C}^{\mathtt{priv}}(f)$ in the standard private-randomness model. Then, for every $k \le \ell$ such that $k > 2\log_2 n + 2\log_2 \ell + O(1)$, there exists an (ℓ, k)-private-randomness protocol for computing f with communication complexity* $\min(2(\ell - k) + 3\log_2 n + O(\mathcal{C}^{\mathtt{priv}}(f)), \ell + \log_2 n + O(\mathcal{C}^{\mathtt{priv}}(f))))$.

Proof. The bounds follow by having one party send a $\min(2\cdot(\ell - k + \log_2 n + O(1)), \ell)$-bit long prefix of its private randomness to the second party, who extracts almost perfect randomness from the two outcomes (using a two-source extractor), sends one half of it back, and then both parties execute the standard protocol. Details follow.

First, recall that the randomness complexity of any protocol for computing f can be reduced to $m \overset{\text{def}}{=} \log_2 n + O(1)$ (while possibly increasing its communication complexity by a constant factor). Second, recall that a seedless (two-source) randomness extractor can extract $2m$ almost random bits from an (ℓ, k)-source and an independent (ℓ', k')-source, provided that $2m \le k + k' - \max(\ell, \ell') - O(1)$ (see [3, Thm. 7(2)]).[6] Now, if $\ell' \overset{\text{def}}{=} 2\cdot(\ell - k + \log_2 n) + O(1) \le \ell$, then an ℓ'-bit prefix of an (ℓ, k)-source has min-entropy $k' \overset{\text{def}}{=} \ell' - (\ell - k) = (\ell - k) + 2\log_2 n + O(1)$, and so $k + k' - \max(\ell, \ell') - O(1) = 2\log_2 n + O(1)$. Hence, sending the prefix of the first source sent to the second party, allows it to extract $2\log_2 n + O(1)$ bits that are almost random. Sending half of these bits to the first party allows the two parties to emulate the original protocol. The communication complexity of the proposed protocol is at most $\ell' + \log_2 n + O(1) + O(\mathcal{C}^{\mathtt{priv}}(f))$, which equals $2(\ell - k) + 3\log_2 n + O(\mathcal{C}^{\mathtt{priv}}(f))$.

As for the case of $\ell' > \ell$, recall that a seedless (two-source) randomness extractor can extract $2m$ almost random bits from a pair of independent (ℓ, k)-source, provided that $2m \le k - 2\log_2 \ell - O(1)$ (see [3, Thm. 7(1)]). In this case, sending the outcome of the first source to the second party allows for the foregoing emulation, at a total communication cost of $\ell + \log_2 n + O(\mathcal{C}^{\mathtt{priv}}(f))$. ◀

References

1 Ran Canetti and Oded Goldreich. Bounds on tradeoffs between randomness and communication complexity. *Comput. Complex.*, 3:141–167, 1993.

2 Clément L. Canonne, Venkatesan Guruswami, Raghu Meka, and Madhu Sudan. Communication with imperfectly shared randomness. *IEEE Trans. Inf. Theory*, 63(10):6799–6818, 2017.

3 Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.

4 Martin Dietzfelbinger, Juraj Hromkovic, and Georg Schnitger. A comparison of two lower-bound methods for communication complexity. *Theor. Comput. Sci.*, 168(1):39–51, 1996.

5 Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *FOCS*, pages 196–205. IEEE Computer Society, 2004.

[6] Note that for every $d \ge 0$, a (ℓ, k)-source may be viewed as an $(\ell + d, k)$-source.

6 Shafi Goldwasser, Madhu Sudan, and Vinod Vaikuntanathan. Distributed computing with imperfect randomness. In *DISC*, volume 3724 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2005.

7 Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

8 James L. McInnes and Benny Pinkas. On the impossibility of private key cryptography with weakly random keys. In *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 421–435. Springer, 1990.

9 Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

10 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications.* Cambridge University Press, 2020.

11 Ronen Shaltiel. An introduction to randomness extractors. In *ICALP (2)*, volume 6756 of *Lecture Notes in Computer Science*, pages 21–41. Springer, 2011.

On the Cut Dimension of a Graph

Troy Lee ✉
Centre for Quantum Software and Information, University of Technology Sydney, Australia

Tongyang Li ✉
Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA, USA
Center on Frontiers of Computing Studies, Peking University, Beijing, China

Miklos Santha ✉
CNRS, IRIF, Université de Paris, France
Centre for Quantum Technologies and MajuLab, National University of Singapore, Singapore

Shengyu Zhang ✉
Tencent Quantum Laboratory, Shenzhen, China

Abstract

Let $G = (V, w)$ be a weighted undirected graph with m edges. The cut dimension of G is the dimension of the span of the characteristic vectors of the minimum cuts of G, viewed as vectors in $\{0,1\}^m$. For every $n \geq 2$ we show that the cut dimension of an n-vertex graph is at most $2n-3$, and construct graphs realizing this bound.

The cut dimension was recently defined by Graur et al. [13], who show that the maximum cut dimension of an n-vertex graph is a lower bound on the number of cut queries needed by a deterministic algorithm to solve the minimum cut problem on n-vertex graphs. For every $n \geq 2$, Graur et al. exhibit a graph on n vertices with cut dimension at least $3n/2 - 2$, giving the first lower bound larger than n on the deterministic cut query complexity of computing mincut. We observe that the cut dimension is even a lower bound on the number of *linear* queries needed by a deterministic algorithm to solve mincut, where a linear query can ask any vector $x \in \mathbb{R}^{\binom{n}{2}}$ and receives the answer $w^T x$. Our results thus show a lower bound of $2n-3$ on the number of linear queries needed by a deterministic algorithm to solve minimum cut on n-vertex graphs, and imply that one cannot show a lower bound larger than this via the cut dimension.

We further introduce a generalization of the cut dimension which we call the ℓ_1-approximate cut dimension. The ℓ_1-approximate cut dimension is also a lower bound on the number of linear queries needed by a deterministic algorithm to compute minimum cut. It is always at least as large as the cut dimension, and we construct an infinite family of graphs on $n = 3k+1$ vertices with ℓ_1-approximate cut dimension $2n-2$, showing that it can be strictly larger than the cut dimension.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Theory of computation → Graph algorithms analysis; Theory of computation → Computational complexity and cryptography

Keywords and phrases Query complexity, submodular function minimization, cut dimension

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.15

Related Version *Previous Version*: `https://arxiv.org/abs/2011.05085`

Funding Research at CQT is funded by the National Research Foundation, the Prime Minister's Office, and the Ministry of Education, Singapore under the Research Centres of Excellence programme's research grant R-710-000-012-135. In addition, this work has been supported in part by the QuantERA ERA-NET Cofund project QuantAlgo and the ANR project ANR-18-CE47-0010 QUDATA.
Troy Lee: Troy Lee is supported in part by the Australian Research Council Grant No: DP200100950.
Tongyang Li: Tongyang Li is supported by the ARO contract W911NF-17-1-0433, NSF grant PHY-1818914, and an NSF QISE-NET Triplet Award (grant DMR-1747426).

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 15; pp. 15:1–15:35
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Let $G = (V, w)$ be a weighted undirected n-vertex graph where w is an $\binom{n}{2}$-dimensional nonnegative real vector assigning a (possibly zero) weight to each edge slot. For a nontrivial subset $\emptyset \neq X \subsetneq V$, let $\Delta(X)$ be the set of edges of G with one endpoint in X and one endpoint in $\bar{X} = V \setminus X$. A *cut* S in G is a subset of edges of the form $\Delta(X)$ for a nontrivial set X. The sets X and $\bar{X}$ are called the *shores* of the cut. For a cut S, its *weight* is the sum of the weights of the edges in S, denoted $w(S)$. The minimum cut problem is to find the minimum of $w(S)$ over all cuts S. The study of algorithms for the minimum cut problem in theoretical computer science goes back at least to the 1960's and has given rise to a vast and beautiful literature. Minimum cut is also a problem of great practical importance with applications to, for example, clustering algorithms and evaluating network reliability. Randomized algorithms can solve the minimum cut problem in nearly linear time: in 1996 Karger gave an algorithm with running time $O(m \log^3(n))$ to compute the minimum cut of a weighted graph with m edges [20]. This was the best known bound until very recently when two independent works improved on it. Gawrychowski, Mozes, and Weimann [8] gave a randomized algorithm with running time $O(m \log^2(n))$ [8] and Mukhopadhyay and Nanongkai [24] gave a randomized algorithm with time complexity $O(m \frac{\log^2(n)}{\log \log n} + n \log^6(n))$. Gawrychowski, Mozes, and Weimann [9] later improved the running time of the Mukhopadhyay and Nanongkai algorithm to $O(m \frac{\log^2(n)}{\log \log n} + n \log^{3+\varepsilon}(n))$.

For *simple* graphs G, randomized algorithms are known with running times $O(m \log(n))$ and $O(m + n \log^3(n))$ [10]. For simple graphs even nearly linear time *deterministic* algorithms are known. Kawarabayashi and Thorup gave an $O(m \log^{12}(n))$ time algorithm [21], which was subsequently improved to $O(m(\log(n) \log \log n)^2)$ by Henzinger, Rao, and Wang [16].

Our work spans two aspects of the study of the minimum cut problem. The first is to query complexity lower bounds on minimum cut. A natural model in which to study the query complexity of minimum cut is for algorithms allowed to make *cut queries*. A cut query algorithm can query any subset $\emptyset \neq X \subsetneq V$ and receives the answer $w(\Delta(X))$. One motivation to study cut query algorithms comes from submodular function minimization. The cut function $f(X) = w(\Delta(X))$ is a submodular function, and finding the minimum cut value is equivalent to finding the minimum value of f over all nontrivial sets X. The problem of minimizing a submodular function is often studied with respect to an evaluation oracle, which in the case of the cut function is exactly a cut query.

Harvey [15] observed that results on the deterministic communication complexity of deciding graph connectivity [14] imply that any deterministic cut query algorithm to compute minimum cut, or even to decide if the graph is connected or not, must make at least cn cut queries, for a constant $c < 1$. Analogous results on the randomized communication complexity of connectivity [2] imply an $\Omega(n/\log(n))$ lower bound on the number of cut queries needed by a randomized algorithm to compute minimum cut (or even connectivity).

On the algorithms side, Rubinstein, Shramm, and Weinberg [26] gave a randomized algorithm computing the minimum cut of a simple graph with $\tilde{O}(n)$ many cut queries. Recently, Mukhopadhyay and Nanongkai [24] used a different approach based on Karger's 2-respecting tree algorithm [20] to also give a randomized $\tilde{O}(n)$ cut query algorithm to compute minimum cut in a general undirected weighted graph.

For deterministic cut query algorithms, there remains a large gap between the best upper and lower bounds. We are not aware of any deterministic algorithm for minimum cut better than learning the entire graph, which can take $\Omega(n^2/\log(n))$ cut queries in the worst case. On the lower bound side, Graur, Pollner, Ramaswamy, and Weinberg [13] recently introduced a very interesting lower bound technique called the *cut dimension*, which we now describe. Let $G = (V, w)$ be a weighted undirected graph with n vertices and m edges, and let $\mathcal{M}(G)$ be the set of minimum cuts of G. For a cut $S \in \mathcal{M}(G)$, let $\chi(S) \in \{0,1\}^m$ be the characteristic vector of S amongst the m edges of G. Let $\vec{\mathcal{M}}(G) = \{\chi(S) : S \in \mathcal{M}(G)\}$. The cut dimension of G, denoted $\mathrm{cdim}(G)$, is the dimension of $\mathrm{span}(\vec{\mathcal{M}}(G))$. It is shown in [13] that for any n-vertex graph G, the cut dimension $\mathrm{cdim}(G)$ is a lower bound on the deterministic cut query complexity of computing minimum cut on weighted n-vertex graphs. Moreover, for every $n \geq 2$ they construct an n-vertex graph G with cut dimension $3n/2 - 2$.

Besides showing lower bounds on cut query complexity, the cut dimension is a natural measure of the complexity of mincuts in a graph. There is a rich literature on the possible structure of mincuts in a graph. Perhaps the first result of this kind is the cactus representation of mincuts by [6]. A cactus for a graph G is a sparse weighted graph C that represents all the mincuts of G. One consequence of the cactus representation is that the number of possible mincuts in an n-vertex weighted graph is at most $\binom{n}{2}$. This upper bound was later given an algorithmic proof via Karger's famous contraction algorithm ([19], Theorem 6.1). The n-vertex cycle graph has $\binom{n}{2}$ many minimum cuts and shows that this bound can be tight.

While the cycle has $\binom{n}{2}$ many mincuts, these cuts live in an n-dimensional space as the n-vertex cycle only has n edges. Is it possible to construct graphs with many cuts that also have high cut dimension? We show that this is not possible, and in fact the cut dimension of an n-vertex graph is at most $2n - 3$.

▶ **Theorem 1** (Main Upper Bound). *For any weighted undirected graph G on $n \geq 2$ vertices it holds that* $\mathrm{cdim}(G) \leq 2n - 3$.

Like the cactus representation, this shows another aspect in which the mincuts of a graph are constrained to have a relatively simple structure. We further show that this bound is tight by constructing graphs with cut dimension $2n - 3$ for every $n \geq 2$.

▶ **Theorem 2** (Main Lower Bound). *For every $n \geq 2$ there exists an n-vertex weighted undirected graph G with* $\mathrm{cdim}(G) = 2n - 3$.

In addition to shedding further light on the structure of minimum cuts, this improves the best known lower bound on the deterministic cut query complexity of the minimum cut problem to $2n - 3$. We additionally show that the cut dimension is even a lower bound on a stronger query model called the linear query model, recently studied in [1]. In the linear query model, the algorithm can query any vector $x \in \mathbb{R}^{\binom{n}{2}}$ and receives the answer $\langle w, x \rangle$, the inner product of w and x. Linear queries can be much more powerful than cut queries as one can completely learn an unweighted graph with a single linear query. By an information theoretic argument learning an unweighted graph can require $\Omega(n^2/\log(n))$ many cut queries since each cut query reveals at most $O(\log(n))$ bits.

We further introduce a lower bound technique which is a generalization of the cut dimension that we call the ℓ_1-approximate cut dimension. This technique looks not just at mincuts in the graph, but all cuts. We again look at the span of the dimension of these cuts with an additional twist. Suppose the weight of a minimum cut in G is λ and cut S has $w(S) = \lambda + \delta$. Abusing notation we will let S represent both a set of edges and the characteristic vector $S \in \{0,1\}^{\binom{n}{2}}$ of S among all edge slots. The vector S can be *perturbed* to $S - u$ for any vector $u \geq 0$ with $\|u\|_{1,w} \leq \delta$. Here $\|u\|_{1,w} = \sum_i |w(i) \cdot u(i)|$ is the ℓ_1 norm of u weighted by the edge weights of the graph. The ℓ_1-approximate cut dimension of G is then the minimum over all valid perturbations of the dimension of the span of the perturbed cut vectors.

The minimization over all perturbations makes the ℓ_1-approximate cut dimension a difficult quantity to lower bound. We are able to show, however, that the ℓ_1-approximate cut dimension can be strictly larger than the cut dimension. For every $k \in \mathbb{N}$ and $n = 3k+1$, we construct an unweighted n-vertex graph G whose ℓ_1-approximate cut dimension is $2n-2$. This has the following application.

▶ **Theorem 3.** *Any deterministic linear query algorithm that correctly computes the minimum cut of all n-vertex weighted undirected graphs must make at least $2n-2$ queries in the worst case.*

Computing the minimum cut of a graph with cut queries is a special case of finding the nontrivial minimum of a symmetric submodular function $f : 2^V \to \mathbb{R}$ with evaluation queries. That is, to find $\min_{X:\emptyset \neq X \subsetneq V} f(S)$ for a submodular f that satisfies $f(X) = f(V \setminus X)$ for all $X \subseteq V$. As linear queries are more powerful than cut queries, Theorem 3 also implies a $2n-2$ evaluation query lower bound for a deterministic algorithm finding a nontrivial minimum of a symmetric submodular function, which is currently the best known.

1.1 Techniques

We give two different proofs of the $2n-3$ upper bound on the cut dimension and two different techniques to create graphs with cut dimension $2n-3$. The first proof is direct and uses the *combinatorial uncrossing* technique, and in particular a key lemma of Jain [18] in his factor of 2 approximation algorithm for the survivable network design problem. The second proof is by induction and follows a framework for constructing a cactus representation of the mincuts of a graph [6, 7]. The second proof uses very few properties of mincuts and seems better suited to also upper bound the ℓ_1-approximate cut dimension, one of our main open questions.

Key to both proofs is the concept of when cuts *cross* each other. Two cuts $\Delta(X), \Delta(Y)$ are said to *cross* if all four of the intersections $X \cap Y, \bar{X} \cap Y, X \cap \bar{Y}, \bar{X} \cap \bar{Y}$ are non-empty. Note that in the definition of crossing it does not matter which shore we take to define the cut, thus crossing is a property of the cuts themselves. A family $\mathcal{L}$ of cuts is called *cross-free* if for all cuts $S, T \in \mathcal{L}$ it holds that S and T do not cross.

First upper and lower bound proof. In the first upper bound proof, we first show that any cross-free family of cuts has cardinality at most $2n-3$ (see Section 4.1). We then use Jain's lemma [18] (stated in Lemma 19) to conclude that for a maximal cross-free subset $\mathcal{L} \subseteq \mathcal{M}(G)$ it holds that $\vec{\mathcal{L}} = \{\chi(S) : S \in \mathcal{L}\}$ spans the set $\vec{\mathcal{M}}(G)$. This shows that the cut dimension of a graph is at most $2n-3$.

In the first lower bound proof we use a tree-representation of a cross-free family of cuts to show that in a complete graph the cut vectors of a cross-free family of cuts are linearly independent (Lemma 27). Thus the lower bound reduces to constructing a graph whose minimum cuts are a cross-free family of cuts of size $2n-3$. Such a construction has already been given by Chandra and Ram [5]. We go a step further, however. For any $\mathcal{L}$ which is a cross-free family of cuts from a complete n-vertex graph with $|\mathcal{L}| = 2n-3$, in Theorem 31 we explicitly give the edge weights of a complete weighted graph G such that $\mathcal{M}(G) = \mathcal{L}$ and therefore $\mathrm{cdim}(G) = 2n-3$. This task is made easier by Lemma 29, which states that if $\mathcal{L}$ is a cross-free family of cuts of size $2n-3$ that all have the same weight, then this must be the weight of a minimum cut of the graph. This lemma is again shown by the combinatorial uncrossing technique. This reduces the construction problem to solving the linear program of finding a positive vector w that makes all cuts in $\mathcal{L}$ have the same weight. We explicitly give a solution to this linear program by viewing it as a flow problem on the tree-representation of $\mathcal{L}$.

Second upper and lower bound proof. The second upper bound proof is by induction and follows methods to construct a cactus representation of mincuts [6, 7]. In the base case $n = 2$ it is easy to see that the cut dimension is at most $2n - 3 = 1$. For the inductive step, when G is an $n > 2$ vertex graph, there are 3 cases to consider. We call a cut of the form $\Delta(\{v\})$ a *star cut*, and we will refer to all other cuts as *non-star* cuts. The first case is where all cuts in $\mathcal{M}(G)$ are star cuts. As the graph has n vertices there are at most n star cuts and so in this case the cut dimension is at most $n \leq 2n - 3$. The second case is where for every non-star cut $S \in \mathcal{M}(G)$ there is a cut $T \in \mathcal{M}(G)$ which crosses S. In this case [6] show that the graph must be a cycle and the cut dimension is again at most $n \leq 2n - 3$.

The interesting case is where there is a non-star cut $\Delta(V_0) \in \mathcal{M}(G)$ which is not crossed by any other cut in $\mathcal{M}(G)$. Let $V_1 = \bar{V}_0$. In this case we use a decomposition of G along the cut $\Delta(V_0)$, that we call the *separation* of G, into two smaller graphs G_b, for $b \in \{0, 1\}$. The graph G_b is formed from G by contracting V_{1-b} into a single new vertex v_{1-b}. We show that $\mathrm{cdim}(G) \leq \mathrm{cdim}(G_0) + \mathrm{cdim}(G_1) - 1$ which implies immediately the upper bound. Indeed, let $k = |V_0| \geq 2$. Then G_0 is a graph on $k + 1$ vertices and G_1 is a graph on $n - k + 1$ vertices, both of which are less than n. The inductive hypothesis therefore gives $\mathrm{cdim}(G) \leq 2(k + 1) - 3 + 2(n - k + 1) - 3 - 1 = 2n - 3$.

For the second lower bound proof we use the *merge* operation which creates from two graphs G_b, for $b \in \{0, 1\}$, and a specified vertex v_{1-b} from each, a composed graph G where the vertices v_0, v_1 are not present but the cut $\Delta(V_0)$ reflects the structure of the star cuts at v_0 and v_1 in the original graphs. The operations separation and merge are inverses in the sense that if we apply merge to $\{G_0, G_1\}$ followed by separation on the resulting graph G, we receive back $\{G_0, G_1\}$. We also show that the inequality $\mathrm{cdim}(G) \leq \mathrm{cdim}(G_0) + \mathrm{cdim}(G_1) - 1$ holds with equality if $\Delta(V_0)$ is a connected graph. This enables us to construct inductively a sequence of graphs $G^{(n)}$ on n vertices whose cut dimension is $2n-3$. In the base case $G^{(3)}$ is the complete graph on 3 vertices where all the edges have the same weight. Then $G^{(n)}$ is defined as the merge of $G^{(3)}$ and $G^{(n-1)}$ where the specified vertices can be chosen arbitrarily. Since the separation of $G^{(n)}$ along the newly constructed complete cut gives back $G^{(3)}$ and $G^{(n-1)}$, from the inductive hypothesis we conclude that $\mathrm{cdim}(G) = \mathrm{cdim}(G_0) + \mathrm{cdim}(G_1) - 1 = 2n - 3$.

ℓ_1-approximate cut dimension. As the cut dimension is at most $2n - 3$, we have to look to other methods in order to show larger lower bounds, if possible. We propose a generalization of the cut dimension which we call the ℓ_1-approximate cut dimension. In order to motivate this, we quickly explain why the cut dimension is a lower bound on the linear query complexity of mincut. The main idea behind the cut dimension lower bound on query complexity is to answer all queries of the algorithm according to an n-vertex graph $G = (V, w)$. Supposing the algorithm makes k queries, we package these into a k-by-$\binom{n}{2}$ matrix A whose rows are the query vectors. If there is a cut $S \in M(G)$ which is not in the rowspace of A, then by the Fredholm alternative there is a vector z such that $Az = \mathbf{0}$, where $\mathbf{0}$ is the all-zero vector, but $\langle S, z\rangle > 0$ and furthermore $z(i) = 0$ whenever $w(i) = 0$. Thus for a sufficiently small $\varepsilon > 0$ we have that $w - \varepsilon z \geq \mathbf{0}$ and so $G' = (V, w - \varepsilon z)$ defines a valid non-negatively weighted graph that has all the same answers to the queries of the algorithm as G. On the other hand, the weight of a minimum cut in G' is strictly smaller than that of G and thus as the algorithm cannot distinguish G and G' it cannot correctly compute the weight of a minimum cut in all n-vertex graphs.

The ℓ_1-approximate cut dimension extends this adversary argument to include *all* the cuts of G instead of just the mincuts. If the minimum cut weight of G is λ and S is a cut with weight $\lambda + \delta$, then the algorithm will still fail if there is a z such that

1. $w - z \geq \mathbf{0}$
2. $Az = \mathbf{0}$
3. $\langle S, z \rangle > \delta$.

The reason is the same: the graph $G' = (V, w - z)$ has all the same answers to the queries made by the algorithm as G yet has a cut with weight strictly smaller than λ.

Taking the dual of the corresponding linear program shows that such a vector z will not exist iff $S - u$ is in the rowspace of A for a vector $u \geq 0$ with $\|u\|_{1,w} \leq \delta$. This leads us to define the (w, c) one-sided row-by-row ℓ_1 approximate rank of a matrix. For a matrix $Y \in \mathbb{R}^{M \times N}$ this is defined by a weight vector $w \in \mathbb{R}^N$ and a cost vector $c \in \mathbb{R}^M$ with $c \geq \mathbf{0}$. It is the minimum rank of a matrix $\tilde{Y}$ such that $\tilde{Y} \leq Y$ and $\|Y(i,:) - \tilde{Y}(i,:)\|_{1,w} \leq c(i)$ for every row i, where $Y(i,:)$ denotes the i^{th} row of Y. Let $G = (V, w)$ be a graph and the weight of a minimum cut in G be λ. The ℓ_1-approximate cut dimension of a graph $G = (V, w)$, denoted $\widetilde{\text{cdim}}(G)$, is the (w, c) one-sided row-by-row ℓ_1-approximate rank of the matrix Y whose rows are the vectors $S \in \{0,1\}^{\binom{n}{2}}$ for every cut S of G, and where $c = Yw - \lambda\mathbf{1}$, and $\mathbf{1}$ is the all-one vector.

Lower bounding the rank under such an ℓ_1 perturbation is a difficult task. However, we are able to show an infinite family of graphs whose ℓ_1-approximate cut dimension is $2n - 2$, thereby showing the ℓ_1-approximate cut dimension can be strictly larger than the cut dimension. This lower bound is of a "direct sum" type. We show that the ℓ_1-approximate cut dimension of K_4, the complete graph on 4 vertices, is 6, giving a tight lower bound of 6 on the number of linear queries needed to compute minimum cut on a 4 vertex graph. We then show that the direct union (see Definition 6) of k copies of K_4 has ℓ_1-approximate cut dimension $6k$. The proof is tailored to the specific properties of the cut vectors of K_4, and makes use of Gaussian elimination and properties of diagonally dominant matrices.

Near-mincuts. Related to the ℓ_1-approximate cut dimension is the question of the cut dimension of *near-mincuts*. For $\alpha \geq 1$ call a cut S of a graph G an α-near-mincut if its weight is at most α times the weight of a minimum cut of G. Let $\mathcal{M}_\alpha(G) = \{S : S \text{ is an } \alpha\text{-near-mincut of } G\}$. It is known that $|\mathcal{M}_\alpha(G)| \leq \binom{n}{2}$ for $\alpha < 4/3$ [25] (see also the beautiful proof given in Theorem 15 of [12]). Even for $\alpha < 3/2$ the number of α-near-mincuts is $O(n^2)$ [17], which is a sharp threshold as there exist graphs with $\Omega(n^3)$ many 3/2-mincuts. There is also a generalization of the cactus representation of mincuts in terms of a tree of deformable polygons that applies to α-near-mincuts for $\alpha < 6/5$ [3]. in Section 8 we show that if G is a *simple* graph then $\dim(\text{span}(\vec{\mathcal{M}}_\alpha(G))) = O(n)$ for any $\alpha < 2$ (Theorem 41). This bound is tight as for $\alpha = 2$ the unweighted complete graph K_n witnesses $\dim(\text{span}(\vec{\mathcal{M}}_2(K_n))) = \binom{n}{2}$. For weighted graphs, on the other hand, we show that for any $\alpha > 1$ there exists an n-vertex weighted graph G with $\dim(\text{span}(\vec{\mathcal{M}}_\alpha(G))) = \binom{n}{2}$.

1.2 Open Problems

Several interesting open problems remain from this work.

- There is still a large gap between the known upper and lower bounds on the deterministic cut/linear query complexity of minimum cut. What is the right answer? We conjecture there is a deterministic cut query algorithm for minimum cut making $O(n^{2-\varepsilon})$ many queries for some $\varepsilon > 0$.
- Is the ℓ_1-approximate cut dimension $O(n)$ for any n-vertex graph? Also can one show a general direct sum theorem for the ℓ_1-approximate cut dimension?

1.3 Organization

The rest of the paper is organized as follows. We review necessary backgrounds about graphs, operations on graphs, and query models in Section 2. In Section 3, we show that the cut dimension is a lower bound on the deterministic linear query complexity of computing minimum cut. We then prove that the cut dimension is at most $2n-3$ in Section 4, and give an explicit construction of graphs with cut dimension $2n-3$ in Section 5. In Section 6, we give another proof for both the upper and lower bounds on $2n-3$ using graph operations. In Section 7 we show a $2n-2$ lower bound on ℓ_1-approximate cut dimension which implies Theorem 3. Finally, in Section 8 we show that for a simple graph G and $1 \le \alpha < 2$ it holds that $\dim(\mathrm{span}(\vec{\mathcal{M}}_\alpha(G))) = O(n)$.

2 Preliminaries

For every natural number n, we denote by $[n]$ the set $\{1, 2, \ldots, n\}$. For a vector $z \in \mathbb{R}^n$ we write $z \ge \mathbf{0}$ if every coordinate of the vector is at least 0, and similarly we write $z = \mathbf{0}$ if z is the all-zero vector. We denote the scalar product of two vectors $z, z' \in \mathbb{R}^n$ by $\langle z, z' \rangle$. For any matrix, denote the rank of A by $\mathrm{rk}(A)$. We denote the disjoint union of sets X and Y by $X \sqcup Y$.

2.1 Graphs, cuts, sets

An undirected *weighted graph* on n vertices is a couple $G = (V, w)$, where V is the set of vertices with $|V| = n$, the set of edge slots $V^{(2)}$ is the set of subsets of V with cardinality 2, and the weight function $w : V^{(2)} \to \mathbb{R}$ is non-negative. We refer to the vertex set of G as $V(G)$. The set of edges of G is defined as $E = \{e \in V^{(2)} : w(e) > 0\}$. When in a graph $G = (V, w)$ the weight of every edge is 1, we say that the graph is *unweighted*, and we refer to it also as $G = (V, E)$; such graph is also called a *simple graph*. For an edge $e = \{u, v\}$, we say that u and v are the endpoints of e. For a subset $X \subseteq V$ of the vertices, we denote by $E(X)$ the set of edges in E which have both endpoints in X, and for disjoint subsets $X, Y \subseteq V$, we denote by $E(X, Y)$ the set of edges with exactly one endpoint in each of the two sets. We extend the weight function w to any subset E' of the edges by $w(E') = \sum_{e \in E'} w(e)$. We will deal only with graphs which have at least 2 vertices.

We fix an ordering $v_1 < v_2 < \cdots < v_n$ of the vertices which induces also an ordering $\{v_1, v_2\}, \{v_1, v_3\}, \ldots, \{v_{n-1}, v_n\}$ of the edge slots as well as an ordering $e_1 < e_2 < \ldots < e_m$ of the $m = |E|$ edges. We view $w \in \mathbb{R}^{\binom{n}{2}}$ as a vector whose i^{th} coordinate gives the (possibly zero) weight of the i^{th} edge slot according to this ordering, and we define $\vec{w} \in \mathbb{R}^m$ as the restriction of w to the edges. With some slight abuse of notation, for a set of edges $S \subseteq E$, we use the same symbol S to also denote the characteristic vector in $\{0,1\}^{\binom{n}{2}}$ of S among all edge slots. We further need the characteristic vector of $S \subseteq E$ among the m edges E, for which we use the notation $\chi(S) \in \{0,1\}^m$. For a family $\mathcal{F}$ of subsets of the edges, we use the notation $\vec{\mathcal{F}} = \{\chi(S) \in \{0,1\}^m : S \in \mathcal{F}\}$.

For $X \subseteq V$, we denote by $\bar{X}$ the set $V \setminus X$. A *cut* S is a set $E(X, \bar{X})$ for some $\emptyset \neq X \subsetneq V$. We call X and $\bar{X}$ the *shores* of S, and we denote the cut by $\Delta(X)$. A cut is a *star cut* if one of its shores is a singleton, otherwise it is *non-star* cut. If the singleton shore of a star cut S is $\{v\}$, then we say that S is a star cut at v. The *weight* of a cut is the sum of the weights of its edges. For a cut S we define the *graph of the cut* S as the unweighted graph $G(S) = (V', E')$ where V' is the set of vertices in V that are endpoints of at least one edge in S, and $E' = S$. We say that a cut S is *connected* if $G(S)$ is a connected graph. A cut is a *minimum* cut, or mincut, for short, if no other cut has smaller weight. We denote by $\mathcal{M}(G)$ be the set of minimum cuts of G. The *cut dimension* of G is $\mathrm{cdim}(G) = \dim(\mathrm{span}(\vec{\mathcal{M}}(G)))$.

Let V be a set of size n. Two sets $X, Y \subseteq V$ are said to *overlap* if $X \cap Y \neq \emptyset, \bar{X} \cap Y \neq \emptyset, X \cap \bar{Y} \neq \emptyset$. A family $\mathcal{G}$ of subsets of V is said to be *laminar* if for all $X, Y \in \mathcal{G}$ it holds that X and Y do not overlap. A set family $\mathcal{G} \subseteq 2^V$ is said to be *closed under overlaps* if for every $X, Y \in \mathcal{G}$ that overlap it holds that $X \cap Y, X \cup Y \in \mathcal{G}$. A laminar subset $\mathcal{L} \subseteq \mathcal{G}$ is said to be *maximal in* $\mathcal{G}$ if for every $X \in \mathcal{G} - \mathcal{L}$ there is a $Y \in \mathcal{L}$ such that X, Y overlap. We say a laminar subset $\mathcal{L}$ is maximal if it is maximal in 2^V.

The sets $X, Y \subseteq V$ *cross* if they overlap and additionally $\bar{X} \cap \bar{Y} \neq \emptyset$. Note that if X, Y cross then so do $X, \bar{Y}$. A set family $\mathcal{G} \subseteq 2^V$ is said to be *cross-free* if for all $X, Y \in \mathcal{G}$ it holds that X and Y do not cross. Observe that if X and Y do not cross then either Y or $\bar{Y}$ is a subset of X or $\bar{X}$. Let $G = (V, w)$ be a graph with n vertices. Two cuts $\Delta(X)$ and $\Delta(Y)$ of G are *crossing* if X and Y are crossing. Let $\mathcal{F} = \{\Delta(X_1), \dots, \Delta(X_k)\}$ be a set of cuts of G. We say that $\mathcal{F}$ is *cross-free family of cuts* if $\mathcal{G} = \{X_1, \dots, X_k\}$ is cross-free. Note that it does not matter which shore we take to be in $\mathcal{G}$.

There is a close relationship between cross-free families of cuts and laminar sets. Let $\mathcal{F} = \{\Delta(X_1), \dots, \Delta(X_k)\}$ be a cross-free family of cuts where each $X_i \subseteq V$, and let $X'_i = X_i$ if $v_1 \notin X_i$ and $X'_i = \bar{X}_i$ otherwise. The *beach* of $\mathcal{F}$ is the set $\mathcal{G} = \{X'_1, \dots, X'_k\}$. For a family of sets $\mathcal{G} \subseteq 2^V$ we say that it is *proper* if $\emptyset, V \notin \mathcal{G}$, and we say that it is *complement free* if it does not contain X, Y with $Y = \bar{X}$.

▷ Claim 4. Let $\mathcal{F}$ be a cross-free family of distinct cuts and $\mathcal{G}$ its beach. Then $\mathcal{G}$ is proper, complement free and laminar.

Proof. First, $\mathcal{G}$ does not contain $\emptyset$ or V because these are not shores of cuts. It is complement free because $\mathcal{F}$ contains distinct cuts, and its beach contains exactly one representative shore from each cut. Finally, we show that it is laminar. Let $X_1, X_2 \in \mathcal{G}$. By definition of a beach, neither of these sets contain v_1, thus $\bar{X}_1 \cap \bar{X}_2 \neq \emptyset$. Therefore if X_1, X_2 overlapped they would also cross, in contradiction to $\mathcal{F}$ being a cross-free family of cuts. ◁

A mincut is *crossless* if no other mincut crosses it. Observe that a star mincut is always crossless. Also, if a mincut $\Delta(X)$ is crossless then for every mincut $\Delta(Y)$, either Y or $\bar{Y}$ is a subset of X or $\bar{X}$. Crossing mincuts have a nice structural property which was already observed by [6].

▷ Claim 5. Let $G = (V, w)$ be a weighted graph. If $\Delta(X), \Delta(Y) \in \mathcal{M}(G)$ cross then $\Delta(X \cap Y), \Delta(X \cup Y) \in \mathcal{M}(G)$.

Proof. We have $\Delta(X \cap Y) \neq \emptyset$ and $\Delta(X \cup Y) \neq V$ because $\Delta(X)$ and $\Delta(Y)$ cross. The cut function is submodular therefore we have

$$w(\Delta(X \cap Y)) + w(\Delta(X \cup Y)) \leq w(\Delta(X)) + w(\Delta(Y)).$$

Let c be the weight of a minimum cut in G. Then the right hand side of the above inequality is equal to $2c$, while its left hand side is at least $2c$. Therefore $w(\Delta(X \cap Y)) + w(\Delta(X \cup Y)) = 2c$ from which the statement follows. ◁

2.2 Operations on graphs

We will use several operations on graphs. The first of these is the *direct union.*

▶ **Definition 6** (direct union). *For two graphs $G_0 = (V_0, w_0), G_1 = (V_1, w_1)$ with disjoint vertex sets, and for vertices $v_0 \in V_0$ and $v_1 \in V_1$, the* direct union *of G_0 and G_1 at vertices v_0, v_1 is the fusion of the two by identifying v_0 and v_1. Formally, the direct union is $G_0^{v_0} \oplus G_1^{v_1} = (V, w)$ where $V = (V_0 \cup V_1 \cup \{v\}) \setminus \{v_0, v_1\}$, for a new vertex $v \notin V_0 \cup V_1$. The weight function of $G_0^{v_0} \oplus G_1^{v_1}$ is defined by*

$$w(\{x,y\}) = \begin{cases} w_b(\{x,y\}) & \textit{if } x,y \in V_b \setminus \{v_b\}, b \in \{0,1\}, \\ w_b(\{x,v_b\}) & \textit{if } x \in V_b \setminus \{v_b\}, y = v, b \in \{0,1\}, \\ 0 & \textit{otherwise.} \end{cases}$$

The cut dimension of a direct union is a simple function of the cut dimensions of its components.

▷ Claim 7. Let $G = G_0^{v_0} \oplus G_1^{v_1}$ be the direct union of G_0 and G_1 at vertices v_0, v_1. Let c_b be the weight of a minimum cut in G_b, for $b = 0, 1$. Then $\mathrm{cdim}(G) = \mathrm{cdim}(G_0) + \mathrm{cdim}(G_1)$ if $c_0 = c_1$, and $\mathrm{cdim}(G) = \mathrm{cdim}(G_b)$ if $c_b < c_{1-b}$.

Proof. Let $\Delta(X)$ be an arbitrary cut of G where $v \notin X$. If $X \not\subseteq V_b$, for $b \in \{0,1\}$, then the weight of the cut $\Delta(X)$ is at least $c_0 + c_1$, and therefore it is not a minimum cut. If $X \subseteq V_b$, for some $b \in \{0,1\}$ then the weight of $\Delta(X)$ in G is the same as the weight of $\Delta(X)$ in G_b. Therefore if $c_0 = c_1$ then every mincut in G_0 and every mincut of G_1 is a mincut of G, and these are the only mincuts. Since their supports are disjoint, we have $\mathrm{cdim}(G) = \mathrm{cdim}(G_0) + \mathrm{cdim}(G_1)$. If $c_b < c_{1-b}$ then only the mincuts of G_b are mincuts of G, and therefore $\mathrm{cdim}(G) = \mathrm{cdim}(G_b)$. ◁

The next two operations, which are inverses of each other, give a decomposition of a graph along a cut into two smaller graphs, and a composition of two graphs into a bigger one by unfolding a star cut in each components. The decomposition operation was essentially defined in [7]. Let $G = (V, w)$ be a weighted graph and let Z be a cut in G with shores X_0 and $X_1 = V \setminus X_0$. The *separation* of G along the cut Z, denoted by $\mathrm{sep}(G, Z)$, is the set of two graphs $\{G_0 = (V_0, w_0), G_1 = (V_1, w_1)\}$, where $V_b = X_b \cup \{v_{1-b}\}$, for $b = 0, 1$ with new vertices v_0, v_1. The respective weight functions are defined by $w_b(\{x,y\}) = w(\{x,y\})$ for any $x, y \in X_b$, and $w_b(\{x, v_{1-b}\}) = \sum_{y \in V_{1-b}} w(\{x,y\})$ for any $x \in X_b$.

Let $G_0 = (V_0, w_0), G_1 = (V_1, w_1)$ be two graphs on disjoint vertex sets, and let $v_b \in V_{1-b}$ be arbitrary vertices for $b \in \{0,1\}$. The *merge* of G_0 and G_1 along the vertices v_1, v_0, denoted by $\mathrm{mer}(\{(G_0, v_1), (G_1, v_0)\})$, is the graph $G = (V, w)$, where $V = (V_0 \cup V_1) \setminus \{v_0, v_1\}$. The weight function in G is defined by $w(\{x,y\}) = w_b(\{x,y\})$ if $x, y \in V_b$, for $b \in \{0,1\}$, and

$$w(\{x,y\}) = w_0(\{x,v_1\}) w_1(\{v_0,y\}), \text{ if } x \in V_0 \text{ and } y \in V_1.$$

It follows from the definitions sep is the left inverse of mer if the star cut at v_1 in V_0 and the star cut at v_0 in V_1 both have weight one, and sep is the right inverse of mer if the weight of the cut Z is one. We formally state the former property.

▷ Claim 8. Let $G_0 = (V_0, w_0)$ and $G_1 = (V_1, w_1)$ have disjoint vertex sets, and let $v_b \in V_{1-b}$ such that $w_b(\Delta(v_{1-b})) = 1$, for $b = 0, 1$. Let Z be the cut in $\mathrm{mer}(\{(G_0, v_1), (G_1, v_0)\})$ whose shores are $V_0 \setminus \{v_1\}$ and $V_1 \setminus \{v_0\}$. Then $w(Z) = 1$ and

$$\mathrm{sep}(\mathrm{mer}(\{(G_0, v_1), (G_1, v_0)\}), Z) = \{G_0, G_1\}.$$

2.3 Query models

▶ **Definition 9** (MINCUT$_n$). *The input in the* MINCUT$_n$ *problem is an n-vertex weighted undirected graph $G = (V, w)$. The required output on G is the weight of a minimum cut in G.*

A deterministic algorithm correctly solves the MINCUT_n problem if it outputs the correct mincut weight for every n-vertex input graph G. We consider algorithms given two models of query access to the input graph $G = (V, w)$, linear queries and cut queries. A *linear query* for G is a vector $x \in \mathbb{R}^{\binom{n}{2}}$, and the query is answered by $\langle x, w \rangle$. A cut query is a vector $x \in \{0,1\}^{\binom{n}{2}}$ which is the characteristic vector of a cut in the complete n-vertex graph. The answer to a cut query is again $\langle x, w \rangle$. Clearly any cut query algorithm can be simulated by a linear query algorithm.

We use $D_{\mathrm{cut}}(\mathrm{MINCUT}_n)$ to denote the minimum, over all deterministic query algorithms $\mathcal{A}$ that correctly solve MINCUT_n, of the maximum over all n-vertex input graphs $G = (V, w)$ of the number of cut queries made by $\mathcal{A}$ on G. $D_{\mathrm{lin}}(\mathrm{MINCUT}_n)$ is defined analogously for linear queries.

Some authors instead define the output of the minimum cut problem to be a cut S that achieves the minimum weight, rather than the weight itself. Over n-vertex weighted graphs let us denote this problem as $\mathrm{ARGMINCUT}_n$. For linear and cut queries, an algorithm that finds a minimum cut S can also return the weight of S with one additional query. Thus $D_{\mathrm{lin,cut}}(\mathrm{ARGMINCUT}_n) \geq D_{\mathrm{lin,cut}}(\mathrm{MINCUT}_n) - 1$, and the lower bounds we prove for MINCUT_n can be applied, minus 1, to $\mathrm{ARGMINCUT}_n$ as well.

3 Lower bounds on the linear query complexity of MINCUT

Graur et al. [13] introduce the cut dimension as a means to show lower bounds on the deterministic cut query complexity of computing minimum cut.

▶ **Theorem 10** ([13]). *If there is an n-vertex weighted graph $G = (V, w)$ with* $\mathrm{cdim}(G) = k$ *then $D_{\mathrm{cut}}(\mathrm{MINCUT}_n) \geq k$.*

We show that this theorem even holds with respect to a stronger computational model where the algorithm is able to make linear queries. We also give a generalization of the cut dimension to a quantity which is at least as large, and can be strictly larger, that we call the ℓ_1-approximate cut dimension. We now give an overview of the Graur et al. [13] argument in the context of linear queries and how we can extend it.

The proof of Theorem 10 is based on an adversary argument. Suppose a deterministic algorithm makes k linear queries and consider the execution of the algorithm on a fixed n-vertex graph $G = (V, w)$ whose set of minimum cuts is $\mathcal{M}(G)$. Make a k-by-$\binom{n}{2}$ matrix A whose rows are the query vectors asked by the algorithm. Suppose we can find a vector $z \in \mathbb{R}^{\binom{n}{2}}$ such that

1. $w - z \geq \mathbf{0}$,
2. $Az = \mathbf{0}$,
3. There is a cut $S \in \mathcal{M}(G)$ such that $\langle S, z \rangle > 0$.

The existence of such a vector z means the algorithm cannot correctly compute minimum cut weight on all weighted n-vertex graphs. The reason is that $G' = (V, w - z)$ is a valid non-negatively weighted graph by (1), has the same answers on all queries asked by the algorithm by (2), and by (3) has minimum cut weight at most $\langle S, w - z \rangle = \langle S, w \rangle - \langle S, z \rangle < \langle S, w \rangle$, which is strictly less than the minimum cut weight of G. As with k queries the algorithm cannot distinguish whether the input is G or G', it cannot correctly output the minimum cut weight for all n-vertex weighted graphs.

A weaker condition than (3) suffices for this argument to work. Suppose that the minimum cut weight in G is c^*. Then the argument still goes through with the condition

3'. There is a cut S such that $\langle S, z\rangle > \langle S, w\rangle - c^*$.

This is because the algorithm cannot distinguish the graph G with minimum cut weight c^* from the graph $G' = (V, w - z)$ which has minimum cut weight at most $\langle S, w - z\rangle < c^*$.

In order to understand what kind of bound this argument gives, for fixed w, A, S we define the quantity $\alpha(w, A, S)$ which is given by the following linear program.

$$\begin{aligned} \alpha(w, A, S) = \underset{z}{\text{maximize}} \quad & \langle S, z\rangle \\ \text{subject to} \quad & w - z \geq 0 \\ & Az = \mathbf{0} \end{aligned}$$

Taking the dual of this program gives

$$\begin{aligned} \alpha(w, A, S) = \underset{v}{\text{minimize}} \quad & \langle S - A^T v, w\rangle \\ \text{subject to} \quad & S - A^T v \geq 0 \end{aligned}$$

The dual tells us that a vector z having large overlap with S and satisfying items (1), (2) above exists iff the vector S is *far away* from the rowspace of A. The notion of far away here is a one-sided ℓ_1 distance weighted by w. It is one-sided because the condition $S - A^T v \geq 0$ tells us we are looking to approximate S by vectors in the rowspace of A that are entrywise at most S. As $S - A^T v \geq 0$ and $w \geq 0$ this means $\langle S - A^T v, w\rangle = \sum_i |w(i) \cdot (S(i) - A^T v)| = \|S - A^T v\|_{1,w}$, where $\|u\|_{1,w}$ is defined to be $\sum_i |u(i)w(i)|$. Thus the value of the dual can be interpreted as the one-sided $\|\cdot\|_{1,w}$ distance between S and the rowspace of A.

This leads us to define an ℓ_1 approximate version of the cut dimension. The notion we need is given by the following definitions.

▶ **Definition 11** (one-sided row-by-row ℓ_1-approximate rank). *Let $Y \in \mathbb{R}^{M\times N}$ be a matrix, $w \in \mathbb{R}^N$ a weight vector and $c \in \mathbb{R}^M$ a cost vector. We define the (w, c) one-sided row-by-row ℓ_1-approximate rank of Y to be the minimum rank of a matrix $\tilde{Y}$ such that $\tilde{Y} \leq Y$ and $\|Y(i, :) - \tilde{Y}(i, :)\|_{1,w} \leq c(i)$, for all $1 \leq i \leq M$.*

▶ **Definition 12** (ℓ_1-approximate cut dimension). *Let $G = (V, w)$ be an n-vertex weighted undirected graph with minimum cut weight c^*. Let M be $(2^{n-1} - 1)$-by-$\binom{n}{2}$ matrix whose rows are $S \in \{0, 1\}^{\binom{n}{2}}$ for all cuts S of G. Let $c = Mw - c^*\mathbf{1}$, where $\mathbf{1}$ is the all one vector. Then the ℓ_1-approximate cut dimension of G, denoted $\widetilde{\mathrm{cdim}}(G)$, is the (w, c) one-sided row-by-row ℓ_1-approximate rank of M.*

▶ **Theorem 13.** *If there is an n-vertex graph weighted graph $G = (V, w)$ with $\widetilde{\mathrm{cdim}}(G) = k$ then $D_{\mathrm{lin}}(\mathrm{MINCUT}_n) \geq k$.*

Proof. Let $G = (V, w)$ be a graph with $\widetilde{\mathrm{cdim}}(G) = k$ and let c^* be the minimum cut weight of G. Suppose for contradiction there is a deterministic $k - 1$ linear query algorithm that correctly computes the minimum cut of any n-vertex graph. Run this algorithm answering queries according to G and package the queries into a $(k - 1)$-by-$\binom{n}{2}$ matrix A.

As the algorithm is correct, for every cut S of G it must be the case that $\alpha(w, A, S) \leq \langle S, w\rangle - c^*$. If not, the graph $G' = (V, w - z)$, where z is an optimal solution to the primal of $\alpha(w, A, S)$, has minimum cut weight strictly smaller than c^*, yet G' cannot be distinguished from G by the algorithm. Thus by the dual formulation of $\alpha(w, A, S)$, this means that for every cut S of G there is a vector $\tilde{S} = A^T v$ in the rowspace of A such that $\tilde{S} \leq S$ and $\|S - \tilde{S}\|_{1,w} \leq \langle S, w\rangle - c^*$. The matrix $\tilde{M}$ whose rows are $\tilde{S}$ for all cuts S therefore witnesses that $\widetilde{\mathrm{cdim}}(G) \leq \mathrm{rk}(A) \leq k - 1$, a contradiction. ◀

▶ **Lemma 14.** *For any weighted graph $G = (V, w)$ we have* $\mathrm{cdim}(G) \leq \widetilde{\mathrm{cdim}}(G)$.

Proof. Suppose that $G = (V, w)$ has minimum cut weight c^*, and let $\mathcal{M}(G)$ be the set of minimum cuts of G. Let M be the $(2^{n-1} - 1)$-by-$\binom{n}{2}$ matrix whose rows are $S \in \{0,1\}^{\binom{n}{2}}$ for all cuts S of G and let $c = Mw - c^*$.

Let Y be the submatrix of M where rows are restricted to cuts in $\mathcal{M}(G)$ and columns are restricted to the edge slots e where $w(e) > 0$. Thus the rows of Y are exactly the vectors $\chi(S)$ for $S \in \mathcal{M}(G)$. and the rank of Y is $\mathrm{cdim}(G)$. Any matrix $\tilde{M}$ which satisfies $\tilde{M} \leq M$ and $\|M(i,:) - \tilde{M}(i,:)\|_{1,w} \leq c(i)$ for all i must contain Y as a submatrix, as $c(i) = 0$ for rows i that correspond to minimum cuts and w is positive on the edge slots labeling the columns of Y. Thus $\mathrm{rk}(\tilde{M}) \geq \mathrm{rk}(Y)$ for any (w, c) one-sided row-by-row ℓ_1 approximation $\tilde{M}$ of M, giving the lemma. ◀

In Section 7 we will see that $\widetilde{\mathrm{cdim}}(G)$ can be strictly larger than $\mathrm{cdim}(G)$. From Theorem 13 and Lemma 14 we obtain the following corollary.

▶ **Corollary 15.** *If there is an n-vertex weighted graph $G = (V, w)$ with* $\mathrm{cdim}(G) = k$ *then* $D_{\mathrm{lin}}(\mathrm{MINCUT}_n) \geq k$.

4 The cut dimension is at most $2n - 3$

In this section we prove Theorem 1 that $\mathrm{cdim}(G) \leq 2n - 3$ for any undirected weighted graph G on $n \geq 2$ vertices. This will follow from two facts:

1. For $n \geq 2$ a cross-free family of cuts in an n-vertex graph has cardinality at most $2n - 3$.
2. If $\mathcal{L} \subseteq \mathcal{M}(G)$ is a maximal cross-free subset of the mincuts of G then $\mathrm{span}(\vec{\mathcal{L}}) = \mathrm{span}(\vec{\mathcal{M}}(G))$.

We remind the reader that $\vec{\mathcal{L}} = \{\chi(S) : S \in \mathcal{L}\}$ where $\chi(S) \in \{0,1\}^{|E|}$ is the characteristic vector of the cut S amongst the edges of G.

These two facts are presented in the next two subsections.

4.1 Cardinality of a cross-free family of cuts

Recall from Claim 4 that if $\mathcal{L}$ is a cross-free family of cuts then the beach $\mathcal{G}$ of $\mathcal{L}$ is a laminar family of sets. A standard inductive proof shows that a laminar family of subsets of a universe of cardinality n that contains no singletons has size at most $n - 1$, and thus a laminar family in general has size at most $2n - 1$. A beach has the additional properties of being proper and complement free which allows one to prove an upper bound of $2n - 3$. This is mentioned by Goemans [11] in the paragraph after Theorem 4 under the heading "Size of a Laminar Family", who observes that the standard inductive proof also implies the bound is attained only if the family includes the universe and at least one set and its complement. See also Corollary 2.15 of [22], where it is shown that a proper laminar family has cardinality at most $2n - 2$.

▶ **Lemma 16.** *Let $n \geq 2$, V a set of cardinality n, and $\mathcal{G} \subseteq 2^V$ be a family of sets which is proper and laminar. Then $|\mathcal{G}| \leq 2n - 2$. If $\mathcal{G}$ is proper, laminar, and complement free then $|\mathcal{G}| \leq 2n - 3$.*

Proof. First we show the $2n - 2$ upper bound. We prove by induction. Consider first the base case where $n = 2$ and $V = \{v_1, v_2\}$. As $\emptyset, V \notin \mathcal{G}$ the only possible elements to include in $\mathcal{G}$ are $\{v_1\}, \{v_2\}$ and $|\mathcal{G}| \leq 2 = 2n - 2$.

Now we assume the statement is true for families of sets on a universe of $n-1$ elements and show it holds for families of sets on a universe of size n. Let $\mathcal{G} \subseteq 2^V$ be a proper laminar family. We say that $X \in \mathcal{G}$ is maximal if there is no set $Y \in \mathcal{G}$ with $X \subset Y$. Let $X_1, \ldots, X_m$ be the maximal sets in $\mathcal{G}$. Note that we must have $X_i \cap X_j = \emptyset$ for all $i \neq j \in [m]$. This is because for distinct maximal sets $X_i - X_j, X_j - X_i \neq \emptyset$ thus if $X_i \cap X_j \neq \emptyset$ they would be overlapping. If $\cup_{i=1}^m X_i \subsetneq V$ then the result already holds by the induction hypothesis. Thus we may assume $m \geq 2$ and $X_1, \ldots, X_m$ form a partition of V. The family $\mathcal{F}_1 = \{Y : Y \subsetneq X_1\}$ is a laminar family on the universe X_1 which does not contain X_1. Hence by the induction hypothesis it has at most $2|X_1| - 2$ many sets. This holds for all $i = 1, \ldots, m$, thus including $X_1, \ldots, X_m$ the total number of sets is $\sum_{i=1}^m 2|X_i| - m \leq 2n - 2$.

Now we show the $2n-3$ upper bound additionally assuming the family is complement free. We show this result directly using the upper bound of $2n-2$ we have just shown on the size of proper laminar families. Let $\mathcal{G} \subseteq 2^V$ be proper, laminar, and complement free, and let $X_1, \ldots, X_m$ be the maximal sets in $\mathcal{G}$, which again must be disjoint. The number of subsets strictly contained in X_i is at most $2|X_i| - 2$ by the previous result. Thus, including $X_1, \ldots, X_m$ we can upper bound the size of $\mathcal{G}$ by $\sum_{i=1}^m 2|X_i| - m$. If $m > 2$ then the upper bound of $2n-3$ already holds. If $m = 1$ then as $\mathcal{G}$ is a proper family we must have $|X_1| \leq n - 1$ in which case the upper bound of $2n-3$ holds as well. Finally, consider the case $m = 2$. In this case, if $|X_1 \cup X_2| < n$ then the bound already holds. If $X_1 \cup X_2 = V$ then $X_2 = \bar{X}_1$ and we must exclude one of these sets, giving a bound of $2n - 2 - 1 = 2n - 3$. ◀

▶ Remark 17. From the proof in the proper, laminar, complement-free case we can observe for what maximal sets equality in the upper bound can hold. The first is the case where there are three maximal sets X_1, X_2, X_3 that form a partition of $[n]$. With $V = [6]$ an example of this type saturating the bound is $\mathcal{G} = \{\{1\}, \ldots, \{6\}, \{1,2\}, \{3,4\}, \{5,6\}\}$. The second is the case where there are two maximal sets X_1, X_2 that form a partition of $[n]$ and exactly one of X_1, X_2 is not included. The latter includes the case where there is a single maximal set X_1 of size $|X_1| = n - 1$. For $V = [6]$, an example of this type is $\mathcal{G} = \{\{2\}, \ldots, \{6\}, \{2,3\}, \{2,3,4\}, \{2,3,4,5\}, \{2,3,4,5,6\}\}$.

Chandran and Ram (Lemma 2.13 in [5]) show that if the set $\mathcal{M}(G)$ of minimum cuts of a graph G is cross-free, then $|\mathcal{M}(G)| \leq 2n - 3$. This is an easy corollary of Lemma 16, which gives something more general.

▶ **Corollary 18.** *Let $G = (V, w)$ be a graph on $n \geq 2$ vertices. Let $\mathcal{L} \subseteq \mathcal{M}(G)$ be a subset of minimum cuts that is cross-free. Then $|\mathcal{L}| \leq 2n - 3$.*

4.2 Spanning

Let $\mathcal{L} \subseteq \mathcal{M}(G)$ be a maximal cross-free subset of $\mathcal{M}(G)$. Here maximal means that for any cut $S \in \mathcal{M}(G) \setminus \mathcal{L}$ there is a cut $T \in \mathcal{L}$ that crosses S. The fact that $\mathrm{span}(\vec{\mathcal{L}}) = \mathrm{span}(\vec{\mathcal{M}}(G))$ essentially follows from a key lemma of Jain in his factor of 2 approximation algorithm for the survivable network design problem (Lemma 4.2 in [18]). Another application of a similar lemma can be found in Goeman's approximation algorithm for the bounded-degree minimum spanning tree problem [11].

The context of Jain's lemma is slightly different than ours, as we now explain. Instead of mincuts, Jain considers the set of cuts $\mathcal{T}$ which saturate the inequalities of a particular linear program. He shows that the set $\mathcal{T}$ has the property that if $\Delta(X), \Delta(Y) \in \mathcal{T}$ cross then either

1. $\Delta(X \cap Y), \Delta(X \cup Y) \in \mathcal{T}$ and $\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y))$, or
2. $X \setminus Y, Y \setminus X \in \mathcal{T}$ and $\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \setminus Y)) + \chi(\Delta(Y \setminus X))$.

As shown by Dinitz, Karzanov, and Lomonosov [6], for crossing mincuts $\Delta(X), \Delta(Y)$ *both* items (1), (2) hold (see Proposition 45 for a proof). Thus Jain's lemma applies to $\mathcal{M}(G)$ as well.

▶ **Lemma 19** ([18]). *Let $G = (V, w)$ be a graph and $\mathcal{L} \subseteq \mathcal{M}(G)$ be a maximal cross-free family of mincuts. Then* $\mathrm{span}(\vec{\mathcal{L}}) = \mathrm{span}(\vec{\mathcal{M}}(G))$.

For completeness, we include a full proof of Lemma 19 in Appendix A.

We now can give the first proof of our main upper bound that for any $n \geq 2$ an n-vertex graph $G = (V, w)$ has $\mathrm{cdim}(G) \leq 2n - 3$.

Proof of Theorem 1. Follows from Corollary 18 and Lemma 19. ◀

5 Explicit construction of graphs with cut dimension $2n - 3$

In this section we prove Theorem 2 by giving a general technique to explicitly construct graphs of cut dimension $2n - 3$. We focus on constructing graphs $G = (V, w)$ where w is strictly positive, i.e. where G is a complete weighted graph. The main lemma of this section, Lemma 27, shows that, in a complete weighted graph, for any cross-free family of cuts $\mathcal{L}$ the vectors in $\vec{\mathcal{L}}$ are linearly independent.

Thus to construct a graph with cut dimension $2n - 3$ it suffices to construct a complete weighted graph whose set of mincuts is a cross-free family of cuts of cardinality $2n - 3$. Such a graph is constructed for every $n \geq 2$ in Theorem 5.2 of [5]. Combining this construction with our linear independence result Lemma 27 gives a proof of our main lower bound Theorem 2.

In Section 5.3 we go further and show for any maximal cross-free family $\mathcal{F} \subseteq 2^{[n]}$ there is a complete weighted graph $G = ([n], w)$ with $\mathcal{M}(G) = \{\Delta(X) : X \in \mathcal{F}\}$. Moreover, we give an explicit formula for the weight vector w. Part of this construction is a lemma, Lemma 29, which may be of independent interest: it says that if $\mathcal{L}$ is a maximal family of cross-free cuts in a graph G, and all cuts in $\mathcal{L}$ have the same weight c, then c is the weight of the minimum cut in G.

A key tool for showing the linear independence of cuts from a cross-free family is the tree representation of a laminar family, which we go over next.

5.1 Tree representation

▶ **Definition 20.** *For an unweighted directed graph $G = (V, E)$ we let $\delta^+(X) = \{(x, y) \in E : x \in X, y \in V - X\}$. For a singleton $v \in V$ we write $\delta^+(v)$ instead of $\delta^+(\{v\})$.*

▶ **Definition 21** (Arborescence). *An* arborescence *is a directed rooted tree where all edges point away from the root. A vertex of an arborescence which is not the root or a leaf we call an* internal vertex*.*

▶ **Definition 22** (Tree representation). *Let T be a directed graph whose underlying undirected graph is a tree. Let U be a finite set and $\phi : U \to V(T)$. For $e = (x, y) \in E(T)$ define S_e as*

$$S_e = \{s \in U : \phi(s) \textit{ is in the same connected component of } T - e \textit{ as } y\} \ .$$

Then (T, ϕ) defines a set family $\mathcal{F} = \mathcal{F}(T, \phi)$ where $\mathcal{F} = \{S_e : e \in E(T)\}$. We say that (T, ϕ) is a tree representation *of $(U, \mathcal{F})$. We call (T, ϕ) a* faithful tree representation *if $|E(T)| = |\mathcal{F}|$. For $v \in V(T)$, if there is a $u \in U$ such that $\phi(u) = v$ then we say that v has a* label*.*

We will need the fact that a laminar set family has a faithful tree representation by an arborescence. A textbook proof of this fact can be found in Korte and Vygen Proposition 2.14 [22]. While they do not explicitly say the tree representation they construct is faithful, this is clear from the proof.

▶ **Proposition 23.** *Let $(U, \mathcal{F})$ be laminar family. Then there is a faithful tree representation (T, ϕ) of $(U, \mathcal{F})$ where T is an arborescence.*

Recall from Claim 4 that if $\mathcal{L}$ is a cross-free family of cuts then its beach $\mathcal{G}$ is laminar, and thus has a tree representation.

▶ **Lemma 24** (Tree structure of maximal cross-free families). *Let $\mathcal{L}$ be a maximal family of cross-free cuts of a graph $G = ([n], w)$ and $\mathcal{G} \subseteq 2^{[n]}$ its beach. Then in a faithful tree representation (T, ϕ) of $\mathcal{G}$ it holds that*

1. *The root r is labeled by 1 and has $|\delta^+(r)| = 1$*
2. *There are $n-1$ leaves of T each with a distinct label in $\{2, \dots, n\}$.*
3. *Every internal vertex v has $|\delta^+(v)| = 2$.*

Proof. As by the definition of a beach, sets do not contain 1, this means that 1 must be the label of the root. As star cuts do not cross any other cut, if $\mathcal{L}$ is maximal it must contain all the star cuts. This means that $\mathcal{G}$ contains the sets $\{2\}, \dots, \{n\}, \{2, \dots, n\}$. Thus the outdegree of the root must be 1, as this outgoing edge represents the set $\{2, \dots, n\}$. Further there must be $n-1$ leaves which are labeled by $2, \dots, n$. We have now accounted for all the labels, thus no internal vertex has a label. Further, if there was a leaf v with parent u such that v did not have a label, then (u, v) would represent the empty set, which by definition is not in $\mathcal{G}$. Thus there are exactly $n-1$ leaves.

It remains to show that every internal vertex v of T which is not the root has $|\delta^+(v)| = 2$. Let v be an internal vertex, and as v is not the root, let u be its parent, and as v is not a leaf let w be a child of v. If $|\delta^+(v)| = 1$ then the edges $(u, v), (v, w)$ would represent the same set, as v is not labeled. This contradicts the fact that (T, ϕ) is a faithful tree representation. Now suppose $|\delta^+(v)| > 2$ and let w, x, y be three of its children. Consider the sets $X_1, X_2, X_3 \in \mathcal{G}$ represented by the edges $(v, w), (v, x), (v, y)$. Further the edge (u, v) represents a set $A \in \mathcal{G}$ with $X_1 \cup X_2 \cup X_3 \subseteq A$. We claim that in this case $\mathcal{L}$ is not maximal because the cut $\Delta(X_1 \cup X_2)$ does not cross any cut in $\mathcal{L}$. Indeed, $X_1 \cup X_2$ is contained in all the sets represented by edges on the path from v to the root, and is disjoint from the sets represented by any other edge of T. Thus we have a contradiction. ◀

▶ **Corollary 25.** *Let $\mathcal{L}$ be a maximal family of cross-free cuts of a graph $G = ([n], w)$. Then $|\mathcal{L}| = 2n - 3$.*

Proof. Let $\mathcal{G}$ be the beach of $\mathcal{L}$ and (T, ϕ) a faithful tree representation of $\mathcal{G}$. As (T, ϕ) is faithful $|E(T)| = |\mathcal{L}|$. Let T' be the undirected graph underlying T. Clearly $|E(T')| = |E(T)|$. We use Lemma 24 to count $|E(T')|$. Let i be the number of internal vertices of T', each of which has degree 3. There are also n non-internal vertices each of which has degree 1. Thus $|E(T')| = (3i + n)/2$. Also as T' is a tree $|E(T')| = |V(T')| - 1 = n + i - 1$. Hence $i = n - 2$ and $|\mathcal{L}| = |E(T)| = |E(T')| = 2n - 3$. ◀

5.2 Linear independence

We now show the main theorem of this section that in a complete weighted graph any set $\vec{\mathcal{L}}$ of cut vectors of a cross-free family of cuts $\mathcal{L}$ is linearly independent. We will use the tree representation (T, ϕ) of the beach $\mathcal{G}$ of $\mathcal{L}$ to do this via the following lemma.

▶ **Lemma 26.** *Let T be an arborescence with root r and $\psi : E(T) \to \mathbb{R}$. Let U be a finite set and $\phi : U \to V(T)$. Suppose that T, ϕ, ψ have the property that*

1. *The root r is labeled and has $|\delta^+(r)| = 1$.*
2. *Every internal vertex v is unlabeled and has $|\delta^+(v)| = 2$.*
3. *Every leaf of T has a label.*
4. *For every $s, t \in U$ it holds that $\sum_{e \in \phi(s) - \phi(t)} \psi(e) = 0$, where $\phi(s) - \phi(t)$ is the set of edges on the undirected path from $\phi(s)$ to $\phi(t)$.*

Then ψ is identically 0.

Proof. We will prove by induction on the depth of the arborescence. We need a slightly different statement for the inductive hypothesis since when considering a sub-arborescence T' of T we do not know that the root of T' has property (1).

Inductive hypothesis. Let T be an arborescence with root r that is unlabeled and has $|\delta^+(r)| = 2$, and further suppose T, ϕ, ψ satisfy conditions (2)-(4) of the proposition. Then letting u, v be the children of r it holds that $\psi((r, u)) = -\psi((r, v))$ and for any other edge $e \in E(T), e \neq (r, u), (r, v)$ it holds that $\psi(e) = 0$.

For the base case consider a tree of depth 1, with root r and two children u, v which are leaves. As they are leaves, u, v are labeled which, considering the path from u to v, means $\psi((r, u)) + \psi((r, v)) = 0$. This concludes the base case.

Now we prove the inductive step. Let r be the root of a tree with children u, v. We consider two cases:

Case 1: one of u, v is a leaf. Suppose without loss of generality that u is a leaf and v is an internal node with children v_1, v_2. By the inductive hypothesis $\psi((v, v_1)) + \psi((v, v_2)) = 0$ and ψ is identically 0 on the subtrees rooted at v_1, v_2. Let y_1, y_2 be leaves that are descendants of v_1, v_2 respectively (and can possibly be y_1, y_2 themselves). Considering the path from u to y_1 and y_2 we have the equations

$$\psi((r, u) + \psi((r, v)) + \psi((v, v_1)) = 0$$
$$\psi((r, u) + \psi((r, v)) + \psi((v, v_2)) = 0$$

As $\psi((v, v_1)) + \psi((v, v_2)) = 0$, adding these equations shows that $\psi((r, u) + \psi((r, v)) = 0$, as desired. Substituting this back into the equations further implies that $\psi((v, v_1)) = \psi((v, v_2)) = 0$ so ψ is identically 0 on the subtree rooted at v completing this case.

Case 2: both u, v are internal vertices. Let the children of u be u_1, u_2 and the children of v be v_1, v_2. By the inductive hypothesis, $\psi(\cdot)$ is identically zero on the sub-trees rooted at u_1, u_2, v_1, v_2 and we have $\psi((u, u_1)) + \psi((u, u_2)) = \psi((v, v_1)) + \psi((v, v_2)) = 0$. We must show that $\psi((u, u_1)) = \psi((u, u_2)) = \psi((v, v_1)) = \psi((v, v_2)) = 0$ and that $\psi((r, u)) + \psi((r, v)) = 0$.

Let x_1, x_2 be a leaves that are descendants of u_1, u_2, respectively, and similarly let y_1, y_2 be leaves that are descendants of v_1, v_2, respectively. By assumption all of these leaves are labeled. Considering the paths from $x_b - y_{b'}$ for $b, b' \in \{0, 1\}$ we obtain the following four constraints on ψ:

$$\psi((u, u_1)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_1)) = 0$$
$$\psi((u, u_1)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_2)) = 0$$
$$\psi((u, u_2)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_1)) = 0$$
$$\psi((u, u_2)) + \psi((r, u)) + \psi((r, v)) + \psi((v, v_2)) = 0$$

Adding all four equations and using $\psi((u,u_1)) + \psi((u,u_2)) = \psi((v,v_1)) + \psi((v,v_2)) = 0$ shows that $\psi((r,u)) + \psi((r,v)) = 0$. Taking this into account, adding the first two equations then shows $\psi((u,u_1)) = 0$, and adding the last two equations shows $\psi((u,u_2)) = 0$. This then also means $\psi((v,v_1)) = \psi((v,v_2)) = 0$.

We have now shown the inductive statement holds. It remains to see why this implies the lemma. Let r be the root of the tree, let u be the child of r, and let u_1, u_2 be the children of u. By the inductive statement we have that $\psi((u,u_1)) + \psi((u,u_2)) = 0$ and ψ is identically zero on the subtree rooted at u_1 and the subtree rooted at u_2. Let x_1, x_2 be leaves which are descendants of u_1, u_2, respectively. As the root has a label, considering the path from r to u_1 implies that $\psi((r,u)) + \psi((u,u_1)) = 0$ and considering the path from r to u_2 implies $\psi((r,u)) + \psi((u,u_2)) = 0$. Adding these equations implies that $\psi((r,u)) = 0$, from which it then follows that $\psi((u,u_1)) = \psi((u,u_2)) = 0$. ◀

▶ **Lemma 27.** *Let $G = ([n], w)$ be a complete weighted graph and let $\mathcal{L}$ be a cross-free family of cuts. Then $\vec{\mathcal{L}} = \{\chi(S) : S \in \mathcal{L}\}$ form a linearly independent set of vectors.*

Proof. We may assume that $\mathcal{L}$ is a *maximal* cross-free family, as showing that a superset of $\vec{\mathcal{L}}$ is linearly independent implies that $\vec{\mathcal{L}}$ is as well. Thus suppose $\mathcal{L}$ is a maximal cross-free family and let $\mathcal{G}$ be its beach. Let (T, ϕ) be a faithful tree representation of $\mathcal{G}$. By Lemma 24 we have that (T, ϕ) satisfy conditions (1)-(3) of Lemma 26.

Now we ask the question: for an edge $\{i,j\} \in E(G)$ which sets $S \in \mathcal{L}$ contain it? This has a very nice description in terms of the tree decomposition. Let $u, v \in V(T)$ be the vertices with $\phi(i) = u, \phi(j) = v$. Then the sets containing i are the sets represented by edges from the root to u; the sets containing j are the sets represented by the edges on the path from the root to v. Therefore the sets which contain i but not j or j but not i, are exactly those represented by the edges on the path from u to v in the undirected tree underlying T. Thus the cuts which contain the edge $\{i,j\}$ are exactly those with a shore which is represented by an edge on the path from u to v in undirected graph underlying T.

Consider a linear combination $\sum_{S \in \mathcal{L}} \alpha_S \chi(S) = \mathbf{0}$ which is equal to the all zero vector. The $\{i,j\}$ coordinate of this equation says that $\sum_{S \in \mathcal{L}, \{i,j\} \in S} \alpha_S \chi(S)(\{i,j\}) = 0$. This sum is exactly over the sets represented by edges on the path from $\phi(i)$ to $\phi(j)$. As this sum must be zero for every edge $\{i,j\}$, this says that if we let $\psi(e) = \alpha_S$ where the edge e represents a shore of S then for any two labeled vertices $u, v \in V(T)$ the sum of $\psi(e)$ over the edges on the path from u to v is zero. Thus also condition (4) of Lemma 26 is satisfied. Hence all of the conditions of Lemma 26 hold which implies that ψ must be identically zero and therefore all coefficients $\alpha_S = 0$. This shows that $\{\chi(S) : S \in \mathcal{L}\}$ is a linearly independent set. ◀

We can now give the first proof of our main lower bound result on the cut dimension Theorem 2, which says that for every integer $n \geq 2$ there is an n-vertex weighted graph $G = (V, w)$ with $\mathrm{cdim}(G) \geq 2n - 3$.

Proof of Theorem 2. For every integer $n \geq 2$, Theorem 5.2 of [5] constructs a complete weighted graph $G = (V, w)$ on n vertices such that $\mathcal{M}(G)$ is a cross-free family of size $|\mathcal{M}(G)| = 2n - 3$. By Lemma 27 the vectors in $\vec{\mathcal{M}}$ form a linearly independent set, thus $\mathrm{cdim}(G) \geq 2n - 3$. ◀

5.3 Constructing graphs with a cross-free set of mincuts

In this subsection we explicitly construct, for any maximal cross-free family $\mathcal{F} \subseteq 2^{[n]}$, a complete weighted graph $G = ([n], w)$ with $\mathcal{M}(G) = \{\Delta(X) : X \in \mathcal{F}\}$. This task is made easier by the next lemma. We first need a definition.

▶ **Definition 28.** *Let $\mathcal{F} \subseteq 2^V$. For a subset $X \subseteq V$, let* $\mathrm{overlap}_{\mathcal{F}}(X) = \{Y \in \mathcal{F} : X, Y \text{ overlap}\}$.

▶ **Lemma 29.** *Let $G = (V, w)$ be a graph and $\mathcal{L}$ be a maximal cross-free family of cuts. Suppose that for all $S \in \mathcal{L}$ it holds that $w(S) = c$. Then the weight of a minimum cut in G is c.*

Proof. Let $\mathcal{G}$ be the beach of $\mathcal{L}$. Suppose for a contradiction that the weight of a minimum cut of G is $< c$. Let $\mathcal{T} = \{Z : \emptyset \neq Z \subsetneq V, v_1 \notin Z, Z \notin \mathcal{G}, w(\Delta(Z)) < c\}$ and

$$X = \operatorname*{argmin}_{Z} \{|\mathrm{overlap}_{\mathcal{G}}(Z)| : Z \in \mathcal{T}\} \ .$$

In the following we always use $\mathrm{overlap}(\cdot)$ with respect to $\mathcal{G}$ and drop the subscript. As $|\mathrm{overlap}(X)| \geq 1$, let $Y \in \mathrm{overlap}(X)$. As shown in Appendix A Lemma 46, both $|\mathrm{overlap}(X \cap Y)|$ and $|\mathrm{overlap}(X \cup Y)|$ are strictly smaller than $|\mathrm{overlap}(X)|$. Thus it must be the case that $X \cap Y, X \cup Y \notin \mathcal{T}$. Let us take the case of $X \cap Y$. It does not contain v_1, as neither X nor Y do, and it is a nonempty set by the definition of overlap. Thus it must be the case that either $w(\Delta(X \cap Y)) \geq c$ or that $X \cap Y \in \mathcal{G}$, which implies $w(\Delta(X \cap Y)) = c$. The same argument holds for $X \cup Y$, thus both $w(\Delta(X \cap Y)), w(\Delta(X \cup Y)) \geq c$.

However by submodularity of the cut function we have $w(\Delta(X \cap Y)) + w(\Delta(X \cup Y)) \leq w(\Delta(X)) + w(\Delta(Y))$, which implies that at least one of $\Delta(X \cap Y), \Delta(X \cup Y)$ must have weight $< c$. Hence we have a contradiction and the lemma holds. ◀

We will additionally need the following theorem which follows from Theorem 5.1 in [5].

▶ **Theorem 30** ([5])**.** *Let $G = (V, w)$ be a complete weighted graph. Then $\mathcal{M}(G)$ is a cross-free family of cuts.*

▶ **Theorem 31.** *Let $n \geq 2$ and $\mathcal{L}$ be a maximal cross-free family of cuts in the n-vertex complete graph. Let A be an $|\mathcal{L}|$-by-$\binom{n}{2}$ matrix whose rows are the vectors $\chi(S)$ for $S \in \mathcal{L}$ and let $z = A^T\mathbf{1}$. Define $w(e) = 2^{-z(e)+1}$ for $e \in [n]^{(2)}$. Then $G = ([n], w)$ is a complete weighted graph with $\mathrm{cdim}(G) = 2n - 3$ and $\mathcal{M}(G) = \mathcal{L}$.*

Proof. It is clear from the definition that $w > 0$ and so defines a complete weighted graph. We will show that $Aw = \mathbf{1}$. By Lemma 29 this shows that the minimum cut weight of G is 1 and so the set of minimum cuts includes $\mathcal{L}$. As w defines a complete weighted graph, by Theorem 30 the set of minimum cuts in G is cross-free and therefore must be exactly $\mathcal{M}(G) = \mathcal{L}$, since $\mathcal{L}$ is maximal. Further, $|\mathcal{L}| = 2n - 3$ by Corollary 25 and the vectors in $\vec{\mathcal{L}}$ are linearly independent by Lemma 27, thus $\mathrm{cdim}(G) = 2n - 3$.

It remains to show $Aw = \mathbf{1}$. We do this using an alternative way of viewing the assignment of edge weights. Let $\mathcal{G} \subseteq 2^{[n]}$ be the beach of $\mathcal{L}$, and (T, ϕ) be a faithful tree representation of $\mathcal{G}$. For vertices $u, v \in V(T)$ let $d(u, v)$ be the length of the shortest path between u, v in the undirected graph underlying T. Now let $\{i, j\} \in [n]^{(2)}$ and suppose $\phi(i) = u, \phi(j) = v$. We claim that $w(\{i, j\}) = 2^{-d(u,v)+1}$. The sets of $\mathcal{G}$ containing i are the sets represented by edges from the root to u; the sets of $\mathcal{G}$ containing j are the sets represented by the edges on the path from the root to v. Therefore the sets which contain i but not j or j but not i, are exactly those represented by the edges on the path from u to v in the undirected tree underlying T. As (T, ϕ) is faithful, each of these edges represents a different set, and therefore the number of edges on the path from u to v is exactly the number of sets of $\mathcal{L}$ which contain $\{i, j\}$.

We now continue with the proof that $Aw = \mathbf{1}$ using this interpretation of the weights. For any cut $S \in \mathcal{L}$ with shore $X \in \mathcal{G}$, take the edge $(u, v) \in E(T)$ representing X. Now imagine we remove the edge (u, v) from T which disconnects T into two components. Let T_u be the component containing u and T_v the component containing v. From T_u, which contains the root r of T, we create a graph T'_u whose underlying undirected graph is the same as T_u, but for which all edges are directed away from u. Thus in T'_u, vertex u becomes the root and r becomes a leaf. Now by item (2) of Lemma 24, every non-leaf vertex in T_v and T'_u has out-degree 2. We inject a unit of flow into u in the graph T'_u and let it propagate according to the rule that at every non-leaf vertex half of the flow is routed along each outgoing edge. We similarly inject a unit of flow into v in the graph T_v and let it propagate according to the same rule. Thus in the tree T_v, each leaf a gets $f(a) = 2^{-d(a,v)}$ amount of flow, where $d(a,v)$ is the number of edges along the path from v to a in T_v. Similarly, if b is a leaf in the tree T'_u, the amount of flow arriving at b is $f(b) = 2^{-d(b,u)}$. Now let $\{i,j\} \in [n]^{(2)}$ with $i \in X, j \in \bar{X}$ and observe that the way we defined $w(\{i,j\})$ satisfies

$$w(\{i,j\}) = 2^{-d(\phi(i),\phi(j))+1} = 2^{-d(\phi(i),v)-d(\phi(j),u)} = f(\phi(i)) \cdot f(\phi(j)) \ .$$

Thus the weight of the cut S is

$$\sum_{i\in X, j\in\bar{X}} w(\{i,j\}) = \sum_{i\in X, j\in\bar{X}} f(\phi(i))\cdot f(\phi(j)) = \left(\sum_{i\in X} f(\phi(i))\right)\cdot\left(\sum_{j\in\bar{X}} f(\phi(j))\right) = 1\cdot 1 = 1 \ . ◀$$

6 Another proof using graph operations

In this section we give another proof of our main theorems: we prove that the cut dimension of any n-vertex graph is at most $2n-3$ and we also prove that this upper bound is tight. An important role will be played by the following lemma, giving an explicit characterization of graphs having at least one non-star mincut, where none of these mincuts is crossless. This characterization has originally appeared in [4, 6]. More modern presentations can be found in Lemma 2.9 of [5] or Lemma 2 of [7].

▶ **Lemma 32.** *Suppose that $G = (V,w)$ is a graph which has a non-star mincut, and every non-star mincut is crossed by a non-star mincut. Then G is a cycle where all edges have the same weight.*

Let us denote by C_n the cycle on the n vertex set $V = \{v_1, \dots, v_n\}$ and with edge set $E = \{\{v_1,v_2\}, \dots, \{v_{n-1},v_n\}, \{v_n,v_1\}\}$, where the weight of every edge is the same. We also need that the cut dimension of C_n is at most n. In fact, it is easy to prove that the its cut dimension is exactly n when $n \geq 3$.

▶ **Lemma 33.** *The cut dimension of C_2 is 1, and* $\mathrm{cdim}(C_n) = n$, *for $n \geq 3$.*

Proof. The statement for $n = 2$ is obvious. For $n \geq 3$ we have $\mathrm{cdim}(C_n) \leq n$ as the graph only has n edges and thus the cut vectors are elements of $\mathbb{R}^n$ which has dimension n.

For the lower bound we construct a set of n linearly independent minimum cut vectors in C_n. Label the coordinates of the vectors by the edges $\{v_1,v_2\}, \dots, \{v_{n-1},v_n\}, \{v_n,v_1\}$. We define the sets $X_1 = \{v_1,v_2\}$ and $X_k = \{v_2,\dots,v_k\}$, for $2 \leq k \leq n$.

We claim that the cut vectors $\xi_k = \chi(\Delta(X_k))$, for $1 \leq k \leq n$, are linearly independent. Let e_i be the i^{th} standard basis vector in $\mathbb{R}^n$. Then we see that $\xi_1 = e_2 + e_n$ and $\xi_k = e_1 + e_k$, for $2 \leq k \leq n$. Thus $\xi_2 + \xi_n - \xi_1 = 2e_1$, so e_1 is in the span of these vectors. Also $e_k = \xi_k - e_1$ is in the span for $2 \leq k \leq n$. Hence these n vectors span all of $\mathbb{R}^n$ and therefore must be linearly independent. ◀

6.1 Two lemmas on graph operations

The main technical part of the second proof of our main theorems is played by the two lemmas in this section. The second lemma gives an upper bound on the cut dimension of a graph G in function of the cut dimension of the smaller graphs obtained when G is separated along a crossless non-star minimum cut Z. Moreover, this upper bound becomes an equality when in addition the cut Z is connected. Our upper and lower bounds for the cut dimension are respectively almost immediate consequences of these results.

▶ **Lemma 34.** *Let $G = (V, w)$ be a weighted graph and let $Z \in \mathcal{M}(G)$ be a crossless non-star minimum cut defined by shores $X_0, X_1 = V \setminus X_0$. For $b \in \{0, 1\}$, let $\mathcal{M}_b = \{S \in \mathcal{M}(G) : S \subseteq Z \cup E(X_b)\}$. Let $\mathrm{sep}(G, Z) = \{G_0 = (V_0, w_0), G_1 = (V_1, w_1)\}$ as defined in Section 2, where $V_b = X_b \cup \{v_{1-b}\}$, for $b \in \{0, 1\}$, with $v_0, v_1 \notin X_0 \cup X_1$. Then $\dim(\mathrm{span}(\vec{\mathcal{M}}_b)) = \mathrm{cdim}(G_b)$, for $b \in \{0, 1\}$.*

Proof. We prove the statement for $b = 0$, the other case follows in exactly the same manner. Let $m = |E|$ and partition E into three disjoint sets $E = E(X_0) \sqcup Z \sqcup E(X_1)$. Call a vertex $x \in X_0$ *friendly* if it has a neighbor in X_1, that is there exists an edge $\{x, y\} \in Z$ for some $y \in X_1$. The edges in Z can then be partitioned into the disjoint union of sets Z_x, over all friendly x, where $Z_x = \{e \in Z : x \in e\}$.

Let $\mathcal{M}(G_0)$ be the set of all minimum cuts of G_0. The set $\vec{\mathcal{M}}(G_0)$ is composed of m_0 dimensional vectors where $m_0 = |E(X_0)| + \deg(v_1)$. Observe that $\deg(v_1)$ is the number of friendly vertices in X_0. We can partition the edges of G_0 into two sets $E(X_0) \sqcup Z_1$ where $Z_1 = \{\{x, v_1\} : x \text{ is friendly}\}$.

We define a natural bijection $\psi : \mathcal{M}_0 \to \mathcal{M}(G_0)$ as follows. Let S be a mincut in $\mathcal{M}_0$ with shores X' and $V \setminus X'$, where $X' \subsetneq X_0$. Note that we can assume this because Z is crossless. Then $\psi(S)$ is the mincut in $\mathcal{M}(G_0)$ whose shores are X' and $(X_0 \setminus X') \cup \{v_1\}$. Let $k = |\mathcal{M}_0| = |\mathcal{M}(G_0)|$.

We now consider two matrices C and D, where C is a k-by-m matrix and D is a k-by-m_0 matrix. Fix an ordering $S_1, \ldots, S_k$ of $\mathcal{M}_0$ and let the i^{th} row of C be $\chi(S_i)$, the characteristic vector of the cut S_i. Likewise the i^{th} row of D is $\chi(\psi(S_i))$. We have $\mathrm{rk}(C) = \dim(\mathrm{span}(\vec{\mathcal{M}}_0))$ and $\mathrm{rk}(D) = \mathrm{cdim}(G_0)$.

The columns of C, D are labeled by edges. For C, we label the edges according to the partition $E = E(X_0) \sqcup Z \sqcup E(X_1)$, with edges in $E(X_0)$ coming first, then edges from Z, then edges from $E(X_1)$. For D, we label the edges according to the partition $E(X_0) \sqcup Z_1$, again with edges from $E(X_0)$ coming first and then those from Z_1. We observe the following facts:

- The edges in $E(X_0)$ are common in G and G_1, and $\chi(\psi(S_i))(e) = \chi(S_i)(e)$, for every $S_i \in \mathcal{M}_0$ and edge $e \in E(X_0)$. This means that columns of C and D labeled by an edge $e \in E(X_0)$ are identical.
- For an edge $e \in E(X_1)$, we have that $\chi(S_i)(e) = 0$, for every $S_i \in \mathcal{M}_0$. Thus columns of C labeled by an edge $e \in E(X_1)$ are all zero.
- Finally, for a friendly $x \in X_0$ consider any edge $e = \{x, y\} \in Z_x$ and the edge $f = \{x, v_1\} \in Z_1$. Then the e^{th} column of C and the f^{th} column of D are identical because for every $S_i \in \mathcal{M}_0$ we have $\chi(S_i)(e) = 1$ iff $x \in X'$ iff $\chi(\psi(S_i))(f) = 1$.

These points together imply that D is actually a submatrix of C, which can be obtained by taking the columns labeled by edges in $E(X_0)$ and then taking $|Z_1|$ more columns of C by choosing one $e \in Z_x$ for every friendly $x \in X_0$. Therefore $\mathrm{rk}(D) \leq \mathrm{rk}(C)$.

We can also see that $\mathrm{rk}(C) \leq \mathrm{rk}(D)$ as C can be obtained from D by repeating columns labeled by edges in Z_1 several times and adding all zero columns, and neither of these operations increase the rank. ◀

▶ **Lemma 35.** *Let G, Z, G_0, G_1 as in Lemma 34. Then $\operatorname{cdim}(G) \leq \operatorname{cdim}(G_0) + \operatorname{cdim}(G_1) - 1$, and if Z is connected then the equality holds.*

Proof. We first prove that $\operatorname{cdim}(G) \leq \operatorname{cdim}(G_0) + \operatorname{cdim}(G_1) - 1$. The important fact is that $\mathcal{M}(G) \subseteq \mathcal{M}_0 \cup \mathcal{M}_1$ because Z is a crossless mincut. Also since $\mathcal{M}_0, \mathcal{M}_1 \subseteq \mathcal{M}(G)$ we in fact have $\mathcal{M}(G) = \mathcal{M}_0 \cup \mathcal{M}_1$. Therefore

$$\begin{aligned}\operatorname{cdim}(G) &= \dim(\operatorname{span}(\vec{\mathcal{M}}(G)))\\ &= \dim(\operatorname{span}(\vec{\mathcal{M}}_0 \cup \vec{\mathcal{M}}_1))\\ &= \dim(\operatorname{span}(\operatorname{span}(\vec{\mathcal{M}}_0) \cup \operatorname{span}(\vec{\mathcal{M}}_1)))\\ &= \dim(\operatorname{span}(\vec{\mathcal{M}}_0)) + \dim(\operatorname{span}(\vec{\mathcal{M}}_1)) - \dim(\operatorname{span}(\vec{\mathcal{M}}_0) \cap \operatorname{span}(\vec{\mathcal{M}}_1))\\ &= \operatorname{cdim}(G_0) + \operatorname{cdim}(G_1) - \dim(\operatorname{span}(\vec{\mathcal{M}}_0) \cap \operatorname{span}(\vec{\mathcal{M}}_1)) \ .\end{aligned}$$

We use Lemma 34 to obtain the last equality. Notice that $Z \in \mathcal{M}_0 \cap \mathcal{M}_1$, which implies that $\dim(\operatorname{span}(\vec{\mathcal{M}}_0) \cap \operatorname{span}(\vec{\mathcal{M}}_1)) \geq 1$, and thus $\operatorname{cdim}(G) \leq \operatorname{cdim}(G_0) + \operatorname{cdim}(G_1) - 1$.

We now prove the inequality in the reverse direction, when Z is connected. Let $d_b = \operatorname{cdim}(G_b) - 1$, for $b = 0, 1$. Let Z_b be the star cut at v_{1-b} in G_b. Since these are mincuts, we can extend them to a basis in the respective graphs. Therefore there exist $A_1, \ldots A_{d_0} \subset X_0$ and $B_1, \ldots B_{d_1} \subset X_1$ such that the family $\{\chi(\Delta(A_1)), \ldots, \chi(\Delta(A_{d_0})), \chi(Z_0)\}$ is independent in $\operatorname{span}(\vec{\mathcal{M}}(G_0))$ and the family $\{\chi(\Delta(B_1)), \ldots, \chi(\Delta(B_{d_1})), \chi(Z_1)\}$ is independent in $\operatorname{span}(\vec{\mathcal{M}}(G_1)$. We claim that in $\operatorname{span}(\vec{\mathcal{M}}(G))$ the set

$$\{\chi(\Delta(A_1)), \ldots, \chi(\Delta(A_{d_0})), \chi(\Delta(B_1)), \ldots, \chi(\Delta(B_{d_1})), \chi(Z)\}$$

of size $d_0 + d_1 + 1$ is independent.

Let us suppose on the contrary that a non-trivial linear combination of these $d_0 + d_1 + 1$ vectors gives $\mathbf{0}$. Then there exist non all zero real numbers $a_1, \ldots, a_{d_0}, b_1, \ldots, b_{d_1}$ and $\varepsilon \in \{0, 1\}$ such that

$$\sum_{i=1}^{d_0} a_i \chi(\Delta(A_i)) + \sum_{j=1}^{d_1} b_j \chi(\Delta(B_j)) = \varepsilon \chi(Z). \tag{1}$$

We define the function $S : V \to \mathbb{R}$ by

$$S(x) = \begin{cases} \sum_{x \in A_i} a_i & \text{if } x \in X_0,\\ \sum_{x \in B_j} b_j & \text{if } x \in X_1.\end{cases}$$

If $x \in X_0$ and $y \in X_1$ are arbitrary elements and $\{x, y\} \in Z$, then $\chi(\Delta(A_i))(\{x, y\}) = 1$ iff $x \in A_i$ and $\chi(\Delta(B_j))(\{x, y\}) = 1$ iff $y \in B_j$. Therefore for every $\{x, y\} \in Z$, the coordinate $\{x, y\}$ of Equation (1) gives

$$S(x) + S(y) = \varepsilon. \tag{2}$$

From Equation (1) we can also deduce that for every $\{x, x'\} \in E(X_0)$ we have

$$\sum_{i=1}^{d_0} a_i \chi(\Delta(A_i))(\{x, x'\}) = 0, \tag{3}$$

and for every $\{y, y'\} \in E(X_1)$ we have

$$\sum_{j=1}^{d_1} b_j \chi(\Delta(B_j))(\{y, y'\}) = 0. \tag{4}$$

Let $\{x_0, y_0\}$ be an arbitrary edge in Z, where $x_0 \in X_0$ and $y_0 \in X_1$. We set $s_0 = S(x_0)$ and $s_1 = S(y_0)$. We know from Equation (2) that

$$s_0 + s_1 = \varepsilon.$$

We claim that for every $\{x, y\} \in Z$, where $x \in X_0$ and $y \in X_1$, we have $S(x) = s_0$ and $S(y) = s_1$. For this consider an arbitrary breadth first search tree with root x_0. Since the graph of the cut Z, the graph $G(Z) = (V', Z)$, is a connected bipartite graph, every vertex in $V' \cap X_0$ will be at some even depth of the tree, and every vertex in $V' \cap X_1$ at some odd depth of the tree. Going through all the vertices depth by depth starting with x_0 at depth 0, Equation (2) gives the claim.

We now distinguish two cases. In the first case at least one of s_0 and s_1 is non-zero, say without loss of generality that $s_0 \neq 0$. For $i = 1, \dots, d_0$, we define

$$a'_i = a_i / s_0.$$

Then Equation (3) implies that in G_0, for every $\{x, x'\} \in E(X_0)$, we have

$$\sum_{i=1}^{d_0} a'_i \chi(\Delta(A_i))(\{x, x'\}) = 0. \tag{5}$$

Also in G_0, if $x \in X_0$ then $\chi(\Delta(A_i))(\{x, v_1\}) = 1$ iff $x \in A_i$. Therefore

$$\sum_{i=1}^{d_0} a'_i \chi(\Delta(A_i))(\{x, v_1\}) = s_0/s_0 = 1. \tag{6}$$

Therefore Equations (5) and (6) imply that

$$\sum_{i=1}^{d_0} a'_i \chi(\Delta(A_i)) = \chi(Z_0), \tag{7}$$

which contradicts the linear independence of $\{\chi(\Delta(A_1)), \dots, \chi(\Delta(A_{d_0})), \chi(Z_0)\}$.

In the second case $s_0 = s_1 = 0$, and thus for all $\{x, y\} \in Z$, with $x \in X_0$ and $y \in X_1$, we have $S(x) = S(y) = 0$. Therefore in G_0, for every edge $\{x, v_1\}$,

$$\sum_{i=1}^{d_0} a_i \chi(\Delta(A_i))(\{x, v_1\}) = 0, \tag{8}$$

and similarly in G_1, for every edge $\{y, v_0\}$,

$$\sum_{j=1}^{d_1} b_j \chi(\Delta(B_j))(\{y, v_0\}) = 0. \tag{9}$$

Since $a_1, \dots, a_{d_0}, b_1, \dots, b_{d_1}$ are not all zero, either $a_1, \dots, a_{d_0}$ is not all zero or $b_1, \dots, b_{d_1}$ is not all zero. If $a_1, \dots, a_{d_0}$ is not all zero then from Equations (3) and (8) it follows that the family $\{\chi(\Delta(A_1)), \dots, \chi(\Delta(A_{d_0}))\}$ is dependent in $\mathrm{span}(\vec{\mathcal{M}}(G_0))$. If $b_1, \dots, b_{d_1}$ is not all zero then similarly from Equations (4) and (9) it follows that the family $\{\chi(\Delta(B_1)), \dots, \chi(\Delta(B_{d_1}))\}$ is dependent in $\mathrm{span}(\vec{\mathcal{M}}(G_1))$. In either case, we reach a contradiction. ◀

6.2 The upper bound

We now can give our second proof of the upper bound on the cut dimension Theorem 1.

Proof of Theorem 1. The proof is by induction. For the base case $n = 2$, the only graph to be considered consists of a single edge and the cut dimension is $1 = 2n - 3$.

Now let $n \geq 3$, and we assume the inductive hypothesis holds for all graphs on at most $n - 1$ vertices. We consider 3 cases.

Case 1: The graph G has only star mincuts, say at vertices $v_1, \ldots v_k$, for some $1 \leq k \leq n$. As there are only k mincuts here we have $\mathrm{cdim}(G) \leq k \leq n \leq 2n - 3$ for $n \geq 3$.

Case 2: There is a non-star mincut in G, and every non-star mincut is crossed by a non-star mincut. Then by Lemma 32, the graph G is a cycle where the edges have all the same weight. In this case by Lemma 33, we have $\mathrm{cdim}(G) = \mathrm{cdim}(C_n) = n \leq 2n - 3$ for $n \geq 3$.

Case 3 is where we use the induction hypothesis: Suppose that G has a non-star crossless mincut Z with shores X_0 and $X_1 = V \setminus X_0$. Let $|X_0| = k$. Then by Lemma 35 there are graphs G_0, G_1 such that $\mathrm{cdim}(G) \leq \mathrm{cdim}(G_0) + \mathrm{cdim}(G_1) - 1$, where G_0 is a graph on $k + 1$ vertices, and G_1 is a graph on $n - k + 1$ vertices. Therefore by the inductive hypothesis

$$\mathrm{cdim}(G) \leq 2(k + 1) - 3 + 2(n - k + 1) - 3 - 1 = 2n - 3 \ . \qquad ◀$$

6.3 The lower bound

We now give our second proof of Theorem 2 that for every $n \geq 2$ there exist graphs G with $\mathrm{cdim}(G) = 2n - 3$. We need a slightly more detailed statement for the inductive hypothesis which is given in the following theorem.

▶ **Theorem 36.** *For every integer $n \geq 2$, there is a complete weighted graph $G = (V, w)$ on n vertices with cut dimension $2n - 3$ and minimum cut weight 1, and where for every $v \in V$, the star cut $\Delta(\{v\})$ is a minimum cut.*

Proof. For $n = 2$ the statement is satisfied by the graph consisting of a single edge of weight one which has cut dimension one and where the two star cuts are minimum cuts. For $n = 3$ we may take the complete graph $G^{(3)} = (V^{(3)}, w^{(3)})$ with all weights $1/2$, which has cut dimension 3.

Now assume that there exists a graph $G^{(n-1)} = (V^{(n-1)}, w^{(n-1)})$ on $n - 1$ vertices satisfying the inductive hypothesis. Let us consider a copy of $G^{(3)} = (V^{(3)}, w^{(3)})$ where $V^{(3)} = \{t, u, v_0\}$ and $V^{(n-1)} \cap V^{(3)} = \emptyset$. We choose $v_1 \in V^{(n-1)}$ arbitrarily. We claim that the n-vertex graph $G^n = (V^{(n)}, w^{(n)})$ defined as $\mathrm{mer}(\{(G^{(n-1)}, v_1), (G^{(3)}, v_0)\})$ satisfies the statement. It follows from the definition of the merge operation that $G^{(n)}$ is a complete weighted graph and that its star cuts are of weight one. In addition Claim 8 asserts that if Z is the cut in G^n whose shores are $V^{(n-1)} \setminus \{v_1\}$ and $V^{(3)} \setminus \{v_0\}$ then $w(Z) = 1$.

We now claim that the weight of a minimum cut of $G^{(n)}$ is one and that the mincut Z is crossless. Consider a non-star cut $\Delta(X)$. If both vertices t, u are on the same shore then the weight of $\Delta(X)$ is the same as the analogous cut in $G^{(n-1)}$ and therefore is at least one. If $\Delta(X)$ crosses Z, then we suppose without loss of generality that $t \in X, u \in \bar{X}$. We show that the weight of $\Delta(X)$ is greater than one, which then implies both claims. The cut contains the edge $\{t, u\}$ which has weight $1/2$. For every $y \in V^{(n-1)} \setminus \{v_1\}$, the cut either contains the edge $\{t, y\}$ or the edge $\{u, y\}$, and these edges have the same weight. Thus the total weight of such edges is half of the weight of Z, that is $1/2$. In addition, the cut contains also at least one edge from $G^{(n-1)}$, therefore its total weight is greater than one.

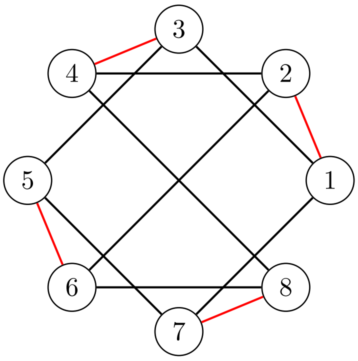


Figure 1 Example graph G showing the necessity of the connected condition in Lemma 34. Red edges have weight 2 and black edges have weight 1. The minimum cut weight is 4 and the cuts achieving this are all the star cuts and $\Delta(\{1,2\}), \Delta(\{3,4\}), \Delta(\{5,6\}), \Delta(\{7,8\}), \Delta(\{1,2,3,4\})$.

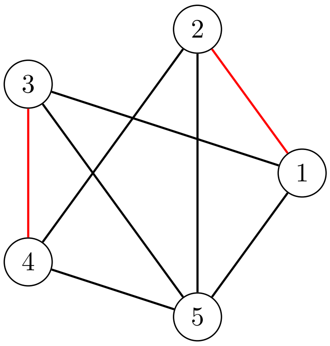


Figure 2 The graph G_0. Red edges have weight 2 and black edges have weight 1. The minimum cut weight is 4 and the cuts achieving this are all the star cuts and $\Delta(\{1,2\}), \Delta(\{3,4\})$. The cut dimension is 7.

Finally Claim 8 says that $\operatorname{sep}(G^{(n)}, Z) = \{G^{(n-1)}, G^{(3)}\}$. Since Z is a crossless non-star minimum cut that is also connected, Lemma 35 implies that $\operatorname{cdim}(G^{(n)}) = \operatorname{cdim}(G^{(n-1)}) + \operatorname{cdim}(G^{(3)}) - 1$, which is $2n - 3$ by the inductive hypothesis. ◀

6.4 On the tightness of Lemma 35

One can wonder whether the connectedness of Z is a necessary hypothesis in Lemma 35. In fact it is, when $Z \in \mathcal{M}(G)$ is not connected then we can have $\operatorname{cdim}(G) < \operatorname{cdim}(G_0) + \operatorname{cdim}(G_1) - 1$. An example is given in Figure 1. The mincuts in this graph are all the star cuts and

$$\Delta(\{1,2\}), \Delta(\{3,4\}), \Delta(\{5,6\}), \Delta(\{7,8\}), \Delta(\{1,2,3,4\}) \ .$$

Thus no mincuts cross each other. Also none of the non-star mincuts are connected.

Consider the case where $Z = \Delta(\{1,2,3,4\})$. When we separate G along this cut we see that $G_0 = G_1$ and they are equal to the graph in Figure 2. The mincuts in G_0 are all star cuts and $\Delta(\{1,2\}), \Delta(\{3,4\})$. All non-star mincuts in G_0 are connected so one can use Lemma 34 to compute that $\operatorname{cdim}(G_0) = 7$, i.e. all these mincut vectors are linearly independent. However, the cut dimension of G is clearly at most 12 as it only has 12 edges. Direct computation shows that in fact $\operatorname{cdim}(G) = 11$.

7 ℓ_1-approximate cut dimension

In this section, we use the ℓ_1-approximate cut dimension method to show Theorem 3 that for any $k \in \mathbb{N}$ and $n = 3k+1$, it holds that $D_{\text{lin}}(\text{MINCUT}_n) \geq 2n-2$.

Let K_4 be the complete graph on 4 vertices with all edge weights equal to 1. The theorem will follow from showing that the ℓ_1-approximate cut dimension of the direct union of k copies of K_4 has ℓ_1-approximate cut dimension $6k$. We start with the base case $k = 1$ to build up the notation and intuition that will be needed for the general case. The following definition and fact will be useful.

▶ **Definition 37** (Strictly diagonally dominant). *Let $A \in \mathbb{R}^{n\times n}$ be a matrix. We say that the i^{th} row of A is* strictly diagonally dominant *if $|A(i,i)| > \sum_{j\neq i} |A(i,j)|$. We say that A is strictly diagonally dominant iff all of its rows are.*

It is well known that a strictly diagonally dominant matrix has full rank. One way to prove this is via the following fact, which we will make use of in the proof of Theorem 3.

▶ **Fact 38.** *Let $A \in \mathbb{R}^{n\times n}$ be a matrix whose i^{th} row is strictly diagonally dominant. If $Au = \mathbf{0}$ for a vector $u \neq \mathbf{0}$ then $|u_i| < \|u\|_\infty$.*

Proof. Suppose for a contradiction that for some $u \neq \mathbf{0}$ it holds that $Au = \mathbf{0}$ and $|u_i| = \|u\|_\infty$ where the i^{th} row of A is strictly diagonally dominant. By normalizing and flipping the sign of u if necessary we may assume $\|u\|_\infty = 1$ and $A(i,i)u_i = |A(i,i)|$. Thus

$$\sum_j A(i,j)u_j = |A(i,i)| + \sum_{j\neq i} A(i,j)u_j \geq |A(i,i)| - \sum_{j\neq i} |A(i,j)| > 0 \ ,$$

a contradiction. ◀

7.1 ℓ_1-approximate cut dimension of K_4

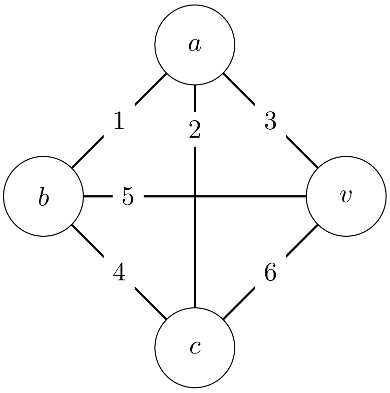


Figure 3 The complete graph on 4 vertices with all edge weights equal to 1. The labels on edges indicate the ordering of edges used to represent cut vectors in the proof.

We label the vertices of K_4 by a, b, c, v, and use the ordering of edges indicated in Figure 3. Let X be the 7-by-6 matrix whose rows correspond to the cut vectors of all the nontrivial cuts

$$X = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} . \tag{10}$$

The cut vectors in X are given in the order

$$\Delta(\{a\}), \Delta(\{b\}), \Delta(\{c\}), \Delta(\{a,b,c\}), \Delta(\{a,b\}), \Delta(\{a,c\}), \Delta(\{b,c\}) \ .$$

The first 4 rows correspond to star cuts which are minimum cuts of weight 3 in K_4. The last three rows correspond to cuts which have weight 4 in K_4. Thus to show a lower bound of 6 on the number of linear queries needed to compute the minimum cut of a 4 vertex graph, we need to show that the $w = (1,1,1,1,1,1), c = (0,0,0,0,1,1,1)$ one-sided ℓ_1 approximate rank of X is 6.

▷ Claim 39. Let $w = \mathbf{1} \in \mathbb{R}^6$, and $c = (0,0,0,0,1,1,1)$. The (w,c) one-sided ℓ_1 approximate rank of X is 6.

Proof. The rank of X at most 6 as this is the number of columns, which takes care of the upper bound.

Now consider the lower bound. To do this we need to lower bound the rank of the matrix

$$Z = X - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ A_1 & A_2 & A_3 \end{bmatrix}$$

where each of $A_1, A_2, A_3 \geq 0$ are 3-by-2 matrices and every row of $A_1 + A_2 + A_3$ sums to at most 1. As the first 4 rows of X correspond to vectors of minimum cuts, no error is allowed on the first 4 rows.

The first 4 rows of Z are equal to the first 4 rows of X, as there is no perturbation allowed on these rows. By doing elementary row operations on the first four rows, which do not change the rank, we can transform the first four rows of Z into the reduced row echelon form of $X(1:4,:)$. Thus we arrive at the following matrix.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ A_1 & A_2 & A_3 \end{bmatrix} .$$

Now we do column operations to zero out the entries in the first four rows and last two columns. For a m-by-2 matrix A we will use the notation A° to denote the matrix A with the order of the columns swapped. We arrive at

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -2 \\ 1 & 0 & 1 & 1 & -2 & 0 \\ 1 & 1 & 0 & 0 & 2 & 2 \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ A_1 & A_2 & A_1^\circ - A_2 - A_2^\circ + A_3 \end{bmatrix} .$$

Finally, we can do row operations to zero out the first four columns in the last three rows.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{4,2} & \mathbf{0}_{4,2} & \mathbf{0}_{4,2} \\ \mathbf{0}_{3,2} & \mathbf{0}_{3,2} & A_1^\circ - A_2 - A_2^\circ + A_3 \end{bmatrix} .$$

The task has now reduced to showing the matrix

$$Z' = \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ -2 & -2 \end{bmatrix} + A_1 - A_2 - A_2^\circ + A_3^\circ$$

has rank 2 for any A_1, A_2, A_3 satisfying the constraints. Let us simplify the matrix $A_1 - A_2 - A_2^\circ + A_3^\circ$. First, let $A_1' = A_1 + A_3^\circ$. Next, note that $D = A_2 + A_2^\circ$ has the property that $D(i,1) = D(i,2)$ for $i \in [3]$. In the sequel we call this the *partner property.*

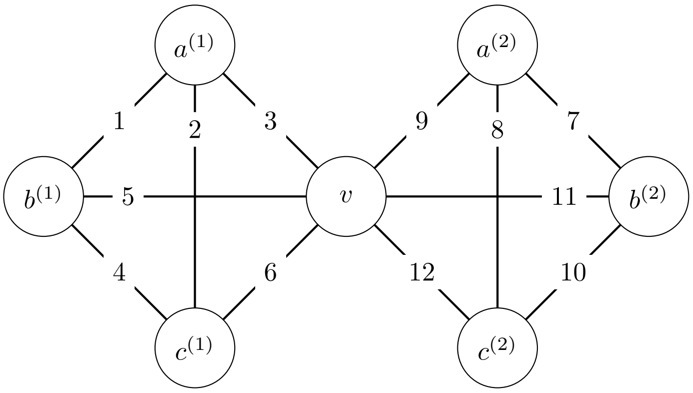
As the row sum of $A_1' + A_2$ is at most 1, unless $A_1'(1:2, 1:2) = \mathbf{0}_{2,2}$ and at least one row sum of $A_2(1:2, 1:2)$ is equal to 1 the first two rows of Z' will be strictly diagonally dominant. If the first two rows of Z' are strictly diagonally dominant then the rank of Z' must be 2, thus we now handle the "unless" case.

First, suppose exactly one row sum of $A_2(1:2, 1:2)$ is equal to 1. Say without loss of generality it is the second one, thus the first row of Z' is strictly diagonally dominant. Then for a sufficiently small ε we can multiply the first column by $1-\varepsilon$ so that the first row remains strictly diagonally dominant and the second row becomes strictly diagonally dominant as well. This does not increase the rank and thus shows again that the rank of Z' is 2.

The remaining case is where both rows of $A_2(1:2, 1:2)$ sum to one. In this case by the partner property we have

$$Z'(1:2, 1:2) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} .$$

On the other hand, the last row of Z' must have both entries ≤ -1. Thus the determinant of the submatrix formed by the first row and the third is strictly negative and so Z' has rank 2. ◁



$$X^{(2)} = \begin{bmatrix} X & \mathbf{0}_{7,6} \\ \mathbf{0}_{7,6} & X \end{bmatrix}$$

Figure 4 Example of the direct union of two copies of K_4. With the ordering of the edges given by the edge labels, the matrix of cut vectors of the cuts $\Delta(\{a^{(i)}\}), \Delta(\{b^{(i)}\}), \Delta(\{c^{(i)}\})$, $\Delta(\{a^{(i)}, b^{(i)}, c^{(i)}\}), \Delta(\{a^{(i)}, b^{(i)}\}), \Delta(\{a^{(i)}, c^{(i)}\}), \Delta(\{b^{(i)}, c^{(i)}\})$ for $i \in [2]$ becomes the matrix $X^{(2)}$ on the right.

7.2 Direct union of K_4 with itself

Now we prove the general case. The key to the proof is the following lemma.

▶ **Lemma 40.** *Let $k \in \mathbb{N}$ and B be the $3k$-by-$2k$ matrix*

$$B = \begin{bmatrix} 2\mathbf{I}_{2k} \\ -2\mathbf{I}_k \otimes [1, 1] \end{bmatrix} .$$

For any matrices $3k$-by-$2k$ matrices A_1, A_2 satisfying the conditions

1. $A_1, A_2 \geq 0$
2. *(partner property) For all $i \in [3k]$ and $j \in [k]$ it holds that $A_2(i, 2j-1) = A_2(i, 2j)$.*
3. *Every row of $A_1 + A_2/2$ sums to at most 1*

it holds that $B + A_1 - A_2$ has rank $2k$.

Proof. The rank is at most $2k$ as that is the number of columns; we focus on showing the columns are linearly independent.

Let $Z = B + A_1 - A_2$. We call the first $2k$ rows of Z rows of type I, and the last k rows of type II. If a type I row is not strictly diagonally dominant, we call it *full.* Notice that a type I row i is full if and only if the i^{th} row of A_1 is zero and the i^{th} row of A_2 sums to 2. In this case, $Z(i, j) \leq 0$ for every $j \neq i$ and it holds that $Z(i, i) = -\sum_{j \neq i} Z(i, j)$. For $i \in [k]$ we call $2i - 1$ and $2i$ *partners.*

Suppose for contradiction there is a vector $\vec{u} \neq \mathbf{0}$ such that $A\vec{u} = \mathbf{0}$. As $\vec{u} \neq \mathbf{0}$ by normalizing and multiplying by -1 as needed we may assume that $\|u\|_\infty = 1$ and i is a coordinate with $\vec{u}(i) = 1$. By Fact 38 the i^{th} row of Z, which is a type I row, cannot be strictly diagonally dominant. Thus the i^{th} row must be full. Therefore for $Z(i, :)\vec{u} = 0$ to hold it must be the case that $\vec{u}(j) = 1$ for every j where $A_2(i, j) > 0$. Such a j must exist as the i^{th} row of A_2 sums to 2. So let j be a coordinate with $A_2(i, j) > 0$ and let j' be the partner of j. By the partner property we also have $A_2(i, j') > 0$ and therefore $\vec{u}(j) = \vec{u}(j') = 1$.

Now consider the type II row ℓ for which $B(\ell, j) = B(\ell, j') = -2$. As $B(\ell, t) = 0$ for $t \notin \{j, j'\}$ this means

$$\begin{aligned} Z(\ell,:)\vec{u} &= Z(\ell,j) + Z(\ell,j') + \sum_{t\notin\{j,j'\}} Z(\ell,t)\vec{u}(t) \\ &\le B(\ell,j) + A_1(\ell,j) + B(\ell,j') + A_1(\ell,j') + \|\vec{u}\|_\infty \sum_{t\notin\{j,j'\}} |Z(\ell,t)| \\ &\le -4 + \sum_t A_1(\ell,t) + A_2(\ell,t) \\ &\le -2 \ , \end{aligned}$$

and we have arrived at a contradiction. ◀

With Lemma 40 in hand we are now ready to prove Theorem 3.

Proof of Theorem 3. Let $G^{(1)}, \dots, G^{(k)}$ be k copies of K_4 where the vertices in $G^{(i)}$ are labeled by $a^{(i)}, b^{(i)}, c^{(i)}, v^{(i)}$ for $i \in [k]$. The graph G is formed by taking the direct union of $G^{(1)}, \dots, G^{(k)}$ at the vertices $v^{(1)}, \dots, v^{(k)}$. That is, the vertices $v^{(1)}, \dots, v^{(k)}$ are all identified by a common vertex denoted v. See Figure 4 for an illustration of the graph for $k = 2$.

The cuts of G we focus on are the $7k$ cuts given by

$$\Delta(\{a^{(i)}\}), \Delta(\{b^{(i)}\}), \Delta(\{c^{(i)}\}), \Delta(\{a^{(i)}, b^{(i)}, c^{(i)}\}), \Delta(\{a^{(i)}, b^{(i)}\}), \Delta(\{a^{(i)}, c^{(i)}\}), \Delta(\{b^{(i)}, c^{(i)}\} \ ,$$

for $i \in [k]$. For any $i \in [k]$ the cuts $\Delta(\{a^{(i)}\}), \Delta(\{b^{(i)}\}), \Delta(\{c^{(i)}\}), \Delta(\{a^{(i)}, b^{(i)}, c^{(i)}\})$ achieve the minimum cut weight of G, which is 3, and the cuts $\Delta(\{a^{(i)}, b^{(i)}\}), \Delta(\{a^{(i)}, c^{(i)}\}), \Delta(\{b^{(i)}, c^{(i)}\}$ have weight 4.

With an ordering of the edges as exemplified in Figure 4, the matrix of cut vectors of these cuts is $X^{(k)} = \mathbf{I}_k \otimes X$, where X is the matrix from Equation (10). In every nonzero block of $X^{(k)}$ the first four rows are minimum cuts with weight 3 and the last 3 rows are cuts with weight 4. Let $c' = (0, 0, 0, 0, 1, 1, 1)$. The theorem will follow from Theorem 13 by showing that the $w = \mathbf{1}_{6k}, c = \mathbf{1}_k \otimes c'$ one-sided ℓ_1 approximate rank of $X^{(k)}$ is $6k$.

To do this, we must show that $X^{(k)} - A$ has rank $6k$ for any matrix $A \ge 0$ which is all zero on any row of $I_k \otimes X$ corresponding to a minimum cut, and where the row sum of A is at most 1 on any row of $I_k \otimes X$ corresponding to a cut of weight 4. In order to make reference to the base case, it will be useful to partition the columns into k blocks of 6 columns, where the i^{th} block is further partitioned into blocks of size 2 represented by the $7k$-by-2 matrices $A_1^{(i)}, A_2^{(i)}, A_3^{(i)}$. In other words, we view A as follows

$$A = \begin{bmatrix} A_1^{(1)} & A_2^{(1)} & A_3^{(1)} & \cdots & A_1^{(k)} & A_2^{(k)} & A_3^{(k)} \end{bmatrix}$$

where each $A_j^{(i)}$ for $j \in [3], i \in [k]$ is a $7k$-by-2 matrix.

As in the base case, we begin by doing Gauss-Jordan elimination on the rows corresponding to mincuts of each X block in $X^{(k)}$. These operations only touch rows corresponding to mincuts where A is zero, thus they do not change A. After these operations we arrive at the matrix $\mathbf{I}_k \otimes X' - A$ where

$$X' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Next, as in the base case, we do column operations to zero out the last two columns in the first four rows of each block of X'. This gives us the matrix $\mathbf{I}_k \otimes X'' - A'$ where

$$X'' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -2 \\ 1 & 0 & 1 & 1 & -2 & 0 \\ 1 & 1 & 0 & 0 & 2 & 2 \end{bmatrix}$$

and the i^{th} block of A' looks like

$$[A_1^{(i)} \mid A_2^{(i)} \mid A_1^{(i)\circ} - A_2^{(i)} - A_2^{(i)\circ} + A_3^{(i)}]. \quad .$$

Here $A_1^{(i)\circ}$ denotes the matrix $A_1^{(i)}$ with the order of the columns swapped. Finally, we use $X''(1:4, 1:4)$ to zero out all other entries of $\mathbf{I}_k \otimes X'' - A'$ in the first 4 columns of each block. This brings us to the matrix $\mathbf{I}_k \otimes X''' - A''$ where

$$X''' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 \end{bmatrix}$$

and the i^{th} block of A'' is

$$[\mathbf{0}_{7k,2} \mid \mathbf{0}_{7k,2} \mid A_1^{(i)\circ} - A_2^{(i)} - A_2^{(i)\circ} + A_3^{(i)}] \quad .$$

Again, each of $A_1^{(i)}, A_2^{(i)}, A_3^{(i)}$ is zero on rows corresponding to minimum cuts. Thus by multiplying the last two columns of each block by -1 and permuting rows and columns we can transform $\mathbf{I}_k \otimes X''' - A''$ into the form

$$\begin{bmatrix} \mathbf{I}_{4k} & \mathbf{0}_{4k,2k} \\ \mathbf{0}_{3k,4k} & B + A_1 - A_2 \end{bmatrix}$$

where B, A_1, A_2 satisfy the conditions of Lemma 40. Thus a rank lower bound of $6k$ follows from the lower bound of $2k$ on the rank of $B + A_1 - A_2$ given in Lemma 40. ◀

8 The dimension of approximate mincuts

Let G be a weighted graph and λ the weight of a minimum cut in G. For $\alpha \geq 1$ define an α-near-mincut of G to be a cut S whose weight is at most $\alpha\lambda$. Let $\mathcal{M}_\alpha(G)$ be the set of all α-near-mincuts of G and $\vec{\mathcal{M}}_\alpha(G) = \{\chi(S) : S \in \mathcal{M}_\alpha(G)\}$. In this section, we look at $\mathrm{cdim}_\alpha(G) = \dim(\mathrm{span}(\vec{\mathcal{M}}_\alpha(G)))$.

The first observation is that if $\alpha = 2$ then the unweighted complete graph K_n satisfies $\mathrm{cdim}_\alpha(K_n) = \binom{n}{2}$. For *simple* graphs we can show $\alpha = 2$ is a sharp threshold.

▶ **Theorem 41.** *Let $1 \leq \alpha < 2$ be a constant and G be a simple n-vertex graph. Then $\mathrm{cdim}_\alpha(G) = O(n)$.*

The key to this theorem is the following lemma of Rubinstein, Schramm, and Weinberg [26].

▶ **Lemma 42** (Lemma 2.6 [26]). *Let G be a simple graph with minimum degree $d_{\min}$ and minimum cut value λ. For constant $0 \leq \epsilon < 1$ let $\mathcal{T}$ be the set of non-star cuts of G whose weight is at most $\lambda + \epsilon d_{\min}$. Then $|\cup_{T \in \mathcal{T}} T| = O(n)$.*

Proof of Theorem 41. Let G be a simple graph. To prove the theorem we create a set of $O(n)$ vectors that span $\vec{\mathcal{M}}_\alpha(G)$. Let $\mathcal{M}_\alpha(G) = \mathcal{T} \sqcup \mathcal{S}$, where $\mathcal{T}$ is the set of non-star cuts of $\mathcal{M}_\alpha(G)$ and $\mathcal{S}$ is the set of star cuts of $\mathcal{M}_\alpha(G)$. Let $E' = \cup_{T \in \mathcal{T}} T$ be the set of edges involved in the cuts in $\mathcal{T}$. Let $\vec{\mathcal{L}} = \{e_i : i \in E'\}$. Note that from the definition of $d_{\min}$, there is a star cut with cut value $d_{\min}$, which implies that $\lambda \leq d_{\min}$. As a result, every α-near-mincut has cut value at most $\alpha\lambda \leq \lambda + (\alpha - 1)d_{\min}$, and hence by Lemma 42 we have $|\vec{\mathcal{L}}| = O(n)$. Also $\mathrm{span}(\vec{\mathcal{T}}) \subseteq \mathrm{span}(\vec{\mathcal{L}})$. Thus $\mathrm{span}(\vec{\mathcal{M}}_\alpha(G)) \subseteq \mathrm{span}(\vec{\mathcal{L}} \cup \vec{\mathcal{S}})$. As $|\mathcal{S}| \leq n$ this is a spanning set of size $O(n)$. ◀

In a previous version of this work we conjectured that for an n-vertex *weighted* graph G it holds that $\mathrm{cdim}_\alpha(G) = O(n)$ for any $\alpha < 4/3$. This turns out to be false, however. The reason is that, on the one hand, in a graph $G = (V, w)$ the characteristic vector of a cut $\chi(S)$ depends only on the set of edges, but not the weight of these edges. On the other hand, $w(S)$ does of course depend on the weight of the edges. We can utilize this difference to construct an example as follows. Let us start with a cycle C_n with all edge weights being 1. While C_n has $\binom{n}{2}$ mincuts with weight 2, these mincuts live in an n-dimensional space as C_n only has n edges. We can then turn C_n into a complete weighted graph G by adding a tiny weight $\varepsilon = 2(\alpha - 1)/\binom{n}{2}$ edge to all pairs of vertices that are not adjacent in the cycle. As adding edges cannot decrease the minimum cut weight, the weight of a minimum cut in G is at least 2. Further, if X is the shore of a minimum cut in C_n then in the graph G we have $w(\Delta(X)) \leq 2 + \binom{n}{2}\varepsilon = 2\alpha$, as the weight is at most its weight in C_n plus the weight of all added edges. Thus $\Delta(X)$ is an α-near-mincut in G. Further, the characteristic vectors $\chi(\Delta(X)) \in \{0,1\}^{\binom{n}{2}}$ of these cuts in G now live in an $\binom{n}{2}$-dimensional space and become linearly independent. This example demonstrates that a reasonable extension of the cut dimension to near-mincuts should take into account the magnitude of the edge weights, as the ℓ_1-approximate cut dimension does.

We now give the formal proof that the graph G mentioned above has the correct properties.

▶ **Lemma 43.** *Let $n \in \mathbb{N}$. Let C_n be the cycle on n vertices and $\mathcal{G}$ the beach of $\mathcal{M}(C_n)$. Let K_n be the complete graph on n vertices. Let $\mathcal{T} = \{\Delta(X) : X \in \mathcal{G}\}$, where here $\Delta(X) \in \{0,1\}^{\binom{n}{2}}$ is the cut in K_n with shore X. Then $\dim(\mathrm{span}(\vec{\mathcal{T}})) = \binom{n}{2}$.*

Proof. For this proof we assume the vertices are labeled by $0, \ldots, n-1$ and use addition modulo n. We will show that all of the standard basis vectors $e_{\{i,j\}}$ are in $\mathrm{span}(\vec{\mathcal{T}})$. For concreteness, we show how to construct the vectors $e_{\{0,j\}}$; by symmetry the same argument can then be used for any $e_{\{i,j\}}$.

We will actually construct the vectors $E_j = \sum_{k=1}^{j} e_{\{0,k\}}$. This suffices as $e_{\{0,j\}} = E_j - E_{j-1}$. First note that $e_{\{0,1\}} = \frac{1}{2}(\chi(\Delta(\{0\}) + \chi(\Delta(\{1\})) - \chi(\Delta(\{0,1\})))$, and thus is in $\mathrm{span}(\vec{\mathcal{T}})$ as all the vectors on the right hand side are in $\vec{\mathcal{T}}$.

Now let $j > 1$ and $X = \{1, \ldots, j\}, X' = X \cup \{0\}$. Then

$$\chi(\Delta(X))(e) - \chi(\Delta(X'))(e) = \begin{cases} 1 & \text{if } e = \{0,k\}, k \in X \\ -1 & \text{if } e = \{0,k\}, k \in \bar{X}' \\ 0 & \text{otherwise} \end{cases} .$$

Thus $E_j = \frac{1}{2}(\Delta(\{0\} + \chi(\Delta(X)) - \chi(\Delta(X')))$. ◀

▶ **Theorem 44.** *Let $n \in \mathbb{N}$. For any $\alpha > 1$ there exists a graph $G = (\{0, \ldots, n-1\}, w)$ such that* $\operatorname{cdim}_\alpha(G) = \binom{n}{2}$.

Proof. We again use addition modulo n on the labels of the vertices. Let $\varepsilon = 2(\alpha - 1)/\binom{n}{2}$. Define $w(\{i, i+1\}) = 1$ for $i \in \{0, \ldots, n-1\}$ and for any other i, j let $w(\{i, j\}) = \varepsilon$. Let $G = (\{0, \ldots, n-1\}, w)$. Thus G is the graph of the cycle C_n with edges of weight ε added between all pairs of vertices that are not adjacent in the cycle. The weight of a minimum cut of G is at least that of C_n, which is 2, as adding edges cannot decrease the weight of a cut. Further, if X is the shore of a minimum cut in C_n then in the graph G we have $w(\Delta(X)) \leq 2 + \binom{n}{2}\varepsilon = 2\alpha$, as the weight is at most its weight in C_n plus the weight of all added edges. Thus $\Delta(X)$ is an α-near-mincut in G and $\operatorname{cdim}_\alpha(G)$ is at least $\binom{n}{2}$ by Lemma 43. It also clearly cannot be larger than $\binom{n}{2}$ and so the theorem is proved. ◀

References

1 Sepehr Assadi, Deeparnab Chakrabarty, and Sanjeev Khanna. Graph connectivity and single element recovery via linear and OR queries. *CoRR*, abs/2007.06098, 2020. arXiv:2007.06098.

2 László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347, 1986. doi:10.1109/SFCS.1986.15.

3 András A. Benczúr and Michel X. Goemans. Deformable polygon representations and near-mincuts. In Martin Grøtschel and Gyula O. H. Katona, editors, *Building Bridges: Between Mathematics and Computer Science*, volume 19 of *Bolyai Society Mathematical Studies*, pages 103–135. Springer, 2008.

4 R. E. Bixby. The minimum number of edges and vertices in a graph with edge connectivity n and m n-bonds. *Netw.*, 5(3):253–298, 1975. doi:10.1002/net.1975.5.3.253.

5 L. Sunil Chandran and L. Shankar Ram. On the number of minimum cuts in a graph. *SIAM J. Discret. Math.*, 18(1):177–194, 2004. doi:10.1137/S0895480103427138.

6 Efim A. Dinitz, Alexander V. Karzanov, and Michael V. Lomonosov. On the structure of the system of minimum edge cuts of a graph. *Studies in discrete optimization*, 1976.

7 Tamás Fleiner and András Frank. A quick proof for the cactus representation of mincuts. *EGRES Quick Proof*, 2009-03, 2009.

8 Pawel Gawrychowski, Shay Mozes, and Oren Weimann. Minimum cut in $O(m \log^2 n)$ time. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 57:1–57:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.57.

9 Pawel Gawrychowski, Shay Mozes, and Oren Weimann. A note on a recent algorithm for minimum cut. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 74–79. SIAM, 2021. doi:10.1137/1.9781611976496.8.

10 Mohsen Ghaffari, Krzysztof Nowicki, and Mikkel Thorup. Faster algorithms for edge connectivity via random 2-out contractions. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1260–1279. SIAM, 2020. doi:10.1137/1.9781611975994.77.

11 Michel X. Goemans. Minimum bounded degree spanning trees. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 273–282. IEEE Computer Society, 2006. doi:10.1109/FOCS.2006.48.

12 Michel X. Goemans and V. S. Ramakrishnan. Minimizing submodular functions over families of sets. *Comb.*, 15(4):499–513, 1995. doi:10.1007/BF01192523.

13 Andrei Graur, Tristan Pollner, Vidhya Ramaswamy, and S. Matthew Weinberg. New query lower bounds for submodular function minimization. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 64:1–64:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.ITCS.2020.64`.

14 András Hajnal, Wolfgang Maass, and György Turán. On the communication complexity of graph properties. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 186–191, 1988. `doi:10.1145/62212.62228`.

15 Nicholas J. A. Harvey. *Matchings, matroids and submodular functions*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2008. URL: `http://hdl.handle.net/1721.1/44416`.

16 Monika Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. *SIAM J. Comput.*, 49(1):1–36, 2020. `doi:10.1137/18M1180335`.

17 Monika Rauch Henzinger and David P. Williamson. On the number of small cuts in a graph. *Inf. Process. Lett.*, 59(1):41–44, 1996. `doi:10.1016/0020-0190(96)00079-8`.

18 Kamal Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Comb.*, 21(1):39–60, 2001. `doi:10.1007/s004930170004`.

19 David R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas, USA*, pages 21–30. ACM/SIAM, 1993. URL: `http://dl.acm.org/citation.cfm?id=313559.313605`.

20 David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, 2000. `doi:10.1145/331605.331608`.

21 Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic edge connectivity in near-linear time. *J. ACM*, 66(1):4:1–4:50, 2019. `doi:10.1145/3274663`.

22 Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2018.

23 László Lovász. *Combinatorial problems and exercises (2. ed.)*. North-Holland, 1993.

24 Sagnik Mukhopadhyay and Danupon Nanongkai. Weighted min-cut: sequential, cut-query, and streaming algorithms. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 496–509, 2020.

25 Hiroshi Nagamochi, Kazuhiro Nishimura, and Toshihide Ibaraki. Computing all small cuts in an undirected network. *SIAM J. Discret. Math.*, 10(3):469–481, 1997. `doi:10.1137/S0895480194271323`.

26 Aviad Rubinstein, Tselil Schramm, and S. Matthew Weinberg. Computing exact minimum cuts without knowing the graph. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 39:1–39:16, 2018. `doi:10.4230/LIPIcs.ITCS.2018.39`.

A Jain's spanning lemma

In this appendix we prove Lemma 19. The proof uses the following key property of mincuts which goes back at least to work of Dinitz, Karzanov, and Lomonosov [6].

▶ **Proposition 45** ([6] "Lemma on a quadrangle"). *Let $G = (V, w)$ be a graph. For any crossing mincuts $\Delta(X), \Delta(Y)$ of G it holds that*

$$\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y)) \ .$$

Proof. If $\Delta(X), \Delta(Y)$ cross then $\Delta(X \cap Y), \Delta(X \cup Y)$ are mincuts of G by Claim 5. Further, by counting the number of times an edge appears on each side it can be seen (eg. Ex. 6.48 in [23]) that

$$\chi(\Delta(X)) + \chi(\Delta(Y)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y)) + 2\chi(E(X - Y, Y - X)) \ . \tag{11}$$

Let the minimum cut value of G be λ. Let m be the number of edges in G and $\vec{w} \in \mathbb{R}^m$ be the positive vector resulting from restricting w to the edges of G. The inner product of $\vec{w}$ with the left hand side of Equation (11) is 2λ, and with the righthand side is $2\lambda + 2\langle \vec{w}, \chi(E(X-Y, Y-X))\rangle$. Thus $\langle \vec{w}, \chi(E(X-Y, Y-X))\rangle = 0$, which implies $\chi(E(X-Y, Y-X)) = \mathbf{0}$ since $\vec{w}$ is positive and $\chi(E(X-Y, Y-X))$ is nonnegative. ◀

Jain's proof uses the technique of *combinatorial uncrossing.* Recall the definition of $\mathrm{overlap}_{\mathcal{G}}(X)$ from Definition 28. A key to the proof is the following simple lemma about $\mathrm{overlap}_{\mathcal{G}}(X)$.

▶ **Lemma 46** ([18]). *Let $\mathcal{F} \subseteq 2^V$ be a set family closed under overlaps and $\mathcal{G} \subseteq \mathcal{F}$ be a maximal laminar subset of $\mathcal{F}$. Then for any $X \in \mathcal{F} - \mathcal{G}$ and $Y \in \mathrm{overlap}_{\mathcal{G}}(X)$*

$$\mathrm{overlap}_{\mathcal{G}}(X \cap Y) \subset \mathrm{overlap}_{\mathcal{G}}(X) \tag{12}$$

$$\mathrm{overlap}_{\mathcal{G}}(X \cup Y) \subset \mathrm{overlap}_{\mathcal{G}}(X)\ . \tag{13}$$

Proof. In the following we always refer to $\mathrm{overlap}(X)$ with respect to $\mathcal{G}$ and drop the subscript. We first show Equation (12). First note that $Y \in \mathrm{overlap}(X) - \mathrm{overlap}(X \cap Y)$. Thus to show Equation (12) it suffices to show $\mathrm{overlap}(X \cap Y) \subseteq \mathrm{overlap}(X)$. Let $W \in \mathrm{overlap}(X \cap Y)$. We want to show that $W \in \mathrm{overlap}(X)$, i.e. that it cannot be the case that $W \subseteq X, X \subseteq W$, or $X \cap W = \emptyset$. We know that the last one cannot hold because $W \in \mathrm{overlap}(X \cap Y)$ implies $W \cap (X \cap Y) \neq \emptyset$.

Also as $W, Y \in \mathcal{G}$ they do not overlap and thus either $Y \subseteq W, W \subseteq Y$, or $Y \cap W = \emptyset$. Again the last one cannot hold as $W \cap (X \cap Y) \neq \emptyset$. The following table shows that assuming $W \notin \mathrm{overlap}(X)$ leads to a contradiction in all 4 remaining cases.

	$Y \subseteq W$	$W \subseteq Y$
$W \subseteq X$	$Y \subseteq X$ $Y \notin \mathrm{overlap}(X)$	$W \subseteq X \cap Y$ $W \notin \mathrm{overlap}(X \cap Y)$
$X \subseteq W$	$X \cap Y \subseteq W$ $W \notin \mathrm{overlap}(X \cap Y)$	$X \subseteq Y$ $Y \notin \mathrm{overlap}(X)$

We now show Equation (13), which follows similarly. Again $Y \in \mathrm{overlap}(X) - \mathrm{overlap}(X \cup Y)$ thus it suffices to show $\mathrm{overlap}(X \cup Y) \subseteq \mathrm{overlap}(X)$. Let $W \in \mathrm{overlap}(X \cup Y)$. We want to show that $W \in \mathrm{overlap}(X)$, i.e. that is not the case that either $W \cap X = \emptyset, X \subseteq W$, or $W \subseteq X$. We cannot have $W \subseteq X$ because this means $W \subseteq X \cup Y$ which contradicts $W \in \mathrm{overlap}(X \cup Y)$. As $W, Y \in \mathcal{G}$ they do not overlap, so we also know either $Y \subseteq W, W \cap Y = \emptyset$, or $W \subseteq Y$. The last one again cannot hold as it implies $W \subseteq X \cup Y$. The following table shows that assuming $W \notin \mathrm{overlap}(X)$ leads to a contradiction in the remaining 4 cases.

	$Y \subseteq W$	$W \cap Y = \emptyset$
$X \subseteq W$	$X \cup Y \subseteq W$ $W \notin \mathrm{overlap}(X \cup Y)$	$X \cap Y = \emptyset$ $Y \notin \mathrm{overlap}(X)$
$X \cap W = \emptyset$	$Y \cap X = \emptyset$ $Y \notin \mathrm{overlap}(X)$	$W \cap (X \cup Y) = \emptyset$ $W \notin \mathrm{overlap}(X \cup Y)$

◀

We are now ready to show the key lemma of Jain.

▶ **Lemma 19** ([18]). *Let $G = (V, w)$ be a graph and $\mathcal{L} \subseteq \mathcal{M}(G)$ be a maximal cross-free family of mincuts. Then $\mathrm{span}(\vec{\mathcal{L}}) = \mathrm{span}(\vec{\mathcal{M}}(G))$.*

Proof. It is clear that $\mathrm{span}(\vec{\mathcal{L}}) \subseteq \mathrm{span}(\vec{\mathcal{M}}(G))$ so we focus on the other direction.

Let $\mathcal{F}$ be the beach of $\mathcal{M}(G)$. By Claim 5 $\mathcal{F}$ is closed under overlaps. Let $\mathcal{G} \subseteq \mathcal{F}$ be the beach of $\mathcal{L}$. As $\mathcal{L}$ is a maximal cross-free subset of $\mathcal{M}(G)$ it follows that $\mathcal{G}$ is a maximal laminar subset of $\mathcal{F}$. Thus $|\mathrm{overlap}_{\mathcal{G}}(X)| \geq 1$ for all $X \in \mathcal{F} - \mathcal{G}$. In the following we will always refer to $\mathrm{overlap}(X)$ with respect to $\mathcal{G}$ and drop the subscript.

Suppose for a contradiction that $\mathrm{span}(\vec{\mathcal{L}})$ is a strict subset of $\mathrm{span}(\vec{\mathcal{M}}(G))$. Let

$$X = \underset{Z \in \mathcal{F}-\mathcal{G}}{\mathrm{argmin}}\{|\mathrm{overlap}(Z)| : \chi(\Delta(Z)) \notin \mathrm{span}(\vec{\mathcal{L}})\} \ .$$

As $\mathrm{overlap}(X) \geq 1$, let $Y \in \mathrm{overlap}(X)$. By Lemma 46

$$|\mathrm{overlap}(X \cap Y)| < |\mathrm{overlap}(X)| \tag{14}$$
$$|\mathrm{overlap}(X \cup Y)| < |\mathrm{overlap}(X)| \ . \tag{15}$$

By the definition of X, and as $\mathcal{F}$ is closed under overlaps, we must have $\chi(\Delta(X \cap Y)), \chi(\Delta(X \cup Y)) \in \mathrm{span}(\vec{\mathcal{L}})$. Also as $Y \in \mathcal{G}$ we have $\chi(\Delta(Y)) \in \vec{\mathcal{L}}$ which implies by Proposition 45 that

$$\chi(\Delta(X)) = \chi(\Delta(X \cap Y)) + \chi(\Delta(X \cup Y)) - \chi(\Delta(Y)) \ .$$

This implies $\chi(\Delta(X)) \in \mathrm{span}(\vec{\mathcal{L}})$, a contradiction. ◀

On p-Group Isomorphism: Search-To-Decision, Counting-To-Decision, and Nilpotency Class Reductions via Tensors

Joshua A. Grochow
Departments of Computer Science and Mathematics, University of Colorado Boulder, CO, USA

Youming Qiao
Centre for Quantum Software and Information, University of Technology Sydney, Australia

Abstract

In this paper we study some classical complexity-theoretic questions regarding Group Isomorphism (GpI). We focus on p-groups (groups of prime power order) with odd p, which are believed to be a bottleneck case for GpI, and work in the model of matrix groups over finite fields. Our main results are as follows.

- Although search-to-decision and counting-to-decision reductions have been known for over four decades for Graph Isomorphism (GI), they had remained open for GpI, explicitly asked by Arvind & Torán (*Bull. EATCS*, 2005). Extending methods from Tensor Isomorphism (Grochow & Qiao, ITCS 2021), we show moderately exponential-time such reductions within p-groups of class 2 and exponent p.
- Despite the widely held belief that p-groups of class 2 and exponent p are the hardest cases of GpI, there was no reduction to these groups from *any* larger class of groups. Again using methods from Tensor Isomorphism (*ibid.*), we show the first such reduction, namely from isomorphism testing of p-groups of "small" class and exponent p to those of class *two* and exponent p.

For the first results, our main innovation is to develop linear-algebraic analogues of classical graph coloring gadgets, a key technique in studying the structural complexity of GI. Unlike the graph coloring gadgets, which support restricting to various subgroups of the symmetric group, the problems we study require restricting to various subgroups of the general linear group, which entails significantly different and more complicated gadgets. The analysis of one of our gadgets relies on a classical result from group theory regarding random generation of classical groups (Kantor & Lubotzky, *Geom. Dedicata*, 1990). For the nilpotency class reduction, we combine a runtime analysis of the Lazard Correspondence with Tensor Isomorphism-completeness results (Grochow & Qiao, *ibid.*).

2012 ACM Subject Classification Computing methodologies → Algebraic algorithms; Theory of computation → Problems, reductions and completeness

Keywords and phrases group isomorphism, search-to-decision reduction, counting-to-decision reduction, nilpotent group isomorphism, p-group isomorphism, tensor isomorphism

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.16

Related Version This paper is based on part of the following preprint:
Previous Version: https://arxiv.org/abs/1907.00309

Funding *Joshua A. Grochow*: Partially supported during the preparation of this work by NSF Grants DMS-1750319 and CCF-2047756.
Youming Qiao: Partially supported during the preparation of this work by NSF Grant DMS-1750319 and Australian Research Council Grant DP200100950.

Acknowledgements The authors would like to thank James B. Wilson for related discussions, and Ryan Williams for pointing out the problem of distinguishing between ETH and #ETH. J. A. G. would like to thank V. Futorny and V. V. Sergeichuk for their collaboration on the related work [28]. Ideas leading to this work originated from the 2015 workshop "Wildness in computer science, physics, and mathematics" at the Santa Fe Institute.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 16; pp. 16:1–16:38
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

In this paper, we study the algorithmic problem of deciding whether two finite groups are isomorphic, known as the Group Isomorphism problem (GpI). Different variants of the GpI problem arise, with correspondingly different complexities, when the groups are given in different ways, e.g. by a generating set of permutations, a generating set of matrices, a full multiplication table, or a black box oracle. In its various incarnations, GpI is a fundamental problem in computational algebra and computational complexity. The generator-enumerator algorithm solves isomorphism in $|G|^{\log |G|+O(1)}$-time [26,58][1], and even the current state of the art for general groups – in any of the aforementioned input models – is still $|G|^{\Theta(\log |G|)}$ [9,10,17,25,49,66,70]. Nonetheless, over the past 15 years there has been significant progress on efficient isomorphism tests in various classes of groups: here is an incomplete list of references [4–6,12,13,15,30,31,47,48,63,65,66].

When given by multiplication tables, GpI reduces to GI [44], and in the other, more realistic (for computer algebra systems) and more succinct models, we get a reduction in the other direction [32,34,52,57]. As a result, the techniques and complexity of GpI are closely bound up with GI. However, since the techniques used in GpI are often independent of the input model, we are free to focus on the abstract structure of the groups in question, and the choice of input model is then essentially just a choice of how we measure and report the running time. For example, if GI is in P, then GpI can be solved in poly($|G|$) time; if GpI for groups given by a generating set of m matrices of size $n \times n$ over $\mathbb{F}_p$ can be solved in $p^{O(n+m)}$ time, then GI is in P.

For GI, a wide variety of algorithmic and structural complexity results are known (see, e.g., [3,33,44]). In particular, there are polynomial-time search-to-decision and counting-to-decision reductions [54], so search, counting, and decision are all equivalent for GI. (This was an early piece of evidence that GI was not likely to be NP-complete, since for NP-complete problems, their counting variants are typically #P-complete, hence at least as hard as all of PH [68].) For GpI, no such reductions are known, even in restricted classes of groups; Arvind and Torán [2, Problem 16] explicitly asked for such reductions. Additionally, for GI, there are many classes of graphs for which the isomorphism problem remains GI-complete – such as graphs of diameter 2 and radius 1, directed acyclic graphs, regular graphs, line graphs, polytopal graphs [74] – but no such analogous results are known for GpI.

In this paper, we make progress on all three of these questions, within the class of groups widely believed to be hardest cases of GpI, namely the p-groups of nilpotency class 2 and exponent p; these are groups of order a power of the prime p, such that G modulo its center is abelian, and such that $g^p = 1$ for all $g \in G$. (Throughout most of this paper we assume p is an odd prime.) For each of our three main results, we now give further motivation before stating it formally.

1.1 Main results

Search-to-decision reductions. The "decision versus search" question is a classical one in complexity theory, having attracted the attention of researchers since the introduction of NP. Efficient search-to-decision reductions for SAT and GI are now standard. Valiant first showed the existence of an NP *relation* for which search does not reduce to decision in polynomial time [69]. A celebrated result of Bellare and Goldwasser shows that, assuming

[1] Miller [58] attributes this algorithm to Tarjan.

$\mathsf{DTIME}(2^{2^{O(n)}}) \neq \mathsf{NTIME}(2^{2^{O(n)}})$, there exists an NP *language* for which search does not reduce to decision in polynomial time [8]. However, as usual for such statements based on complexity-theoretic assumptions, the problems constructed by such a proof are considered somewhat unnatural, and natural problems for which search seems not reducible to decision are rare. The most famous candidate may be Factoring (with the decision version being Primality)[2] and Nash Equilibrium [18] (the decision version is trivial).

▶ **Theorem A.** *Let p be an odd prime, and let* GpIso2Exp(p) *denote the isomorphism problem for p-groups of class 2 and exponent p in the model of matrix groups over $\mathbb{F}_p$. For groups of order p^n, there is a search-to-decision reduction for* GpIso2Exp(p) *running in time $p^{O(n)} = \mathrm{poly}(|G|)$.*

▶ Remark 1. This runtime is really only square-root (*moderately*) exponential: The running time of the best-known algorithm for GpIso2Exp(p) is essentially $p^{\Theta(n^2)}$, and the best-known witness size, if we think in terms of nondeterministic algorithms, is $\Theta(n^2)$ [50]. So our search-to-decision reduction in time $p^{O(n)}$ is akin to having such a reduction running in time $2^{\Theta(\sqrt{N})}$ for a problem that is solvable in $2^{\Theta(N)}$ time (resp., has witness size $\Theta(N)$).

We note that that GpIso2Exp(p) seems different from all the problems listed above in terms of search-to-decision reductions, in the following ways. First, unlike SAT and GI, a polynomial-time search-to-decision reduction has been open for decades, whereas those for SAT and GI are straightforward. Note that a polynomial-time reduction would need to run in time $\mathrm{poly}(n, \log p)$, and we find it unlikely that the time complexity of our reduction can be brought down this far with current techniques. Second, unlike Factoring and Nash Equilibrium, whose decision versions are computationally easy, its decision version also seems to require deeper techniques. Indeed, it is a long-standing open problem to test isomorphism of p-groups of class 2 and exponent p in time polynomial in the group order, which already can be exponential in the input size if the input is given by a generating set of matrices.

Counting-to-decision reductions. Counting-to-decision reductions are also of great interest in complexity theory. An efficient counting-to-decision reduction for GI is also a well-known result [54]. In contrast, for SAT, a polynomial-time counting-to-decision reduction would imply that PH collapses [68].

▶ **Theorem B.** *For p an odd prime, $p \geq n^{\Omega(1)}$, there is a randomized counting-to-decision reduction for* GpIso2Exp(p) *for groups of order p^n, running in time $p^{O(n)} = \mathrm{poly}(|G|)$.*

As with Theorem A, the runtime here is only moderately exponential, see Remark 1.

Also as in the case of search-to-decision, GpIso2Exp(p) seems different from the problems listed above in terms of reducing counting to decision. First, a polynomial-time counting-to-decision reduction for GpIso2Exp(p) remains open after 40 years, whereas the reduction for GI was found within the first decade of the rise of computational complexity theory. Second, unlike SAT, for which there have been no non-trivial algorithms to reduce exact counting to decision, we show a moderately exponential-time algorithm for GpIso2Exp(p). As Ryan Williams pointed out to us, asking for the existence of subexponential-time counting-to-decision reduction for SAT seems to lead to asking for the relation between the decision [35] and the counting [22] versions of the Exponential Time Hypothesis.

[2] Here we are thinking of Factoring as the search problem corresponding to the relation $\{(n, d) : d \text{ is a proper divisor of } n\} \subseteq \mathbb{N} \times \mathbb{N}$, so that the existence problem is then precisely Primality.

Nilpotency class reduction. Unlike the case of Graph Isomorphism, for GpI essentially the only class of groups for which isomorphism is known to be as hard as the general case are those which are directly indecomposable, that is, they cannot be written as a direct product $A \times B$ with both A, B nontrivial [42, 72, 73]. However, this result is the group analogue of saying that isomorphism of connected graphs is GI-complete, so although useful (and much less trivial than in the case of graphs vs connected graphs), from a structural perspective it is more like a zero-th step.

For a variety of reasons (e. g., [29]), p-groups of nilpotency class 2 and exponent p are widely believed to be the hardest cases of GpI, but to date there is no known reduction from isomorphism in *any* larger class of groups to this class. The Tensor Isomorphism-completeness of testing isomorphism in this class of groups (when given by generating matrices over $\mathbb{F}_p$) suggests an additional reason for hardness [32] (see also Section 6.1). Here, we leverage that completeness result to give a reduction within GpI itself. While it falls short of being GpI-complete (equivalent to GpI), this is the first such reduction that we are aware of.

To state our result, we need to first recall the definition of nilpotency class. We will give an inductive definition: a group G is nilpotent of class 1 if it is abelian, and nilpotent of class $c > 1$ if $G/Z(G)$ (G modulo its center) is nilpotent of class $c - 1$. Recall that a finite group is nilpotent iff it is the direct product of its Sylow p-subgroups, so from the comment above, isomorphism of nilpotent groups is polynomial-time equivalent to isomorphism of p-groups (for varying p).

▶ **Theorem P.** *Let p be an odd prime. For groups given by generating sets of m matrices of size $n \times n$ over $\mathbb{F}_{p^e}$,* Group Isomorphism *for p-groups of exponent p and class $c < p$ reduces to* Group Isomorphism *for p-groups of exponent p and class 2 in time* $\mathrm{poly}(n, m, e \log p)$.

In fact, because the Lazard Correspondence works whenever all subgroups generated by 3 elements have nilpotency class $< p$, our reduction also works in this more general setting. For example, as a consequence of Theorem P, testing isomorphism of 5-groups in which every 3-generated subgroup has class 4 (the groups themselves may have larger class) reduces to testing isomorphism of 5-groups of class 2 in the matrix group model over fields of characteristic 5.

▶ Remark 2. Two additional results would suffice to get the analogous result in the Cayley table model. The first is to compute the Lazard Correspondence in the Cayley table model in time $\mathrm{poly}(|G|)$; we thank an anonymous ITCS reviewer for pointing out that this can be achieved by applying the matrix Lazard Correspondence (see Proposition 26) to the left regular representation of the group on itself. The second is to improve the blow-up in the reduction from (Lie) Algebra Isomorphism to 3TI from [28]. Currently this reduction increases the dimension quadratically, which means the size of the group becomes $|G|^{O(\log |G|)}$ after the reduction; instead, we would need a reduction that increases the dimension only linearly.

▶ Remark 3. One may also ask whether our theorems can be combined, in order to get search-to-decision and counting-to-decision reductions for p-groups of class $c < p$ instead of only class 2. We believe this should be approachable, but again the quadratic increase in dimension in reductions, mentioned in the previous remark, gets in the way. The quadratic increase makes the square-root exponential reductions into ordinary exponential reductions, negating any gains.

1.2 Main techniques and proof strategies

All our results are based on the connection with Tensor Isomorphism (TI) [32]. Let $\Lambda(n, \mathbb{F})$ denote the space of $n \times n$ skew-symmetric (alternating) matrices over $\mathbb{F}$. Then the Baer Correspondence [7] gives an equivalence between

$$\left\{\begin{matrix}p\text{-groups of class 2, ex-}\\ \text{ponent } p,\ G/Z(G) \cong\\ \mathbb{Z}_p^n,\ Z(G) \cong \mathbb{Z}_p^m\end{matrix}\right\} \longleftrightarrow \left\{\begin{matrix}\mathcal{A} \leq \Lambda(n, \mathbb{F}_p)\\ \dim \mathcal{A} = m\end{matrix}\right\} \longleftrightarrow \left\{\begin{matrix}\text{Nilpotent } \mathbb{F}_p\text{-Lie algebras}\\ \text{of class 2, } L/Z(L) \cong \mathbb{F}_p^n,\\ Z(L) \cong \mathbb{F}_p^m\end{matrix}\right\}$$

in such a way that two such groups are isomorphic iff the corresponding Lie algebras are isomorphic iff the corresponding matrix spaces $\mathcal{A}, \mathcal{B} \leq \Lambda(n, \mathbb{F}_p)$ are isometric. Here, we say that two such linear subspaces are *isometric* if there is an invertible matrix $L \in \mathrm{GL}(n, \mathbb{F}_p)$ such that $\mathcal{B} = L^t \mathcal{A} L := \{L^t A L : A \in \mathcal{A}\}$.[3] The corresponding computational problem is:

▶ **Definition 4** (The Alternating Matrix Space Isometry problem).
Input: $A_1, \dots, A_m$ *and* $B_1, \dots, B_m$*,* $n \times n$ *alternating*[4] *matrices over a field* $\mathbb{F}$*,*
Decide: Is there a $L \in \mathrm{GL}(n, \mathbb{F})$*, such that the linear span of* $\{A_i : i \in [m]\}$ *is equal to the linear span of* $\{L^t B_i L : i \in [m]\}$*?*

Our search- and counting-to-decision reductions (Theorems A and B) actually follow from analogous results on Alternating Matrix Space Isometry (Theorems A′ and B′), using a constructive version of the Baer Correspondence communicated to us by James B. Wilson (Lemma 24). The viewpoint of alternating matrix spaces made the constructions much easier to find and reason about.

Our nilpotency class reduction uses a constructive version of the Lazard Correspondence (Proposition 26), which generalizes the Baer Correpsondence to nilpotency class $c < p$; the TI-completeness of Lie Algebra Isomorphism for nilpotent Lie algebras of class 2 (a combination of reductions from [28] and [32]); and finally the aforementioned constructive Baer Correspondence to go back to p-groups of class 2.

In the remainder of this section we give more details of the techniques involved.

1.2.1 Linear algebraic coloring gadgets

Our most novel technique is to devise linear algebraic analogues for Alternating Matrix Space Isometry of the graph coloring gadget, a key technique in the structural complexity study of Graph Isomorphism (see, e. g., [44]). This technique is crucial in the following theorems, used to prove Theorems A and B, respectively.

▶ **Theorem A′.** *Let* q *be a prime power. There is a search-to-decision reduction for* Alternating Matrix Space Isometry *which, given* $n \times n$ *alternating matrix spaces* $\mathcal{A}, \mathcal{B}$ *over* $\mathbb{F}_q$ *of dimension* m*, computes an isometry between them if they are isometric, in time* $q^{\tilde{O}(n)}$ *or in time* $q^{O(n+m)}$*. The reduction queries the decision oracle with inputs of dimension at most* $O(n^2)$*.*

[3] For bilinear maps – which are another way of viewing matrix spaces – the corresponding notion is often called "pseudo-isometry", with "isometry" of bilinear maps being a more restrictive notion. We chose our nomenclature by analogy with individual matrices: just as we call two matrix spaces $\mathcal{A}, \mathcal{B}$ "conjugate" when $L\mathcal{A}L^{-1} = \mathcal{B}$, or "equivalent" when $L\mathcal{A}M = \mathcal{B}$, we call two matrix spaces "isometric" when there is an isometry-transformation that sends one such space to another. We are careful to use "pseudo-isometry" when we refer to the corresponding notions for matrix *tuples* or for bilinear maps.

[4] An $n \times n$ matrix A over $\mathbb{F}$ is alternating if for every $v \in \mathbb{F}^n$, $v^t A v = 0$. When $\mathbb{F}$ is not of characteristic 2, this is equivalent to being skew-symmetric $A^t = -A$.

▶ **Theorem B′.** *For q a prime power with $q = n^{\Omega(1)}$, there is a randomized counting-to-decision reduction for* Alternating Matrix Space Isometry *which, given $n \times n$ alternating matrix spaces $\mathcal{A}, \mathcal{B}$ over $\mathbb{F}_q$ of dimension m, computes the number of isometries from $\mathcal{A}$ to $\mathcal{B}$ in time $q^{O(n)}$. The reduction queries the decision oracle with inputs of dimension at most $O(n^2)$.*

Let us first briefly review the graph coloring gadgets. Suppose we have a graph $G = (V, E)$ with the vertices colored, i. e., there is a map $f : V \to \{1, \ldots, c\} =: [c]$, where we view $[c]$ as the set of colors. Let $n = |V|$. Suppose we want to construct an uncolored graph $\tilde{G}$, in which the color information carried by f is encoded. One way to achieve this is the following. (See [44] for other more efficient constructions.) For every $v \in V$, if $v \in V$ is assigned color $k \in [c]$, then attach a "star" of size kn to v, that is add kn new vertices to G and attach them all to v. We then get a graph $\tilde{G}$ with $O(cn^2)$ vertices, and we see that an automorphism of $\tilde{G}$, when restricting to V, has to map $v \in V$ to another $v' \in V$ of the same color, as degrees need to be preserved under automorphisms.

Such an idea can be carried out in the 3-tensor context as in [28], but with a significant loss of efficiency which prevents its use for search- and counting-to-decision reductions and indicates the needs for new techniques. To illustrate the situation, we consider a toy problem. To ease the presentation, we adopt a perspective on 3-tensors that we hope is clear on its own; the analogy with the graph case is fairly close, but not immediately obvious, and we present it in full detail in Section 3. Note that by slicing a 3-tensor along one direction, we get a tuple of matrices (see also Section 2); in the following of this subsection we shall mostly work with matrix tuples.

Let $\mathbf{A} = (A_1, \ldots, A_m) \in \mathrm{M}(n, \mathbb{F})^m$ be a tuple of matrices, where A_i's are linearly independent. There are two natural actions on $\mathbf{A}$. The first action is $S = (s_{i,j}) \in \mathrm{GL}(m, \mathbb{F})$ on $\mathbf{A}$ by sending A_j to $\sum_{i \in [m]} s_{i,j} A_i$. Denote the resulting matrix tuple by $\mathbf{A}^S$. The second action is $(L, R) \in \mathrm{GL}(n, \mathbb{F}) \times \mathrm{GL}(n, \mathbb{F})$ on $\mathbf{A}$ by sending A_j to LA_jR^t for $j = 1, \ldots, m$. Denote the resulting matrix tuple by $L\mathbf{A}R^t$. For two tuples $\mathbf{A}, \mathbf{B}$, and for the purposes of this illustration, let us define the set of isomorphisms as $\mathrm{Iso}(\mathbf{A}, \mathbf{B}) = \{S \in \mathrm{GL}(m, \mathbb{F}) : \exists L, R \in \mathrm{GL}(n, \mathbb{F}), L\mathbf{A}R^t = \mathbf{B}^S\}$.

In the counting-to-decision reduction we will need to test isomorphism of such tuples under the action by *diagonal* matrices. Let $\mathrm{diag}(m, \mathbb{F})$ denote the subgroup of $\mathrm{GL}(m, \mathbb{F})$ consisting of diagonal matrices. Our goal then is to construct $\tilde{\mathbf{A}} = (\tilde{A}_1, \tilde{A}_2, \tilde{A}_3) \in \mathrm{M}(N, \mathbb{F})^3$ and $\tilde{\mathbf{B}}$, such that $\mathrm{Iso}(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}) = \mathrm{Iso}(\mathbf{A}, \mathbf{B}) \cap \mathrm{diag}(3, \mathbb{F})$. The construction we use, from [28], is as follows. Let $N = 2^3 \cdot n = 8n$, and let

$$\tilde{A}_1 = \begin{bmatrix} A_1 & 0 & 0 & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tilde{A}_2 = \begin{bmatrix} A_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I_{2n} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tilde{A}_3 = \begin{bmatrix} A_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{4n} \end{bmatrix}, \tag{1}$$

where I_s denotes the identity matrix of size s, and 0's denote all-zero matrices of appropriate sizes, and define $\tilde{\mathbf{B}}$ similarly. By [28, Lemma 2.2], we have $\mathrm{Iso}(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}) = \mathrm{Iso}(\mathbf{A}, \mathbf{B}) \cap \mathrm{diag}(3, \mathbb{F})$. The proof, while not difficult, relies on certain algebraic machineries like the Krull–Schmidt Theorem for quiver representations. For our purpose, we only point out that a key in the proof is that $\mathrm{Iso}(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}) \subseteq \mathrm{diag}(3, \mathbb{F})$, which can be easily checked by comparing the ranks of the $\tilde{A}_i, \tilde{B}_i$. (We note that, because L and R act independently on the rows and columns of the $\tilde{A}_i$, for individual slices rank is essentially the only invariant we have.)

The preceding gadget construction can be generalized to handle subgroups of $\mathrm{GL}(n,\mathbb{F})$ of the form

$$\left\{ \begin{bmatrix} S_1 & 0 & \dots & 0 \\ 0 & S_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S_c \end{bmatrix} : S_i \in \mathrm{GL}(n_i,\mathbb{F}) \right\},$$

where $c = O(\log n)$. We shall refer to this gadget as the Futorny–Grochow–Sergeichuk gadget, or FGS gadget for short.

However, the FGS gadget cannot be used for search- and counting-to-decision reductions in Theorems A and B. The key bottleneck is the restriction that $c = O(\log n)$. To check why this is so reveals an interesting distinction between the combinatorial and the linear algebraic worlds. Recall that in the graph setting, if there are c colors, we need stars of size at most cn. While in the linear algebraic setting, if there are c components, the biggest identity matrix needs to be of size $2^c \cdot n \times 2^c \cdot n$. The reason is that we can do non-trivial linear combinations of the matrices $\tilde{A}_i$, so several matrices of small ranks might be combined to get a matrix of large rank. Indeed, in Eq. 1, if $\tilde{A}_3$ was accompanied with I_{3n} instead of I_{4n}, then a non-trivial linear combination of $\tilde{A}_1$ and $\tilde{A}_2$ could be of rank the same as $\tilde{A}_3$, and the argument that $\mathrm{Iso}(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}) \subseteq \mathrm{diag}(m,\mathbb{F})$ would not go through. That's why we need such exponential growth as the number of components grow.

To address this challenge, we devise two new gadgets, which restrict to the monomial group and the diagonal group, respectively.

The monomial group of $\mathrm{GL}(n,\mathbb{F})$, denoted as $\mathrm{Mon}(n,\mathbb{F})$, consists of monomial matrices, i.e. a matrix with exactly one non-zero entry in each row and each column. We design a gadget that restricts to $\mathrm{Mon}(n,\mathbb{F})$, which is the key in the search-to-decision reduction (Theorem A′).

In the case of $\mathbb{F} = \mathbb{F}_q$ and $q = n^{\Omega(1)}$, we design a gadget that restricts to $\mathrm{diag}(n,q)$, which is the key in the counting-to-decision reduction (Theorem B′). The gadget for restricting to monomial groups cannot be used in the counting-to-decision reduction. Its construction is already delicate, and the analysis is involved, relying on a celebrated result of Kantor and Lubotzky regarding random generation of classical groups [41].

1.2.2 Constructive Lazard Correspondence

In light of the TI-completeness of isomorphism of class 2 p-groups given by matrices over finite fields of characteristic p [32], the key idea here is how to reduce isomorphism for other classes of groups to some tensor problem. For groups in general it is unclear how to do this, as tensors are multilinear and groups are not. But for p-groups of nilpotency class $< p$, the Lazard Correspondence gives an equivalence between the category of such groups and a corresponding category of Lie algebras (over the same field, nilpotent of the same class). If this correspondence were computationally efficient, we would then be in the fortunate setting in which Lie Algebra Isomorphism is multilinear, and is in TI [28], so we can then reduce back to isomorphism of class 2 p-groups. We observe (Proposition 26) that when the groups are given by matrices in characteristic p, the Lazard Correspondence can be efficiently computed using the usual matrix logarithm and exponential.

The restriction to groups of nilpotency class $c < p$ comes entirely from the Lazard Correspondence, which is also known only to work under this same assumption (see [60] for details, and what can be said when $c = p$, but unfortunately already when $c = p$ one no longer gets an equivalence up to isomorphism). Despite this restriction, we note that we know of no prior reductions from *any* class of groups to p-groups of class 2.

In Remark 2 we discuss the ingredients necessary to get the same result for GpI in the Cayley table model, which seems approachable.

1.3 Organization of the paper

In Section 2 we present preliminaries and notation. In Section 3 we present more details of the analogy with individualizing vertices in graphs by attaching stars, using the example of reducing Monomial Code Equivalence to Tensor Isomorphism. In Section 4 we present our gadget to restrict to the monomial subgroup, an example use of this to reduce GI to Alternating Matrix Space Isometry, and Theorem A′. In Section 5 we prove Theorem B′. In Section 6 we present the constructive Baer and Lazard Correspondences, and use them to derive Theorems A and B from Theorems A′ and B′, respectively, as well as proving Theorem P. Finally, in Section 7 we conclude with open questions and discuss the relationship between this work and the authors' line of work on Tensor Isomorphism.

2 Preliminaries

Table 1 Summary of notation related to 3-way arrays and tensors.

Font	Object	Space of objects
$A, B, \ldots$	matrix	$\mathrm{M}(n, \mathbb{F})$ or $\mathrm{M}(\ell \times n, \mathbb{F})$
$\mathbf{A}, \mathbf{B}, \ldots$	matrix tuple	$\mathrm{M}(n, \mathbb{F})^m$ or $\mathrm{M}(\ell \times n, \mathbb{F})^m$
$\mathcal{A}, \mathcal{B}, \ldots$	matrix space	[Subspaces of $\mathrm{M}(n, \mathbb{F})$ or $\Lambda(n, \mathbb{F})$]
$\mathtt{A}, \mathtt{B}, \ldots$	3-way array	$\mathrm{T}(\ell \times n \times m, \mathbb{F})$

Vector spaces. Let $\mathbb{F}$ be a field. In this paper we only consider finite-dimensional vector spaces over $\mathbb{F}$. We use $\mathbb{F}^n$ to denote the vector space of length-n *column* vectors. The ith standard basis vector of $\mathbb{F}^n$ is denoted $\vec{e_i}$. Depending on the context, $\mathbf{0}$ may denote the zero vector space, a zero vector, or an all-zero matrix. For S a set of vectors, we use $\langle S \rangle$ to denote the subspace spanned by elements in S.

Some groups. The general linear group of degree n over a field $\mathbb{F}$ is denoted by $\mathrm{GL}(n, \mathbb{F})$. The symmetric group of degree n is denoted by S_n. The natural embedding of S_n into $\mathrm{GL}(n, \mathbb{F})$ is to represent permutations by permutation matrices. The subgroup of $\mathrm{GL}(n, \mathbb{F})$ consisting of diagonal matrices is called the *diagonal subgroup*, denoted by $\mathrm{diag}(n, \mathbb{F})$. A *monomial matrix* is a product of a diagonal and a permutation matrix; equivalently, each row and each column has exactly one non-zero entry. The collection of monomial matrices forms a subgroup of $\mathrm{GL}(n, \mathbb{F})$, which we call the *monomial subgroup* and denote by $\mathrm{Mon}(n, \mathbb{F})$. It is the semi-direct product $\mathrm{diag}(n, \mathbb{F}) \rtimes S_n \cong (\mathbb{F}^*)^n \rtimes S_n$.

Nilpotent groups. If A, B are two subsets of a group G, then $[A, B]$ denotes the sub*group* generated by all elements of the form $[a, b] = aba^{-1}b^{-1}$, for $a \in A, b \in B$. The *lower central series* of a group G is defined as follows: $\gamma_1(G) = G$, $\gamma_{k+1}(G) = [\gamma_k(G), G]$. A group is *nilpotent* if there is some c such that $\gamma_{c+1}(G) = 1$; the smallest such c is called the *nilpotency class* of G, or sometimes just "class" when it is understood from context. A finite group is nilpotent if and only if it is the product of its Sylow subgroups; in particular, all groups of prime power order are nilpotent.

Matrices. Let $\mathrm{M}(\ell \times n, \mathbb{F})$ be the linear space of $\ell \times n$ matrices over $\mathbb{F}$, and $\mathrm{M}(n, \mathbb{F}) := \mathrm{M}(n \times n, \mathbb{F})$. Given $A \in \mathrm{M}(\ell \times n, \mathbb{F})$, A^t denotes the transpose of A.

A matrix $A \in \mathrm{M}(n, \mathbb{F})$ is *alternating*, if for any $u \in \mathbb{F}^n$, $u^t A u = 0$. That is, A represents an alternating bilinear form. Note that in characteristic $\neq 2$, alternating is the same as skew-symmetric, but in characteristic 2 they differ (in characteristic 2, skew-symmetric=symmetric). The linear space of $n \times n$ alternating matrices over $\mathbb{F}$ is denoted by $\Lambda(n, \mathbb{F})$.

The $n \times n$ *identity matrix* is denoted by I_n, and when n is clear from the context, we may just write I. The *elementary matrix* $E_{i,j}$ is the matrix with the (i,j)th entry being 1, and other entries being 0. The (i,j)*-th elementary alternating matrix* is the matrix $E_{i,j} - E_{j,i}$.

Matrix tuples. We use $\mathrm{M}(\ell \times n, \mathbb{F})^m$ to denote the linear space of m-tuples of $\ell \times n$ matrices. Boldface letters like $\mathbf{A}$ and $\mathbf{B}$ denote matrix tuples. Let $\mathbf{A} = (A_1, \dots, A_m), \mathbf{B} = (B_1, \dots, B_m) \in \mathrm{M}(\ell \times n, \mathbb{F})^m$. Given $P \in \mathrm{M}(\ell, \mathbb{F})$ and $Q \in \mathrm{M}(n, \mathbb{F})$, $P\mathbf{A}Q := (PA_1Q, \dots, PA_mQ) \in \mathrm{M}(\ell, \mathbb{F})$. Given $R = (r_{i,j})_{i,j \in [m]} \in \mathrm{M}(m, \mathbb{F})$, $\mathbf{A}^R := (A'_1, \dots, A'_m) \in \mathrm{M}(m, \mathbb{F})$ where $A'_i = \sum_{j \in [m]} r_{j,i} A_j$.

▶ Remark 5. In particular, note that the coefficients in the formula of defining A'_i correspond to the entries in the ith *column* of R. While this choice is immaterial (we could have chosen the opposite convention), all of our later calculations are consistent with this convention.

Given $\mathbf{A}, \mathbf{B} \in \mathrm{M}(\ell \times n, \mathbb{F})^m$, we say that $\mathbf{A}$ and $\mathbf{B}$ are *isometric*, if there exists $P \in \mathrm{GL}(n, \mathbb{F})$, such that $P^t \mathbf{A} P = \mathbf{B}$. Finally, $\mathbf{A}$ and $\mathbf{B}$ are *pseudo-isometric* if there exist $P \in \mathrm{GL}(n, \mathbb{F})$ and $R \in \mathrm{GL}(m, \mathbb{F})$, such that $P^t \mathbf{A} P = \mathbf{B}^R$.

Matrix spaces. Linear subspaces of $\mathrm{M}(\ell \times n, \mathbb{F})$ are called matrix spaces. Calligraphic letters like $\mathcal{A}$ and $\mathcal{B}$ denote matrix spaces. By a slight abuse of notation, for $\mathbf{A} \in \mathrm{M}(\ell \times n, \mathbb{F})^m$, we use $\langle \mathbf{A} \rangle$ to denote the subspace spanned by those matrices in $\mathbf{A}$. For $\mathbf{A}, \mathbf{B} \in \mathrm{M}(n, \mathbb{F})^m$, we say that the spaces $\langle \mathbf{A} \rangle, \langle \mathbf{B} \rangle$ are isometric iff the tuples $\mathbf{A}, \mathbf{B}$ are pseudo-isometric.

3-way arrays. Let $\mathrm{T}(\ell \times n \times m, \mathbb{F})$ be the linear space of $\ell \times n \times m$ 3-way arrays over $\mathbb{F}$. We use the fixed-width teletypefont for 3-way arrays, like $\mathtt{A}$, $\mathtt{B}$, etc..

Given $\mathtt{A} \in \mathrm{T}(\ell \times n \times m, \mathbb{F})$, we can think of $\mathtt{A}$ as a 3-dimensional table, where the (i,j,k)th entry is denoted as $\mathtt{A}(i,j,k) \in \mathbb{F}$. We can slice $\mathtt{A}$ along one direction and obtain several matrices, which are then called slices. For example, slicing along the first coordinate, we obtain the *horizontal* slices, namely ℓ matrices $A_1, \dots, A_\ell \in \mathrm{M}(n \times m, \mathbb{F})$, where $A_i(j,k) = \mathtt{A}(i,j,k)$. Similarly, we also obtain the *lateral* slices by slicing along the second coordinate, and the *frontal* slices by slicing along the third coordinate.

We will often represent a 3-way array as a matrix whose entries are vectors. That is, given $\mathtt{A} \in \mathrm{T}(\ell \times n \times m, \mathbb{F})$, we can write

$$\mathtt{A} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ w_{\ell,1} & w_{\ell,2} & \dots & w_{\ell,n} \end{bmatrix},$$

where $w_{i,j} \in \mathbb{F}^m$, so that $w_{i,j}(k) = \mathtt{A}(i,j,k)$. Note that, while $w_{i,j} \in \mathbb{F}^m$ are column vectors, in the above representation of $\mathtt{A}$, we should think of them as along the direction "orthogonal to the paper." Following [45], we call $w_{i,j}$ the *tube fibers* of $\mathtt{A}$. Similarly, we can have the *row fibers* $v_{i,k} \in \mathbb{F}^n$ such that $v_{i,k}(j) = \mathtt{A}(i,j,k)$, and the *column fibers* $u_{j,k} \in \mathbb{F}^\ell$ such that $u_{j,k}(i) = \mathtt{A}(i,j,k)$.

Given $P \in \mathrm{M}(\ell, \mathbb{F})$ and $Q \in \mathrm{M}(n, \mathbb{F})$, let $P\mathtt{A}Q$ be the $\ell \times n \times m$ 3-way array whose kth frontal slice is PA_kQ. For $R = (r_{i,j}) \in \mathrm{GL}(m, \mathbb{F})$, let $\mathtt{A}^R$ be the $\ell \times n \times m$ 3-way array whose kth frontal slice is $\sum_{k' \in [m]} r_{k',k} A_{k'}$. Note that these notations are consistent with the notations for matrix tuples above, when we consider the matrix tuple $\mathbf{A} = (A_1, \ldots, A_k)$ of frontal slices of $\mathtt{A}$.

3 Warm up: reducing Monomial Code Equivalence to Tensor Isomorphism

The purpose of this section is to present a concrete example that illustrates what we mean by a gadget restricting to monomial subgroups. We also explain why the gadget would be viewed as a linear algebraic analogue of attaching stars in the graph setting as mentioned in Section 1.2.1.

We will give a reduction here to the Tensor Isomorphism (TI) problem, so we begin by recalling its definition:

▶ **Definition 6** (The d-Tensor Isomorphism problem). d-Tensor Isomorphism *over a field $\mathbb{F}$ is the problem: given two d-way arrays $\mathtt{A} = (a_{i_1,\ldots,i_d})$ and $\mathtt{B} = (b_{i_1,\ldots,i_d})$, where $i_k \in [n_k]$ for $k \in [d]$, and $a_{i_1,\ldots,i_d}, b_{i_1,\ldots,i_d} \in \mathbb{F}$, decide whether there are $P_k \in \mathrm{GL}(n_k, \mathbb{F})$ for $k \in [d]$, such that for all $i_1, \ldots, i_d$,*

$$a_{i_1,\ldots,i_d} = \sum_{j_1,\ldots,j_d} b_{j_1,\ldots,j_d} (P_1)_{i_1,j_1} (P_2)_{i_2,j_2} \cdots (P_d)_{i_d,j_d}.$$

Let $\mathtt{A}$ be an $\ell \times n \times m$ 3-way array, with lateral slices $L_1, L_2, \ldots, L_n$ (each an $\ell \times m$ matrix). For any vector $v \in \mathbb{F}^n$, we get an associated lateral matrix L_v, which is a linear combination of the lateral slices as given, namely $L_v := \sum_{j=1}^n v_j L_j$ (note that when $v = \vec{e_j}$ is the j-th standard basis vector, the associated lateral matrix is indeed L_j). By analogy with adjacency matrices of graphs, L_v is a natural analogue of the neighborhood of a vertex in a graph. Correspondingly, we get a notion of "degree," which we may define as

$$\deg_{\mathtt{A}}(v) \quad := \quad \mathrm{rk} L_v = \mathrm{rk}(\sum_{j=1}^{n} v_j L_j) = \dim \mathrm{span}\{L_v w : w \in \mathbb{F}^m\} = \dim \mathrm{span}\{u^t L_v : u \in \mathbb{F}^\ell\}.$$

The last two characterizations are analogous to the fact that the degree of a vertex v in a graph G may be defined as the number of "in-neighbors" (nonzero entries the corresponding row of the adjacency matrix) or the number of "out-neighbors" (nonzero entries in the corresponding column).

To "individualize" v, we can enlarge $\mathtt{A}$ with a gadget to increase $\deg_{\mathtt{A}}(v)$, as in the graph case. Note that $\deg_{\mathtt{A}}(v) \leq \min\{\ell, m\}$ because the lateral matrices are all of size $\ell \times m$. For notational simplicity, let us individualize $v = \vec{e_1} = (1, 0, \ldots, 0)^t$. To individualize v, we will increase its degree by $d = \min\{\ell, m\} + 1 > \max_{v \in \mathbb{F}^n} \deg_{\mathtt{A}}(v)$. Extend $\mathtt{A}$ to a new 3-way array $\mathtt{A}_v$ of size $(\ell + d) \times n \times (m + d)$; in the "first" $\ell \times n \times m$ "corner", we will have the original array $\mathtt{A}$, and then we will append to it an identity matrix in one slice to increase $\deg(v)$. More specifically, the lateral slices of $\mathtt{A}_v$ will be

$$L'_1 = \begin{bmatrix} L_1 & 0 \\ 0 & I_d \end{bmatrix} \qquad \text{and} \qquad L'_j = \begin{bmatrix} L_j & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{for } j > 1).$$

Now we have that $\deg_{\mathtt{A}_v}(v) \geq d$. This almost does what we want, but now note that any vector $w = (w_1, \ldots, w_n)$ with $w_1 \neq 0$ has $\deg_{\mathtt{A}_v}(w) = \mathrm{rk}(w_1 L'_1 + \sum_{j \geq 2} w_j L_j) \geq d$. We can nonetheless consider this a sort of linear-algebraic individualization.

Leveraging this trick, we can then individualize an entire basis of $\mathbb{F}^n$ simultaneously, so that $d \leq \deg(v) < 2d$ for any vector v in our basis, and $\deg(v') \geq 2d$ for any nonzero v' outside the basis (not a scalar multiple of one of the basis vectors), as we do in the following result. This is also a 3-dimensional analogue of the reduction from GI to CodeEq [52,59,62] (where they use Hamming weight instead of rank).

We now come to the concrete result. Given two $d \times n$ matrices A, B over $\mathbb{F}$ of rank d, the Monomial Code Equivalence problem is to decide whether there exist $Q \in \mathrm{GL}(d, \mathbb{F})$ and a monomial matrix $P \in \mathrm{Mon}(n, \mathbb{F}) \leq \mathrm{GL}(n, \mathbb{F})$ (product of a diagonal matrix and a permutation matrix) such that $QAP = B$. Monomial equivalence of linear codes is a basic notion in coding theory [11], and Monomial Code Equivalence was recently studied in the context of post-quantum cryptography [67].

▶ **Proposition 7.** Monomial Code Equivalence *reduces to* 3-Tensor Isomorphism.

Proof. Without loss of generality we assume $d > 1$, as the problem is easily solvable when $d = 1$. We treat a $d \times n$ matrix A as a 3-way array of size $d \times n \times 1$, and then follow the outline proposed above, of individualizing the entire standard basis $\vec{e_1}, \ldots, \vec{e_n}$. Since the third direction only has length 1, the maximum degree of any column is 1, so it suffices to use gadgets of rank 2. More specifically, (see Figure 1) we build a $(d + 2n) \times n \times (1 + 2n)$ 3-way array $\mathtt{A}$ whose lateral slices are

$$L_j = \begin{bmatrix} a_{1,j} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{d,j} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} & \cdots & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & I_2 & \cdots & \mathbf{0}_{2\times 2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_{2\times 1} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} & \cdots & \mathbf{0}_{2\times 2} \end{bmatrix}$$

where the I_2 block is in the j-th block of size 2 (that is, rows $d+2(j-1)+\{1,2\}$ and columns $2(j-1)+\{1,2\}$).

It will also be useful to visualize the frontal slices of $\mathtt{A}$, as follows. Here each entry of the "matrix" below is actually a $(1 + 2n)$-dimensional vector, "coming out of the page":

$$\mathtt{A} = \begin{bmatrix} \tilde{a}_{1,1} & \tilde{a}_{1,2} & \ldots & \tilde{a}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{d,1} & \tilde{a}_{d,2} & \ldots & \tilde{a}_{d,n} \\ e_{1,1} & \mathbf{0} & \ldots & \mathbf{0} \\ e_{1,2} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & e_{2,1} & \ldots & \mathbf{0} \\ \mathbf{0} & e_{2,2} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & e_{n,1} \\ \mathbf{0} & \mathbf{0} & \ldots & e_{n,2} \end{bmatrix},$$

where

$$\begin{aligned} \tilde{a}_{i,j} &= \begin{bmatrix} a_{i,j} \\ \mathbf{0}_{2n\times 1} \end{bmatrix} \in \mathbb{F}^{1+2n} \\ e_{i,j} &= \vec{e}_{1+2(i-1)+j} \in \mathbb{F}^{1+2n} \text{ for } i \in [n], j \in [2] \end{aligned}$$

and the frontal slices are

$$\begin{aligned} A_1 &= \begin{bmatrix} A \\ \mathbf{0}_{2n\times n} \end{bmatrix} \\ A_{1+2(i-1)+j} &= E_{d+2(i-1)+j,i} \qquad \text{for } i \in [n], j \in [2] \end{aligned}$$

(In $\mathtt{A}$ we turn the vectors $\tilde{a}_{i,j}$ and $e_{i,j}$ "on their side" so they become perpendicular to the page.)

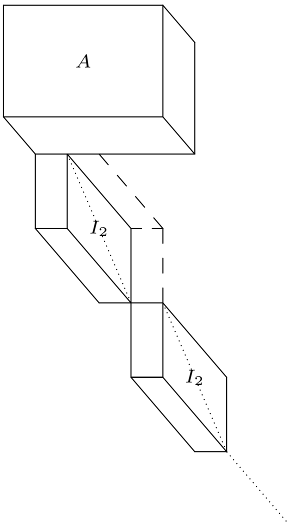


Figure 1 Pictorial representation of the reduction for Proposition 7.

We claim that A and B are monomially equivalent as codes if and only if $\mathtt{A}$ and $\mathtt{B}$ are isomorphic as 3-tensors.

($\Rightarrow$) Suppose $QADP = B$ where $Q \in \mathrm{GL}(d, \mathbb{F})$, $D \in \mathrm{diag}(n, \mathbb{F})$ and $P \in S_n \leq \mathrm{GL}(n, \mathbb{F})$. Then by examining the frontal slices it is not hard to see that for $Q' = \begin{bmatrix} Q & 0 \\ 0 & (DP)^{-1} \otimes I_2 \end{bmatrix}$ (where $(DP)^{-1} \otimes I_2$ denotes a $2n \times 2n$ block matrix, where the pattern of the nonzero blocks and the scalars are governed by $(DP)^{-1}$, and each 2×2 block is either zero or a scalar multiple of I_2) we have $Q'A_1DP = B_1$ and $Q'A_{1+2(i-1)+j}DP = B_{1+2(\pi(i)-1)+j}$, where π is the permutation corresponding to P. Thus $\mathtt{A}$ and $\mathtt{B}$ are isomorphic tensors, via the isomorphism $(Q', DP, \mathrm{diag}(I_1, P))$.

($\Leftarrow$) Suppose there exist $Q \in \mathrm{GL}(d + 2n, \mathbb{F})$, $P \in \mathrm{GL}(n, \mathbb{F})$, and $R \in \mathrm{GL}(1 + 2n, \mathbb{F})$, such that $Q\mathtt{A}P = \mathtt{B}^R$. First, note that every lateral slice of $\mathtt{A}$ is of rank either 2 or 3, and the actions of Q and R do not change the ranks of the lateral slices. Furthermore, any non-trivial linear combination of more than 1 lateral slice results in a lateral matrix of rank ≥ 4. It follows that P cannot take nontrivial linear combinations of the lateral slices, hence it must be monomial.

Now consider the frontal slices. Note that, as we assume $d > 1$, every frontal slice of $Q\mathtt{A}P$, except the first one, is of rank 1. Therefore, R must be of the form $\begin{bmatrix} r_{1,1} & \mathbf{0}_{1\times(n-1)} \\ \vec{r'} & R' \end{bmatrix}$ where R' is $(n-1) \times (n-1)$. Since R is invertible, we must have $r_{1,1} \neq 0$, and the first frontal slice of $\mathtt{B}^R$ contains all the rows of B scaled by $r_{1,1}$ in its first d rows. The first frontal slice of $Q\mathtt{A}P$ is a matrix that generates, by definition (and since we've shown P is monomial), a code monomially equivalent to A. Since the first frontal slices of $Q\mathtt{A}P$ and $\mathtt{B}^R$ are equal, and the latter is just a scalar multiple of B_1, we have that A and B are monomially equivalent as codes as well. ◀

4 Search-to-decision reduction by restricting to monomial groups

4.1 The gadget restricting to the monomial group

In this section, we present the gadget that restricts to the monomial group in the setting of Alternating Matrix Space Isometry. To show this, we will need the concept of monomial isometry; see Some Groups above. Recall that a matrix is monomial if, equivalently, it can be written as DP where D is a nonsingular diagonal matrix and P is a permutation matrix. We say two matrix spaces $\mathcal{A}, \mathcal{B}$ are *monomially isometric* if there is some $M \in \mathrm{Mon}(n, \mathbb{F})$ such that $M^t \mathcal{A} M = \mathcal{B}$.

▶ **Lemma 8.** Alternating Matrix Space Monomial Isometry *reduces to* Alternating Matrix Space Isometry.

More specifically, there is a $\mathrm{poly}(n, m)$*-time algorithm* r *taking alternating matrix tuples to alternating matrix tuples, such that for* $\mathbf{A}, \mathbf{B} \in \Lambda(n, \mathbb{F})^m$*, the matrix spaces* $\mathcal{A} = \langle \mathbf{A} \rangle$ *and* $\mathcal{B} = \langle \mathbf{B} \rangle$ *are monomially isometric if and only if the matrix spaces* $\langle r(\mathbf{A}) \rangle$ *and* $\langle r(\mathbf{B}) \rangle$ *are isometric.*

The gadget used in Lemma 8 is essentially to apply the gadget in Proposition 7 "in two directions." Still, to prove the correctness requires some work.

Proof. For $\mathbf{A} = (A_1, \dots, A_m) \in \Lambda(n, \mathbb{F})^m$, define $r(\mathbf{A})$ to be the alternating matrix tuple $\tilde{\mathbf{A}} = (\tilde{A}_1, \dots, \tilde{A}_{m+n^2}) \in \Lambda(n + n^2, \mathbb{F})^{m+n^2}$, where

1. For $k = 1, \dots, m$, $\tilde{A}_k = \begin{bmatrix} A_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$.
2. For $k = m + (i-1)n + j$, $i \in [n]$, $j \in [n]$, $\tilde{A}_k$ is the elementary alternating matrix $E_{i,in+j} - E_{in+j,i}$.

At this point, some readers may wish to look at the large matrix in Equation 2 and/or at Figure 2.

It is clear that r can be computed in time $\tilde{O}((m + n^2)(n^2 + n)) = \mathrm{poly}(n, m)$. Given alternating matrix tuples $\mathbf{A}, \mathbf{B}$, let $\mathcal{A}, \mathcal{B}$ be the corresponding matrix spaces they span, and let $\tilde{\mathcal{A}} = \langle r(\mathbf{A}) \rangle$ and $\tilde{\mathcal{B}} = \langle r(\mathbf{B}) \rangle$. We claim that $\mathcal{A}$ and $\mathcal{B}$ are monomially isometric if and only if $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are isometric.

To prove this, it will help to think of our matrix tuples $\mathbf{A}, \tilde{\mathbf{A}}$, etc. as (corresponding to) 3-way arrays, and to view these 3-way arrays from two different directions. Towards this end, write the 3-way array corresponding to $\mathbf{A}$ as

$$\mathtt{A} = \begin{bmatrix} \mathbf{0} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ -a_{1,2} & \mathbf{0} & a_{2,3} & \dots & a_{2,n} \\ -a_{1,3} & -a_{2,3} & \mathbf{0} & \dots & a_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -a_{1,n} & -a_{2,n} & -a_{3,n} & \dots & \mathbf{0} \end{bmatrix},$$

where $a_{i,j}$ are vectors in $\mathbb{F}^m$ ("coming out of the page"), namely $a_{i,j}(k) = A_k(i,j)$. The frontal slices of this array are precisely the matrices $A_1, \dots, A_m$.

The 3-way array corresponding to $\tilde{\mathbf{A}} = r(\mathbf{A})$ is then the $(n+1)n \times (n+1)n \times (m + n^2)$ array:

$$\tilde{\mathtt{A}} = \left[\begin{array}{ccccc:ccc:ccc:c:ccc}
\mathbf{0} & \tilde{a}_{1,2} & \tilde{a}_{1,3} & \cdots & \tilde{a}_{1,n} & e_{1,1} & \cdots & e_{1,n} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\
-\tilde{a}_{1,2} & \mathbf{0} & \tilde{a}_{2,3} & \cdots & \tilde{a}_{2,n} & \mathbf{0} & \cdots & \mathbf{0} & e_{2,1} & \cdots & e_{2,n} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \ddots & \ddots & \ddots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \ddots & \ddots & \vdots \\
-\tilde{a}_{1,n} & -\tilde{a}_{2,n} & -\tilde{a}_{3,n} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & e_{n,1} & \cdots & e_{n,n} \\
\hdashline
-e_{1,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
-e_{1,n} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\
\hdashline
\mathbf{0} & -e_{2,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
\mathbf{0} & -e_{2,n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\
\hdashline
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
\hdashline
\mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -e_{n,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -e_{n,n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0}
\end{array}\right], \tag{2}$$

where $\tilde{a}_{i,j} = \begin{bmatrix} a_{i,j} \\ \mathbf{0} \end{bmatrix} \in \mathbb{F}^{m+n^2}$ (here think of the vector $a_{i,j}$ as a column vector, *not* coming out of the page; in the above array we then lay the column vector $\tilde{a}_{i,j}$ "on its side" so that it is coming out of the page), and $e_{i,j} := e_{m+(i-1)n+j} \in \mathbb{F}^{m+n^2}$, which we can equivalently write as $\begin{bmatrix} \mathbf{0}_m \\ e_i \otimes e_j \end{bmatrix}$, where we think of $e_i \otimes e_j$ here as a vector of length n^2. Note that all the the nonzero blocks besides upper-left "A" block only have nonzero entries that are strictly *behind* the nonzero entries in the upper-left block.

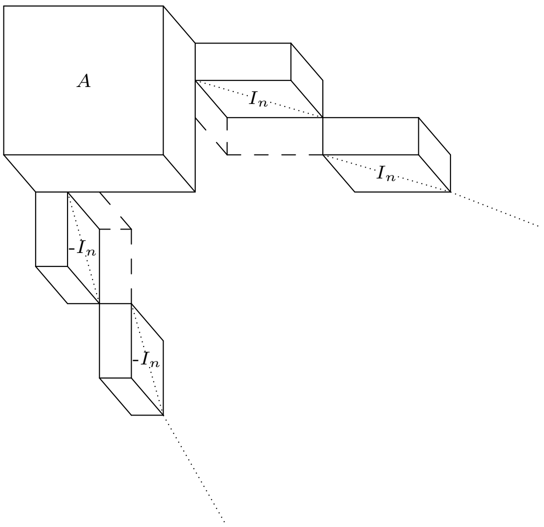


Figure 2 Pictorial representation of the reduction for Lemma 8.

The second viewpoint, which we will also use below, is to consider the lateral slices of $\mathtt{A}$, or equivalently, to view $\mathtt{A}$ from the side. When viewing $\mathtt{A}$ from the side, we see the $(n+1)n \times (m+n^2) \times (n+1)n$ 3-way array:

$$\mathtt{A}^{lat} = \left[\begin{array}{cccc|ccc|c|ccc}
\ell_{1,1} & \ell_{1,2} & \dots & \ell_{1,m} & e_{n+1} & \dots & e_{2n} & \dots & 0 & \dots & 0 \\
\vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\ell_{n,1} & \ell_{n,2} & \dots & \ell_{n,m} & 0 & \dots & 0 & \dots & e_{n^2+1} & \dots & e_{n^2+n} \\
\hline
0 & 0 & \dots & 0 & e_1 & \dots & 0 & \dots & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & 0 & 0 & \dots & e_1 & \dots & 0 & \dots & 0 \\
\hline
\vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\hline
0 & 0 & \dots & 0 & 0 & \dots & 0 & \dots & e_n & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & 0 & 0 & \dots & 0 & \dots & 0 & \dots & e_n
\end{array}\right], \tag{3}$$

where every $\ell_{i,k} \in \mathbb{F}^{n^2+n}$ has only the first n components being possibly non-zero, namely, $\ell_{i,k}(j) = A_k(i,j)$ for $i \in [n], j \in [n], k \in [m]$ and $\ell_{i,k}(j) = 0$ for any $j > n$.

For the only if direction. Suppose there exist $P \in \mathrm{Mon}(n, \mathbb{F})$ and $Q \in \mathrm{GL}(m, \mathbb{F})$, such that $P^t \mathbf{A} P = \mathbf{B}^Q$. We can construct $\tilde{P} \in \mathrm{Mon}(n+n^2, \mathbb{F})$ and $\tilde{Q} \in \mathrm{GL}(m+n^2, \mathbb{F})$ such that $\tilde{P}^t \tilde{\mathbf{A}} \tilde{P} = \tilde{\mathbf{B}}^{\tilde{Q}}$. In fact, we will show that we can take $\tilde{P} = \begin{bmatrix} P & \mathbf{0} \\ \mathbf{0} & P' \end{bmatrix}$ where $P' \in \mathrm{Mon}(n^2, \mathbb{F})$, and $\tilde{Q} = \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & Q' \end{bmatrix}$ where $Q' \in \mathrm{Mon}(n^2, \mathbb{F})$. It is not hard to see that this form already ensures that the first m matrices in the vector $\tilde{P}^t \tilde{\mathbf{A}} \tilde{P}$ and those of $\tilde{\mathbf{B}}^{\tilde{Q}}$ are the same, since when $\tilde{P}, \tilde{Q}$ are of this form, those first m matrices are controlled entirely by the P (resp., Q) in the upper-left block of $\tilde{P}$ (resp., $\tilde{Q}$).

The remaining question is then how to design appropriate P' and Q' to take care of the last n^2 matrices in these tuples. This actually boils down to applying the following simple identity, but "in 3 dimensions:" Let P be the permutation matrix corresponding to $\sigma \in S_n$, so that $Pe_i = e_{\sigma(i)}$, and $e_i^t P = e_{\sigma^{-1}(i)}^t$. Let $D = \mathrm{diag}(\alpha_1, \dots, \alpha_n)$ be a diagonal matrix. Then

$$P^t D P = \mathrm{diag}(\alpha_{\sigma^{-1}(1)}, \dots, \alpha_{\sigma^{-1}(n)}). \tag{4}$$

To see how Equation 4 helps in our setting, it is easier to focus attention on the lower right $n^2 \times n^2$ sub-array of $\mathtt{A}^{lat}$, which can be represented as a symbolic matrix

$$M = \begin{bmatrix} x_1 I_n & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & x_2 I_n & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & x_n I_n \end{bmatrix}.$$

Here we think of the x_i's as independent variables, whose indices correspond to "how far into the page" they are. That is, x_i corresponds to the vector $\vec{e_i}$ in $\mathtt{A}^{lat}$, which is coming out of the page and has its only nonzero entry i slices back from the page.

Then the action of P permutes the x_i's and multiplies them by some scalars, the action of P' is on the left-hand side, and the action of Q' is on the right-hand side. Let σ be the permutation supporting P. Then P sends M to

$$M^P = \begin{bmatrix} \alpha_{\sigma(1)} x_{\sigma(1)} I_n & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \alpha_{\sigma(2)} x_{\sigma(2)} I_n & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \alpha_{\sigma(n)} x_{\sigma(n)} I_n \end{bmatrix}.$$

So setting $P' = \sigma \otimes I_n$, Q' the monomial matrix supported by $\sigma \otimes I_n$ with scalars being $1/\alpha_i$'s, we have $P'^t M^P Q' = M$ by Equation 4.

For the if direction. Suppose there exist $\tilde{P} \in \mathrm{GL}(n + n^2, \mathbb{F})$ and $\tilde{Q} \in \mathrm{GL}(m + n^2, \mathbb{F})$, such that $\tilde{P}^t \tilde{\mathbf{A}} \tilde{P} = \tilde{\mathbf{B}}^{\tilde{Q}}$. The key feature of these gadgets now comes into play: consider the lateral slices of $\tilde{\mathtt{A}}$, which are the frontal slices of $\mathtt{A}^{lat}$ (which may be easier to visualize by looking at Equation 3 and Figure 2). The first n lateral slices of $\tilde{\mathtt{A}}$ and $\tilde{\mathtt{B}}$ are of rank $\geq n$ and $< 2n$, while the other lateral slices are of rank $< n$ (in fact, they are of rank 1; note that without loss of generality we may assume $n > 1$, for the only 1×1 alternating matrix space is the zero space). Furthermore, left multiplying a lateral slice by $\tilde{P}^t$ and right multiplying it by $\tilde{Q}$ does not change its rank. However, the action of $\tilde{P}$ here is by $\tilde{P}^t \tilde{\mathbf{A}} \tilde{P}$, and while the $\tilde{P}^t$ here corresponds to left multiplication on the lateral slices (=frontal slices of $\mathtt{A}^{lat}$), the $\tilde{P}$ on the right here corresponds to taking linear combinations of the lateral slices. In other words, just as $\mathtt{A}^{lat}$ is the "side view" of $\tilde{\mathtt{A}}$, $(\tilde{P}^t \mathtt{A}^{lat} \tilde{Q})^{\tilde{P}}$ is the side view of $(\tilde{P}^t \tilde{\mathtt{A}} \tilde{P})^{\tilde{Q}}$. Taking linear combinations of the lateral slices could, in principle, alter their rank; we will use the latter possibility to show that $\tilde{P}$ must be of a constrained form.

Write $\tilde{P} = \begin{bmatrix} P_{1,1} & P_{1,2} \\ P_{2,1} & P_{2,2} \end{bmatrix}$ where $P_{1,1}$ is of size $n \times n$. We first claim that $P_{1,2} = \mathbf{0}$. For if not, then in $(\mathtt{A}^{lat})^{\tilde{P}}$ (the side view), one of the last n^2 frontal slices receives a nonzero contribution from one of the first n frontal slices of $\mathtt{A}^{lat}$. Looking at the form of these slices from Equation 3, we see that any such nonzero combination will have rank $\geq n$, but this is a contradiction since the corresponding slice in $\mathtt{B}^{lat}$ has rank 1. Thus $P_{1,2} = \mathbf{0}$, and therefore $P_{1,1}$ must be invertible, since $\tilde{P}$ is.

Finally, we claim that $P_{1,1}$ has to be a monomial matrix. If not, then some frontal slice of $(\mathtt{A}^{lat})^{\tilde{P}}$ among the first n would have a contribution from more than one of these n slices. Considering the lower-right $n^2 \times n^2$ sub-matrix of such a slice, we see that it would have rank exactly kn for some $k \geq 2$, which is again a contradiction since the first n slices of $\mathtt{B}^{lat}$ all have rank $< 2n$. It follows that $P_{1,1}^t A_i P_{1,1}$, $i \in [m]$, are in $\mathcal{B}$, and thus $\mathcal{A}$ and $\mathcal{B}$ are monomially isometric via $P_{1,1}$. ◀

4.1.1 Application: reducing Graph Isomorphism to Alternating Matrix Space Isometry

An application of the monomial-restricting gadget is to give an immediate reduction from Graph Isomorphism to Alternating Matrix Space Isometry. While a reduction between these two problems is already known (cf. [32] for details), we choose to present it as an illustration of using this gadget.

▶ **Proposition 9.** Graph Isomorphism *reduces to* Alternating Matrix Space Isometry.

Proof. For a graph $G = ([n], E)$, let $\mathbf{A}_G$ be the alternating matrix tuple $\mathbf{A}_G = (A_1, \dots, A_{|E|})$ with $A_e = E_{i,j} - E_{j,i}$ where $e = \{i, j\} \in E$, and let $\mathcal{A}_G = \langle \mathbf{A}_G \rangle$ be the alternating matrix space spanned by that tuple. If P is a permutation matrix giving an isomorphism between two graphs G and H, then it is easy to see that $P^t \mathcal{A}_G P = \mathcal{A}_H$, and thus the corresponding matrix spaces are isometric. The converse direction is not clear, though it is recently shown to be true in [34] with a rather intricate proof. Instead, we will provide a conceptually simpler proof, by showing that this construction gives a reduction to *monomial* isometry, and then using Lemma 8 to reduce to ordinary Alternating Matrix Space Isometry.

Let us thus establish that the preceding construction gives a reduction from GI to Alternating Matrix Space Monomial Isometry. We will show that $G \cong H$ if and only if $\mathcal{A}_G$ and $\mathcal{A}_H$ are monomially isometric. The forward direction was handled above. For the converse, suppose $P^t D^t \mathcal{A}_G D P = \mathcal{A}_H$ where D is diagonal and P is a permutation matrix. We claim that in this case, P in fact gives an isomorphism from G to H. First let us establish that P alone gives an isometry between $\mathcal{A}_G$ and $\mathcal{A}_H$. Note that for any diagonal matrix $D = \operatorname{diag}(\alpha_1, \dots, \alpha_n)$ and any elementary alternating matrix $E_{i,j} - E_{j,i}$, we have $D^t(E_{i,j} - E_{j,i})D = \alpha_i \alpha_j (E_{i,j} - E_{j,i})$. Since $\mathcal{A}_G$ has a basis of elementary alternating matrices, the action of D on this basis is just to re-scale each basis element, and thus $D^t \mathcal{A}_G D = \mathcal{A}_G$. Thus, we have $P^t \mathcal{A}_G P = \mathcal{A}_H$.

Finally, note that $P^t(E_{i,j} - E_{j,i})P = E_{\pi(i),\pi(j)} - E_{\pi(j),\pi(i)} = A_{\pi(e)}$, where $\pi \in \mathrm{S}_n$ is the permutation corresponding to P, and by abuse of notation we write $\pi(e) = \pi(\{i, j\}) = \{\pi(i), \pi(j)\}$ as well. Since the elementary alternating matrices are linearly independent, and $\mathcal{A}_H$ has a basis of elementary alternating matrices, the only way for $A_{\pi(e)}$ to be in $\mathcal{A}_H$ is for it to be equal to one of the basis elements (one of the matrices in $\mathbf{A}_H$). In other words, $\pi(e)$ must be an edge of H. As P is invertible, we thus have that P gives an isomorphism $G \cong H$. ◀

4.2 Search-to-decision reduction for Alternating Matrix Space Isometry

▶ **Theorem A′.** *Given an oracle deciding* Alternating Matrix Space Isometry*, the task of finding an isometry between two alternating matrix spaces $\mathcal{A}, \mathcal{B} \in \Lambda(n, \mathbb{F}_q)$, if it exists, can be solved using at most $q^{O(n)}$ oracle queries each of size at most $O(n^2)$, and in time either $q^{O(n)} \cdot n! = q^{\tilde{O}(n)}$, or $q^{O(n+m)}$.*

Proof. We first present the gadget construction. Then based on this gadget, we present the search-to-decision reduction.

Gadget construction. Let $\mathbf{A} = (A_1, \dots, A_m)$ be an ordered linear basis of $\mathcal{A}$, and let $\mathtt{A} \in \mathrm{T}(n \times n \times m, \mathbb{F}_q)$ be the 3-way array constructed from $\mathbf{A}$, so we can write

$$\mathtt{A} = \begin{bmatrix} \mathbf{0} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ -a_{1,2} & \mathbf{0} & a_{2,3} & \cdots & a_{2,n} \\ -a_{1,3} & -a_{2,3} & \mathbf{0} & \cdots & a_{3,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -a_{1,n} & -a_{2,n} & -a_{3,n} & \cdots & \mathbf{0} \end{bmatrix},$$

where $a_{i,j} \in \mathbb{F}^m$, $1 \le i < j \le n$ thought of as a vector coming out of the page.

We first consider a 3-way array $\tilde{\mathtt{A}}_i$ constructed from $\mathtt{A}$, for any $1 \leq i \leq n-1$, as $\tilde{\mathtt{A}}_i =$

$$\left[\begin{array}{cccc|ccc|ccc|ccc|ccc|ccc}
\mathbf{0} & a_{1,2} & \cdots & a_{1,i} & a_{1,i+1} & \cdots & a_{1,n} & -e_{1,1} & \cdots & -e_{1,2n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
-a_{1,2} & \mathbf{0} & \cdots & a_{2,i} & a_{2,i+1} & \cdots & a_{2,n} & \mathbf{0} & \cdots & \mathbf{0} & -e_{2,1} & \cdots & -e_{2,2n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
-a_{1,i} & -a_{2,i} & \cdots & \mathbf{0} & a_{i,i+1} & \cdots & a_{i,n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -e_{i,1} & \cdots & -e_{i,2n} & \mathbf{0} & \cdots & \mathbf{0} \\
\hline
-a_{1,i+1} & -a_{2,i+1} & \cdots & -a_{i,i+1} & \mathbf{0} & \cdots & a_{i+1,n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -f_{1,1} & \cdots & -f_{1,n} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
-a_{1,n} & -a_{2,n} & \cdots & -a_{i,n} & -a_{i+1,n} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & -f_{n-i,1} & \cdots & -f_{n-i,n} \\
\hline
e_{1,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
e_{1,2n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\hline
\mathbf{0} & e_{2,1} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & e_{2,2n} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\hline
\mathbf{0} & \mathbf{0} & \cdots & e_{i,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & e_{i,2n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\hline
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & f_{1,1} & \cdots & f_{n-i,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & f_{1,n} & \cdots & f_{n-i,n} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0}
\end{array}\right],$$

where $e_{j,k}$ is the $(m+2n(j-1)+k)$th standard basis vector, and $f_{j,k}$ is the $(m+2ni+n(j-1)+k)$th standard basis vector. A pictorial description can be seen by combining Figure 2 (for the $e_{j,k}$) and [32, Figure 3] (for the $f_{j,k}$).

We claim the following.

▷ Claim 10. If there exist invertible matrices P and Q to satisfy $P^t\tilde{\mathtt{A}}_iP = \tilde{\mathtt{B}}_i^Q$, then P must be in the form $\begin{bmatrix} P_{1,1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & P_{2,2} & \mathbf{0} \\ P_{3,1} & P_{3,2} & P_{3,3} \end{bmatrix}$, where $P_{1,1}$ is a monomial matrix of size $i \times i$, $P_{2,2}$ is of size $(n-i) \times (n-i)$, and $P_{3,3}$ is of size $(2ni+n) \times (2ni+n)$.

Furthermore, there exist such P and Q if and only if $\mathtt{A}$ and $\mathtt{B}$ are isometric by a matrix of the form $\begin{bmatrix} P_{1,1} & \mathbf{0} \\ \mathbf{0} & P_{2,2} \end{bmatrix}$ where $P_{1,1}$ is a monomial matrix of size $i \times i$.

Proof. This claim is immediate by combining the arguments for the FGS gadget [28] as used in [32], and the monomial-restricting gadget introduced in Section 4.1. We only outline the argument and point out some subtle issues here.

First, observe that for the lateral slices of $\tilde{\mathtt{A}}_i$:

- The first i lateral slices have rank in $[2n, 3n)$. Note that the rank is *strictly* less than $3n$ because some tube fibers (coming out of the page) are $\mathbf{0}$ in the upper-left $n \times n$ sub-array.
- The next $n-i$ lateral slices have rank in $[n, 2n)$.
- The remaining $2ni+n$ lateral slices have rank in $[1, n)$ (since $i \geq 1$.)

Because of the above, for P and Q to satisfy $P^t\tilde{\mathtt{A}}_iP = \tilde{\mathtt{B}}_i^Q$, P must be in the required form.

It is the furthermore statement that requires certain care. The only if direction is straightforward: after observing that P has to be of the above form, we can easily verify that $\begin{bmatrix} P_{1,1} & \mathbf{0} \\ \mathbf{0} & P_{2,2} \end{bmatrix}$ is an isometry from $\mathtt{A}$ to $\mathtt{B}$. For the if direction, starting from $\begin{bmatrix} P_{1,1} & \mathbf{0} \\ \mathbf{0} & P_{2,2} \end{bmatrix}$ and $Q_{1,1} \in \mathrm{GL}(m, \mathbb{F})$, we need to design $P_{3,3} \in \mathrm{GL}(2ni+n, \mathbb{F})$ and $Q_{2,2} \in \mathrm{GL}(2ni+n(n-i), \mathbb{F})$ such that letting $P = \begin{bmatrix} P_{1,1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & P_{2,2} & \mathbf{0} \\ 0 & 0 & P_{3,3} \end{bmatrix}$ and $Q = \begin{bmatrix} Q_{1,1} & 0 \\ 0 & Q_{2,2} \end{bmatrix}$, we have $P^t\tilde{\mathtt{A}}_iP = \tilde{\mathtt{B}}_i^Q$. This can be achieved by combining the arguments for the only if directions in the proofs of Lemma 8 and [32, Proposition 3.3]. ◁

The search-to-decision reduction. Given these preparations, we now present the search-to-decision reduction for Alternating Matrix Space Isometry. Recall that this requires us to use the decision oracle $\mathcal{O}$ to compute an explicit isometry transformation $P \in \mathrm{GL}(n,q)$, if $\mathcal{A}$ and $\mathcal{B}$ are indeed isometric. Think of P as sending the standard basis $(\vec{e_1},\dots,\vec{e_n})$ to another basis $(v_1,\dots,v_n)$, where $\vec{e_i}$ and v_i are in $\mathbb{F}_q^n$.

In the first step, we guess v_1, the image of $\vec{e_1}$, and a complement subspace of $\langle v_1\rangle$, at the cost of $q^{O(n)}$. For each such guess, let P_1 be the matrix which sends $\vec{e_1} \mapsto v_1$ and sends $\langle \vec{e_2},\dots,\vec{e_n}\rangle$ to the chosen complementary subspace arbitrarily. We apply P_1 to $\mathtt{A}$, and still call the resulting 3-way array $\mathtt{A}$ in the following. Then construct $\tilde{\mathtt{A}}_1$ and $\tilde{\mathtt{B}}_1$, and feed these two instances to the oracle $\mathcal{O}$. Note that, since $P_{1,1}$ (using notation as above) must be monomial, any equivalence between $\tilde{\mathtt{A}}_1$ and $\tilde{\mathtt{B}}_1$ must preserve our choice of v_1 up to scale. Thus, clearly, if $\mathtt{A}$ and $\mathtt{B}$ are indeed isometric and we guess the correct image of $\vec{e}_1$, then the oracle $\mathcal{O}$ will return yes (and conversely).

In the second step, we guess v_2, the image of $\vec{e}_2$, and a complement subspace of $\langle v_2\rangle$ within $\langle \vec{e_2},\dots,\vec{e_n}\rangle$, at the cost of $q^{O(n)}$. Note here that the previous step guarantees that there is an isometry respecting the direct sum decomposition $\langle v_1\rangle \oplus \langle \vec{e_2},\dots,\vec{e_n}\rangle$, so we need only search for a complement of v_2 within $\langle \vec{e_2},\dots,\vec{e_n}\rangle$, and *not* a more general complement of $\langle v_1, v_2\rangle$ in all of $\mathbb{F}_q^n$. This is crucial for the runtime, as at the $n/2$ step, the latter strategy would result in searching through $q^{\Theta(n^2)}$ possibilities.

For each such guess, we apply the corresponding transformation to $\mathtt{A}$ (and again call the resulting 3-way array $\mathtt{A}$). Then construct $\tilde{\mathtt{A}}_2$ and $\tilde{\mathtt{B}}_2$, and feed these two instances to the oracle $\mathcal{O}$. Clearly, if $\mathcal{A}$ and $\mathcal{B}$ are indeed isometric and we guess the correct image of $\vec{e_2}$ (and $\vec{e_1}$ from the previous step), then the oracle $\mathcal{O}$ will return yes. However, there is a small caveat here, namely we may guess some image of e_2, such that $\mathcal{A}$ and $\mathcal{B}$ are actually isometric by some matrix P of the form $\begin{bmatrix} P_{1,1} & \mathbf{0} \\ \mathbf{0} & P_{2,2}\end{bmatrix}$ where $P_{1,1}$ is a monomial matrix of size 2 (instead of the more desired diagonal matrix). But this is fine, as it still ensures $P_{1,1}$ to be monomial, which is the key property to keep. This means that our choices of $\{v_1, v_2\}$ is correct as a set up to scaling, so we proceed.

In general, in the ith step, we maintain the property that $\mathcal{A}$ and $\mathcal{B}$ are isometric by some $P = \begin{bmatrix} P_{1,1} & \mathbf{0} \\ \mathbf{0} & P_{2,2}\end{bmatrix}$ where $P_{1,1}$ is a monomial matrix of size $(i-1)\times(i-1)$. We guess v_i, the image of $\vec{e_i}$ in $\langle \vec{e_i},\dots,\vec{e_n}\rangle$, and a complement subspace of $\langle v_i\rangle$ within $\langle \vec{e_i},\dots,\vec{e_n}\rangle$. This cost is $q^{O(n)}$. For each such guess, we apply the corresponding transformation to $\mathtt{A}$ (and call the resulting 3-way array $\mathtt{A}$). Then construct $\tilde{\mathtt{A}}_i$ and $\tilde{\mathtt{B}}_i$, and feed these two instances to the oracle $\mathcal{O}$. Once we guess correctly, we ensure that $\mathcal{A}$ and $\mathcal{B}$ are isometric by $P = \begin{bmatrix} P_{1,1} & \mathbf{0} \\ \mathbf{0} & P_{2,2}\end{bmatrix}$ where $P_{1,1}$ is a monomial matrix of size $i \times i$.

So after the $(n-1)$th step, we know that $\mathcal{A}$ and $\mathcal{B}$ are isometric by a monomial transformation. As the number of all monomial transformations is $(q-1)^n \cdot n! \le q^n \cdot 2^{n\log n} = q^{\tilde{O}(n)}$, we can enumerate all monomial transformations and check correspondingly. This gives an algorithm in time $q^{\tilde{O}(n)}$. By resorting to Proposition 11 which solves Alternating Matrix Space Monomial Isometry in time $q^{O(n+m)}$, we have an algorithm in time $q^{O(n+m)}$.

Note that all the instances we feed into the oracle $\mathcal{O}$ are of size $O(n^2)$. This concludes the proof. ◀

4.3 A simply-exponential algorithm for monomial isometry of alternating matrix spaces

We now state the algorithm for monomial isometry used in Theorem A′.

▶ **Proposition 11.** *Let $\mathcal{A}, \mathcal{B} \leq \Lambda(n,q)$ be m-dimensional. Then there exists a $q^{O(n+m)}$-time algorithm that decides whether $\mathcal{A}$ and $\mathcal{B}$ are monomially isometric, and if so, computes an explicit monomial isometry.*

Proof. Let $\mathcal{A}, \mathcal{B} \leq \Lambda(n,q)$ be two m-dimensional alternating matrix spaces. Clearly, by incurring a multiplicative factor of q^n, we can reduce to the problem of testing whether $\mathcal{A}$ and $\mathcal{B}$ are permutationally isometric, i.e. whether there exists a permutation matrix $T \in \mathrm{GL}(n,q)$, such that $T^t \mathcal{A} T = \mathcal{B}$. We will solve this problem in time $2^{O(n)} \cdot q^{O(m)}$. This would give an algorithm with total running time $q^n \cdot 2^{O(n)} \cdot q^{O(m)} = q^{O(n+m)}$. The basic idea of the algorithm comes from Luks's dynamic programming technique for Hypergraph Isomorphism [53].

Reducing to a generalized linear code equivalence problem. Suppose $\mathcal{A} = \langle A_1, \dots, A_m \rangle$, and $\mathcal{B} = \langle B_1, \dots, B_m \rangle$. Let $\mathtt{A}$ and $\mathtt{B}$ be the $n \times n \times m$ 3-way arrays formed by the given bases of $\mathcal{A}$ and $\mathcal{B}$. For $S \subseteq [n]$ of size s, let $(A_i)_S$ be the submatrix of A_i with row and column indices in S. Then let $\mathtt{A}_S$ be the $s \times s \times m$ 3-way array formed by $((A_1)_S, \dots, (A_m)_S)$. Similarly we can define $\mathtt{B}_S$ for $S \subseteq [n]$.

For each $S \subseteq [n]$ of size s, let $\mathrm{Iso}(\mathtt{A}_{[s]}, \mathtt{B}_S)$ be the coset in $\mathrm{S}_n \times \mathrm{GL}(m,q)$, such that $(A,B) \in \mathrm{S}_n \times \mathrm{GL}(m,q)$ if and only if the natural action of (A,B) sends $\mathtt{A}_{[s]}$ to $\mathtt{B}_S$. Since all the matrices are alternating, their diagonal entries are zero, and thus $\mathtt{A}_{\{i\}}$ and $\mathtt{B}_{\{i\}}$ are both the $1 \times 1 \times m$ zero vector for any i. It follows that if $s = 1$ and $S = \{i\}$, $\mathrm{Iso}(\mathtt{A}_{[1]}, \mathtt{B}_S) = G \times \mathrm{GL}(m,q)$, where G is the coset of S_n consisting of permutations sending 1 to i.

Suppose we have computed $\mathrm{Iso}(\mathtt{A}_{[s]}, \mathtt{B}_S)$ for all $s < t$. Fix $T \subseteq [n]$, $|T| = t$, and let us compute $\mathrm{Iso}(\mathtt{A}_{[t]}, \mathtt{B}_T)$. For any $(A,B) \in \mathrm{Iso}(\mathtt{A}_{[t]}, \mathtt{B}_T)$, A sends $[t-1]$ to some $T' \subseteq T$ of size $t-1$. So in this case, $(A,B) \in \mathrm{Iso}(\mathtt{A}_{[t-1]}, \mathtt{B}_{T'})$, which has been computed. Let $T \setminus T' = \{t'\}$. On the other hand, for $(A,B) \in \mathrm{Iso}(\mathtt{A}_{[t-1]}, \mathtt{B}_{T'})$ to be in $\mathrm{Iso}(\mathtt{A}_t, \mathtt{B}_T)$, (A,B) needs to send the tth horizontal slice of $\mathtt{A}_{[t]}$ to the t'th horizontal slice of $\mathtt{B}_T$.

We first identify T' with $[t-1]$. We then note that every horizontal slice of $\mathtt{A}_{[t]}$ has a row of zeros. So the problem now becomes: given two $(t-1) \times m$ matrices P and Q over $\mathbb{F}_q$, decide whether P and Q are the same under $G \leq \mathrm{S}_{t-1} \times \mathrm{GL}(m,q)$. (Note that $G = \mathrm{Iso}(\mathtt{A}_{[t-1]}, \mathtt{B}_{T'})$ from above.) Clearly, this is a generalization of the Linear Code Equivalence problem. Furthermore, if we could solve this problem in time $2^{O(n)} \cdot q^{O(m)}$, we would have achieved our original goal.

Solving the generalized linear code equivalence problem. We solve the above problem again by a dynamic programming scheme as follows. For $R \subseteq [t-1]$ of size r, P_R denotes the $r \times m$ submatrix of P with row indices from R. Let $\mathrm{Iso}'(P_{[r]}, Q_R)$ be the coset in $\mathrm{S}_{t-1} \times \mathrm{GL}(m,q)$, such that $(C,D) \in \mathrm{Iso}'(P_{[r]}, Q_R)$ if and only if the natural action of (C,D) sends $P_{[r]}$ to Q_R. If $r = 0$, then $\mathrm{Iso}'(P_\emptyset, Q_\emptyset) = G$ where $G \leq \mathrm{S}_{t-1} \times \mathrm{GL}(m,q)$ is given as an input.

Suppose we have computed $\mathrm{Iso}'(P_{[r]}, Q_R)$ for any $r < u$. Fix $U \subseteq [t-1]$, $|U| = u$, and let us compute $\mathrm{Iso}'(P_{[u]}, Q_U)$. For any $(C,D) \in \mathrm{Iso}'(P_{[u]}, Q_U)$, C sends $[u-1]$ to some $U' \subseteq U$ of size $u-1$. So in this case, $(A,B) \in \mathrm{Iso}(P_{[u-1]}, Q_U)$, which has been computed. Let $U \setminus U' = \{u'\}$. On the other hand, for $(C,D) \in \mathrm{Iso}(P_{[u-1]}, Q_{U'})$ to be in $\mathrm{Iso}(P_{[u]}, Q_U)$, D

needs to send the uth row of $P_{[u]}$ to the u'th row of Q_U. This subcoset of $\mathrm{Iso}(P_{[u-1]}, Q_{U'})$ can be computed in time $q^{O(m)}$, by treating $\mathrm{GL}(m, q)$ as a permutation group on $\mathbb{F}_q^m$. We then take a union over size-$(u-1)$ subsets U' to obtain a generating set for $\mathrm{Iso}(P_{[u]}, Q_U)$. If necessary, we can reduce the generating set size by applying the standard permutation group machinery, as our time bound is $2^{O(n)} \cdot q^{O(m)}$, which is quite generous. ◀

5 Counting-to-decision reduction by restricting to diagonal groups

In this section, we devise a gadget to achieve the restriction to the group of diagonal matrices, and use it to do the counting to decision reduction for Alternating Matrix Space Isometry.

5.1 Preliminaries

Some preparations are in order.

▶ **Observation 12.** *Let $n \geq 23$. Then any permutation $\sigma \in \mathrm{S}_n$ either fixes a set of 6 points $P \subseteq [n]$, or moves a set of 6 points $P \subseteq [n]$ to another set of 6 points $Q \subseteq [n]$ such that these two sets are disjoint.*

Proof. Suppose σ fixes at most 5 points. Then there are at least 18 points that are not fixed by σ. Suppose σ has t non-trivial cycles of length $l_1, \dots, l_t$, such that $\sum_i l_i \geq 18$. For a cycle $(p_1, \dots, p_s)$, we can choose $p_1, p_3, \dots, p_{2\cdot\lfloor s/2\rfloor-1}$ and put them in P, and $p_2, p_4, \dots, p_{2\cdot\lfloor s/2\rfloor}$ in Q. Do this for every cycle, we obtain the desired P and Q. The worst case is when every cycle is of length 3. Since there are at least 18 points not fixed by σ, P is of size ≥ 6. ◀

We shall make repeated uses of the following facts.

▶ **Fact 13.**

1. *Given $a_i \in \mathbb{R}$, $0 \leq a_i \leq 1$, $i \in [m]$, $\prod_{i\in[m]}(1-a_i) \geq 1 - \sum_{i\in[m]} a_i$.*
2. *Let $m, N \in \mathbb{N}$ and $1 \leq m \leq N$. A random matrix $A \in \mathrm{M}(N \times m, q)$ is of rank m with probability $\geq 1 - 2/q^{N-m+1}$.*
3. *For $n \in \mathbb{N}$, $0 \leq d \leq n$, the number of dimension-d subspaces of $\mathbb{F}_q^n$ is equal to the Gaussian binomial coefficient*
$$\binom{n}{d}_q := \frac{(q^n-1)\cdot(q^n-q)\cdot \dots \cdot (q^n-q^{d-1})}{(q^d-1)\cdot(q^d-q)\cdot \dots \cdot (q^d-q^{d-1})}.$$
4. *The Gaussian binomial coefficient satisfies:*
$$q^{(n-d)d} \leq \binom{n}{d}_q \leq q^{(n-d)d+d}.$$
5. *For $d \in \mathbb{N}$, the number of complement subspaces of a fixed dimension-d subspace of $\mathbb{F}_q^n$ is $q^{d(n-d)}$.*

Proof. For (2), $\Pr[\mathrm{rk}(A) = m] = (1-\frac{1}{q^N})\cdot(1-\frac{q}{q^N})\cdot \dots \cdot (1-\frac{q^{m-1}}{q^N})$. By (1), we have $\Pr[\mathrm{rk}(A) = m] \geq 1 - \sum_{i=N-m+1}^{N} \frac{1}{q^i} = 1 - \frac{1}{q^{N-m+1}} - \sum_{i=N-m+2}^{N} \frac{1}{q^i} \geq 1 - \frac{2}{q^{N-m+1}}$. ◀

5.2 Describing the gadget

Let $\mathcal{A} \leq \Lambda(n, q)$ be an alternating matrix space, and let $\mathbf{A} = (A_1, \dots, A_m) \in \Lambda(n, q)^m$ be an ordered linear basis of $\mathcal{A}$. Let $\mathtt{A} \in \mathrm{T}(n \times n \times m, \mathbb{F}_q)$ be the 3-way array constructed from $\mathbf{A}$, i.e. the ith frontal slice of $\mathtt{A}$ is A_i.

We shall assume $n = \Omega(1)$, and $q = n^{\Omega(1)}$ throughout the remainder of this section.

The form of the gadget. To describe the gadget, it is easier to view $\mathtt{A}$ from the lateral viewpoint. That is, for $i \in [n]$, let $C_i = [A_1 e_i, \dots, A_m e_i] \in \mathrm{M}(n \times m, q)$. Let $\mathbf{C} = (C_1, \dots, C_n) \in \mathrm{M}(n \times m, q)^n$. Then construct $\mathbf{C}' = (C'_1, \dots, C'_n)$, $C'_i = \begin{bmatrix} C_i & 0 \\ 0 & G_i \end{bmatrix}$, where G_i is of size $6n \times 4n^2$. For $i \in [n]$, $G_i = \begin{bmatrix} 0 & \dots & 0 & H_i & 0 & \dots & 0 \end{bmatrix}$, where H_i is of size $6n \times 4n$ in the ith block, and 0 denotes an all-zero matrix of size $6n \times 4n$. The H_i will be described below.

Note that from the frontal viewpoint of looking at $\mathtt{A}$, G_i's are inserted, vertically, below and behind $\mathtt{A}$. So to preserve the alternating structure, $-G_i$'s also need to be inserted, horizontally, on the right and behind $\mathtt{A}$. We therefore get $\tilde{\mathtt{A}}$, which is of size $7n \times 7n \times (m + 4n^2)$.

Conditions imposed on the H_i's. Of course, the key to the construction above lies in the properties of the H_i's. Let $V_i \le \mathbb{F}_q^{6n}$ be the subspace spanned by the columns of H_i. We shall impose the following conditions on H_i.

1. For any $i \in [n]$, $\mathrm{rk}(H_i) = \dim(V_i) = 4n$.
2. For any $i, j \in [n]$, $i \neq j$, $\mathrm{rk}([H_i H_j]) = \dim(V_i \cup V_j) = 6n$.
3. For any $(i_1, i_2, i_3, i_4, i_5, i_6) \in [n]^6$ and $(j_1, j_2, j_3, j_4, j_5, j_6) \in [n]^6$, such that $|\{i_1, \dots i_6\} \cup \{j_1, \dots, j_6\}| = 12$, i.e. i_k and j_ℓ all different, the coset $C = \{T \in \mathrm{GL}(6n, q) : \forall k \in [6], T(V_{i_k}) = V_{j_k}\}$ is empty. Note that for any $i \in [n]$, $T(V_i)$ is spanned by the columns of TH_i.
4. For any $(i_1, i_2, i_3, i_4, i_5, i_6) \in [n]^6$, i_k all different, the group $S = \{T \in \mathrm{GL}(6n, q) : \forall k \in [6], T(V_{i_k}) = V_{i_k}\}$ consists of only of scalar matrices.

▶ Remark 14. Given $H_1, \dots, H_n \in \mathrm{M}(6n \times 4n, q)$, whether they satisfy the four conditions can be verified in polynomial time.

Conditions (1) and (2) are easily verified in deterministic polynomial time.

For condition (3), it can be formulated as a linear algebraic problem as follows. Let X be a $6n \times 6n$ variable matrix. Let Y_k, $k \in [6]$, be $4n \times 4n$ variable matrices. Set up the equations $XH_{i_k} = H_{j_k} Y_k$, and solve the linear equations to get a subspace of $\mathbb{F}_q^{(6n)^2 + 6 \cdot (4n)^2}$. The question is then whether this subspace contains $(T, R_1, \dots, R_6)$ where $T \in \mathrm{GL}(6n, q)$ and $R_i \in \mathrm{GL}(4n, q)$. This is an instance of the symbolic determinant identity testing (SDIT) problem, so it admits a randomized efficient algorithm when $q = n^{\Omega(1)}$.

In fact, this instance of SDIT problem can be solved in deterministic polynomial time. For this let us also check out condition (4). Here, let X and Y_i be from above, and set up the equations $XH_{i_k} = H_{i_k} Y_k$. Solve the linear equations to get a subspace of $\mathbb{F}_q^{(6n)^2 + 6 \cdot (4n)^2}$. This subspace turns out to be an algebra under the natural multiplications. Indeed, if $AH_{i_k} = H_{i_k} B_k$ and $A' H_{i_k} = H_{i_k} B'_k$, then $AA' H_{i_k} = H_{i_k} B_k B'_k$. To compute the unit group in a matrix algebra can be solved by a polynomial-time Las Vegas algorithm by [16]. Given the unit group, whether it consists of only scalar matrices can be verified easily in deterministic polynomial time.

Then the linear space in condition (3) is a module over the algebra defined in the last paragraph. Because of this structure, the SDIT problem for such instances can be solved in deterministic polynomial time [14, 19, 37].

5.3 Construction and properties of the gadget

The following three propositions reveal the construction and functions of the gadget described above.

First about the construction. Instead of constructing the above H_i's explicitly in a deterministic way, we shall show that random choices suffice.

▶ **Proposition 15.** *Let $H_i \in \mathrm{M}(6n \times 4n, q)$, $i \in [n]$, be random matrices. Then H_i's satisfy the four conditions in Section 5.2 with probability $\geq 1 - \frac{n^{O(1)}}{q^{\Omega(1)}}$.*

Second about the functionality. The following proposition formally explains this.

▶ **Proposition 16.** *Suppose $\mathtt{A}$ and $\mathtt{B}$ are two 3-tensors constructed from ordered bases of m-dimensional alternating matrix spaces $\mathcal{A}, \mathcal{B} \leq \Lambda(n, q)$. Let $\tilde{\mathtt{A}}$ and $\tilde{\mathtt{B}}$ be constructed as above, and let $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ be the alternating matrix spaces spanned by the frontal slices of $\tilde{\mathtt{A}}$ and $\tilde{\mathtt{B}}$, respectively. Then $\mathcal{A}$ and $\mathcal{B}$ are isometric via a diagonal matrix if and only if $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are isometric.*

Finally we shall use this gadget to achieve a counting-to-decision reduction for Alternating Matrix Space Isometry. Formally, we have the following.

▶ **Proposition 17.** *Suppose we are given $\mathcal{A}, \mathcal{B} \leq \Lambda(n, q)$ and a decision oracle for* Alternating Matrix Space Isometry*. Then there exists a Las Vegas randomized algorithm that computes the number of isometries from $\mathcal{A}$ to $\mathcal{B}$ in time $q^{O(n)}$.*

The next three subsections are devoted to the proofs of Propositions 15 (Section 5.3.3), 16 (Section 5.3.1), and 17 (Section 5.3.2). Note that, because the proof of Proposition 15 is more complicated compared to the other two, we postpone it to the last.

▶ Remark 18. In fact, we expect that this construction works even for small finite fields. The bottleneck lies in Proposition 15. If the probability $\frac{n^{O(1)}}{q^{\Omega(1)}}$ could be improved to $\frac{n^{O(1)}}{q^{\Omega(n)}}$, then we would be done. We believe it possible to utilize the structure of invariant subspaces under matrix actions over $\mathbb{F}_q$ to achieve this. However, we expect that the calculations will be tedious and heavy, so we hope to leave this to a future work.

5.3.1 Restricting to the diagonal group

Briefly speaking, conditions 1 and 2 ensure that we first restrict to monomial matrices. Conditions 3 and 4 prevent non-trivial permutations due to the following. As we assume $n = \Omega(1)$, by Observation 12, $\sigma \in \mathrm{S}_n$ either fixes 6 elements in $[n]$, or moves a set of 6 elements to another, disjoint, set of 6 elements. Condition 3 ensures that the second case could not happen. Condition 4 ensures that in the first case, the only possible invertible matrices that "preserves" the matrices G_i for $i \in P$ when multiplying from the left are scalar matrices.

We now prove Proposition 16.

Proof of Proposition 16. Recall that we construct such $\tilde{\mathtt{A}}$ and $\tilde{\mathtt{B}}$ from $\mathtt{A}$ and $\mathtt{B}$, respectively, using the method in Section 5.2. Let $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ be alternating matrix spaces in $\Lambda(7n, q)$, spanned by the frontal slices of $\tilde{\mathtt{A}}$ and $\tilde{\mathtt{B}}$, respectively.

We want to show that $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are isometric if and only if $\mathcal{A}$ and $\mathcal{B}$ are isometric via diagonal matrices. The if direction is straightforward. Suppose there exist $P = \mathrm{diag}(\alpha_1, \dots, \alpha_n) \in \mathrm{diag}(n, q)$ and $Q \in \mathrm{GL}(m, q)$ such that $P^t \mathtt{A} P = \mathtt{B}^Q$. Let $\tilde{P} = \begin{bmatrix} P & 0 \\ 0 & I_{6n} \end{bmatrix} \in \mathrm{GL}(7n, q)$. Let $\tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & Q' \end{bmatrix} \in \mathrm{GL}(m + 4n^2)$, where $Q' = \mathrm{diag}(\alpha_1 I_{4n}, \dots, \alpha_n I_{4n})$. Then it is easy to verify that $\tilde{P}^t \tilde{\mathtt{A}} \tilde{P} = \tilde{\mathtt{B}}^{\tilde{Q}}$.

Now we turn to the only if direction. If $\tilde{\mathtt{A}}$ and $\tilde{\mathcal{B}}$ are isometric, then there exists $\tilde{P} \in \mathrm{GL}(7n, q)$ and $\tilde{Q} \in \mathrm{GL}(m + 4n^2, q)$, such that $\tilde{P}^t \tilde{\mathtt{A}} \tilde{P} = \tilde{\mathtt{B}}^{\tilde{Q}}$. Let $\tilde{P} = \begin{bmatrix} P_{1,1} & P_{1,2} \\ P_{2,1} & P_{2,2} \end{bmatrix}$, where $P_{1,1}$ is of size $n \times n$. It can be checked easily, from the lateral viewpoint, that $P_{1,2} = 0$. As if not, then some H_i would appear in one of the last $6n$ lateral slices in $\tilde{\mathtt{A}}\tilde{P}$. This would set this slice to be of rank $\geq 4n$ by condition (1), which contradicts that the corresponding lateral slice of $\tilde{\mathtt{B}}^{\tilde{Q}}$ is of rank $\leq n$. It follows that $P_{1,1} \in \mathrm{GL}(n, q)$ and $P_{2,2} \in \mathrm{GL}(6n, q)$.

We first claim that $P_{1,1}$ has to be a monomial matrix. If not, then one of the first n lateral slice of $\tilde{\mathtt{A}}\tilde{P}$ has two distinct H_i and H_j. By condition (2), this slice is of rank $\geq 6n$, which contradicts that the corresponding lateral slice of $\tilde{\mathtt{B}}^{\tilde{Q}}$ is of rank $\leq 5n$.

We further claim that $P_{1,1}$ has to be a diagonal matrix. If not, then suppose the non-trivial permutation underlying $P_{1,1}$ is $\sigma \in \mathrm{S}_n$. Since we assumed $n = \Omega(1)$, by Observation 12, one of the following two cases has to happen.

- $\exists\{i_1, \dots, i_6\} \subseteq [n]$, $\{j_1, \dots, j_6\} \subseteq [n]$, $|\{i_1, \dots, i_6\} \cup \{j_1, \dots, j_6\}| = 12$, such that $\sigma(i_k) = j_k$ for $k \in [6]$. We then claim the following.

  ▷ Claim 19. For $\tilde{P}^t\tilde{\mathtt{A}}\tilde{P} = \tilde{\mathtt{B}}^{\tilde{Q}}$ to hold, a necessary condition is that $\forall k \in [6]$, $P_{2,2}H_{j_k}$ and H_{i_k} have the same linear span.

  Proof. To see this, note that the i_kth lateral slice of $\tilde{P}^t\tilde{\mathtt{A}}\tilde{P}$ is the j_kth lateral slice of $\tilde{P}^t\tilde{\mathtt{A}}$ (up to a scalar multiple). It is equal to the i_kth lateral slice of $\tilde{\mathtt{B}}^{\tilde{Q}}$. Then $\tilde{P}^t$ acts on the left on the j_kth lateral slice of $\tilde{\mathtt{A}}$. Noting that $P^t = \begin{bmatrix} P_{1,1}^t & P_{2,1}^t \\ 0 & P_{2,2}^t \end{bmatrix}$ and the j_kth lateral slice of $\tilde{\mathtt{A}}$ is $C'_{j_k} = \begin{bmatrix} C_{j_k} & 0 \\ 0 & G_{j_k} \end{bmatrix}$, we see that $P^tC'_{j_k} = \begin{bmatrix} * & * \\ 0 & P_{2,2}^t G_{j_k} \end{bmatrix}$. (Here, C_i and G_i are defined in Section 5.2.) On the other hand, we see that the i_kth lateral slice of $\tilde{\mathtt{B}}^{\tilde{Q}}$ is the i_kth lateral slice multiplied from the right by $\tilde{Q}$. Our claim follows then by comparing the last $6n$ rows. ◁

  But the condition (3) excludes the existence of such $P_{2,2}$, so this cannot happen.
- $\exists\{i_1, \dots, i_6\} \subseteq [n]$, i_k all different, such that $\sigma(i_k) = i_k$. In this case, for $\tilde{P}^t\tilde{\mathtt{A}}\tilde{P} = \tilde{\mathtt{B}}^{\tilde{Q}}$ to hold, by the same argument as in the proof of Claim 19, a necessary condition is that $P_{2,2}H_{i_k}$ and H_{i_k} have the same linear span. Then the condition (4) ensures that $P_{2,2} = \lambda I_{6n}$ for some $\lambda \neq 0 \in \mathbb{F}$ in this setting. Then because σ is non-trivial, σ moves some $i \in [n]$ to $j \in [n]$, $i \neq j$. By comparing the jth lateral slice of $\tilde{P}^t\tilde{\mathtt{A}}$ and the ith lateral slice of $\tilde{\mathtt{B}}^{\tilde{Q}}$, $P_{2,2}H_i = \lambda H_i$ and H_j have the same linear span, which is not possible because the condition (2) ensures that H_i and H_j span different subspaces.

We then have shown that $P_{1,1}$ must be a diagonal matrix. By comparing the top-left-front sub-tensors of size $n \times n \times m$ of $\tilde{P}^t\tilde{\mathtt{A}}\tilde{P}$ and $\tilde{\mathtt{B}}^{\tilde{Q}}$, we arrive at the desired conclusion that $\mathcal{A}$ and $\mathcal{B}$ are isometric via the diagonal matrix $P_{1,1}$. ◀

5.3.2 Using the gadget for counting-to-decision reduction

The strategy follows closely the counting to decision reduction for graph isomorphism.

We first review the strategy for counting to decision reduction for graph isomorphism [54]. Suppose we are given two graphs with the vertex set being $[n]$, i.e. $G, H \subseteq \binom{[n]}{2}$. We first use the decision oracle to decide whether G and H are isomorphic. If not, the number of isomorphisms is 0. If so, we turn to compute the order of $\mathrm{Aut}(G)$. Let $A = \mathrm{Aut}(G)$. For $i \in [n]$, let $A_i = \{\sigma \in A : \forall 1 \leq j \leq i, \sigma(j) = j\}$. Set $A_0 = A$. We then have the tower of subgroups $A_0 \geq A_1 \geq \cdots \geq A_n = \{\mathrm{id}\}$. The order of A_0 is then the product of $[A_i : A_{i+1}]$, the index of A_{i+1} in A_i, for $i = 0, 1, \dots, n-1$. Let G_i be the graph with the first i vertices in G individualized. Then $\mathrm{Aut}(G_i) \cong A_i$. To compute $[A_i : A_{i+1}]$, we note that it is equal to the size of the orbit of the vertex $i+1$ under A_i. For each $j \geq i+1$, construct from G_i two graphs G_i' and G_i'' as follows. In G_i', individualize $i+1$, and in G_i'', individualize j. Then j is in the orbit of $i+1$ under A_i if and only if G_i' and G_i'' are isomorphic. Enumerating over $j \geq i+1$ gives us the size of the orbit of $i+1$ under A_i. This finishes an overview of the idea for counting to decision reduction for graph isomorphism.

We then apply the above strategy to get a counting to decision reduction for alternating matrix space isometry to prove Proposition 17.

Proof of Proposition 17. Our goal is to compute the number of isomorphisms from $\mathcal{A}$ to $\mathcal{B}$, where $\mathcal{A}, \mathcal{B} \leq \Lambda(n,q)$ are of dimension m. First, we use the decision oracle first to decide whether $\mathcal{A}$ and $\mathcal{B}$ are isometric. If not, the number of isometries is 0. If so, we need to caculate the order of the autometry group of $\mathcal{A}$, $\mathrm{Aut}(\mathcal{A})$. To do that, we first randomly sample n $6n \times 4n$ matrices $H_1, \dots, H_n$ over $\mathbb{F}_q$, and verify whether they satisfy the four conditions in Section 5.2 using Remark 14. Note that this is where the algorithm needs to be a Las Vegas algorithm.

Let $A = \mathrm{Aut}(\mathcal{A})$. Recall that e_i denotes the ith standard basis vector in $\mathbb{F}_q^n$. For $i \in [n]$, let $A_i = \{T \in A : \forall 1 \leq j \leq i, T(e_i) = \lambda_i e_i, \lambda_i \neq 0 \in \mathbb{F}_q\}$. Note that $A_n = A \cap \mathrm{diag}(n,q)$. We can calculate the order of A_n in time $q^{O(n)}$ by brute-force, i.e., enumerating all invertible diagonal matrices. Set $A_0 = A$. We then have the tower of subgroups $A_0 \geq A_1 \geq \cdots \geq A_n$.

To compute the order of A_0, it is enough to compute $[A_i : A_{i+1}]$. Note that for $T, T' \in A_i$, $TA_{i+1} = T'A_{i+1}$ as left cosets in A_i if and only if $T(e_{i+1}) = \lambda T'(e_{i+1})$ for some $\lambda \neq 0 \in \mathbb{F}_q$. So $[A_i : A_{i+1}]$ is equal to the size of the orbit of e_{i+1} under A_i in the projective space. Let $v \in \mathbb{F}_q^n$. To test whether v is in the orbit of e_{i+1} under A_i in the projective space, we tranform $\mathcal{A}$ by $P^t \cdot P$, where $P \in \mathrm{GL}(n,q)$ sends e_{i+1} to v and e_j to e_j for $j \neq i+1$, to get $\mathcal{A}'$. We then add the diagonal restriction gadget to the first $i+1$ lateral slices and the first $i+1$ horizontal slices of $\mathcal{A}$ and $\mathcal{A}'$, to obtain $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{A}}'$ respectively. Then feed $\mathcal{A}$ and $\mathcal{A}'$ to the decision oracle. By the functionality of the diagonal restriction gadget, v is in the orbit of e_{i+1} in the projective space if and only if $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{A}}'$ are isometric. Enumerating $v \in \mathbb{F}_q^n$ up to scalar multiples gives us the size of the orbit of e_{i+1} under A_i in the projective space. This finishes the description of the algorithm.

A small caveat in the above is that our gadget requires $n = \Omega(1)$, so we cannot start from A_0 at the beginning. This issue can be revolved by noting that the order of A_c, for any constant c, can be computed in time $q^{O(n)}$, by enumerating all possible images of $e_1, \dots, e_c$ in time $q^{O(n)}$, adding the diagonal restriction gadget, and utilizing the decision oracle. ◀

5.3.3 Random H_i's satisfy the requirements when $q = n^{\Omega(1)}$

In the following we will encounter random matrices over $\mathbb{F}_q$ as well as random subspaces in $\mathbb{F}_q^n$. There is a subtle point which we want to clarify now. Let $m \leq n$. Note that there are $\binom{n}{m}_q$ subspaces of $\mathbb{F}_q^n$, and there are $N_1 = (q^n - 1) \cdot \ldots \cdot (q^n - q^{m-1})$ rank-m matrices of size $n \times m$. It can be seen easily that each m-dimensional subspace V of $\mathbb{F}_q^n$ has $N_2 = (q^m - 1) \cdot \ldots \cdot (q^m - q^{m-1})$ many representations as rank-m matrices of size $n \times m$, i.e. the columns of the matrix span V. It follows that we can work with random rank-m matrices of size $n \times m$ as if we are working with random m-dimensional subspaces of $\mathbb{F}_q^n$. Such correspondences will be used implicitly for other structures, including direct sum decompositions.

Now let us get back to our question. We shall show that a random choice of H_i, $i \in [n]$, would satisfy the four conditions we imposed on H_i's. We will prove that for conditions $k = 1, 2, 3$,

$$\Pr[\text{random } H_i \text{ not satisfy condition } k] \leq \frac{n^{O(1)}}{q^{\Omega(n)}}.$$

Once these hold, by a union bound, we have

$$\Pr[\exists i \in [3], \text{random } H_i \text{ not satisfy condition } i] \leq \frac{n^{O(1)}}{q^{\Omega(n)}}.$$

For condition (4), we will prove that

$$\Pr[\text{random } H_i \text{ not satisfy condition 4} \mid H_i \text{ satisfy conditions } 1, 2, 3] \leq \frac{n^{O(1)}}{q^{\Omega(1)}}.$$

This then would allow us to conclude that when $q = n^{\Omega(1)}$, random H_i's satisfy all the four conditions.

We examine the first three conditions one by one.

1. For condition (1), by Fact 13 (2), we have $\Pr[\exists i \in [n], \mathrm{rk}(H_i) < 4n] \leq n \cdot \Pr[\mathrm{rk}(H_i) < 4n] \leq \frac{2n}{q^{2n+1}}$.
2. For condition (2), noting that the block matrix $(H_i H_j)$ is a random $6n \times 8n$ matrix over $\mathbb{F}_q$, by Fact 13 (2), we have $\Pr[\exists i \neq j \in [n], \mathrm{rk}((H_i H_j)) < 6n] \leq \binom{n}{2} \cdot \frac{2}{q^{8n-6n+1}} \leq \frac{n^2}{q^{2n+1}}$.
3. For condition (3), let $I = (H_{i_1} \dots H_{i_6})$, and $J = (H_{j_1} \dots H_{j_6})$. We see that C is non-empty if and only if there exists $L \in \mathrm{GL}(6n, q)$ and $R_k \in \mathrm{GL}(4n, q)$, $k \in [6]$, such that $LH_{i_k}R_k = H_{j_k}$. Note that the orbit of I under this group action is of size at most $q^{(6n)^2+6\cdot(4n)^2} = q^{132n^2}$. Since i_k and j_ℓ are all different, the probability of J belonging to this orbit is $\leq \frac{q^{132n^2}}{q^{144n^2}} = \frac{1}{q^{12n^2}}$. We then have $\Pr[\exists i_k, j_k \in [n], k \in [6], i_k, j_k \text{ all different}, C = \emptyset] \leq \binom{n}{12}\frac{2}{q^{12n^2}} \leq \frac{n^{12}}{q^{12n^2}}$.

We now focus on condition (4). For condition (4), we first assume that the conditions (1) and (2) as above hold. Then V_i's are random $4n$-dimensional subspaces of $\mathbb{F}_q^{6n}$. Note that

$$\Pr[\exists i_k \in [n], k \in [6], i_k \text{ all different}, S \text{ non-scalar}] \leq n^6 \cdot \Pr[S \text{ non-scalar for } V_1, \dots, V_6].$$

So we turn to study $\Pr[S \text{ non-scalar for } V_1, \dots, V_6]$, and will show that it is $\leq \frac{1}{q^{\Omega(1)}}$.

Let $U_1 = V_1 \cap V_2$, $U_2 = V_2 \cap V_3$, and $U_3 = V_1 \cap V_3$. Let $W_1 = V_4 \cap V_5$, $W_2 = V_5 \cap V_6$, and $W_3 = V_4 \cap V_6$. Since conditions (1) and (2) hold, we have $\dim(U_i) = \dim(W_i) = 2n$. We claim that with probability $\geq 1 - 2/q$, $\mathbb{F}_q^{6n} = U_1 \oplus U_2 \oplus U_3$, i.e., $U_1 \cup U_2 \cup U_3$ span $\mathbb{F}_q^{6n}$. This can be seen as follows. Since we assumed conditions (1) and (2), this happens if and only if $V_1 \cap V_2$ and V_3 together span $\mathbb{F}_q^{6n}$. Therefore we calculate, using Fact 13 (1), (3), and (5), that

$$\begin{aligned}
& \Pr[V_3 \text{ is a complement subspace of } V_1 \cap V_2] \\
= \; & q^{2n\cdot 4n} / \binom{6n}{4n}_q = \frac{(q^{6n} - q^{2n})(q^{6n} - q^{2n+1}) \dots (q^{6n} - q^{6n-1})}{(q^{6n} - 1)(q^{6n} - q) \dots (q^{6n} - q^{4n-1})} \\
\geq \; & \frac{(q^{6n} - q^{2n})(q^{6n} - q^{2n+1}) \dots (q^{6n} - q^{6n-1})}{q^{6n} \cdot q^{6n} \cdot \dots \cdot q^{6n}} = (1 - 1/q^{4n})(1 - 1/q^{4n-1}) \dots (1 - 1/q) \\
\geq \; & 1 - \sum_{i=1}^{4n} 1/q^i \geq 1 - 2/q.
\end{aligned}$$

It follows that with probability $\geq 1 - 4/q$, we can assume in addition that W_i form a direct sum decomposition of $\mathbb{F}_q^{6n}$.

Therefore, we turn to bound the probability that there exists a non-scalar invertible matrix stabilizing these two direct sum decompositions of $\mathbb{F}_q^{6n}$. Since i_k are all different, the two direct sum decompositions $U_1 \oplus U_2 \oplus U_3$ and $W_1 \oplus W_2 \oplus W_3$ are independent.

So we can assume that U_i is spanned by those standard basis vectors $e_{2n(i\vec{-}1)+1}, \dots, e_{2\vec{n}i}$, $i = 1, 2, 3$. The group that stabilizes this direct sum decomposition $U_1 \oplus U_2 \oplus U_3$ consists of $\begin{bmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{bmatrix} \in \mathrm{GL}(6n, \mathbb{F}_q)$ where D_i is of size $2n \times 2n$.

The question then becomes to bound the probability for a random $W_1 \oplus W_2 \oplus W_3$ to be stabilized by a non-scalar matrix of the above form. This can be formulated as the following linear algebraic problem. (Recall the correspondence between random m-dimensional subspaces and random rank-m matrices as discussed at the beginning of the subsection.) Let $W = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix} \in \mathrm{GL}(6n, q)$ be a block matrix where W_{ij} is of size $2n \times 2n$. Suppose the columns of $\begin{bmatrix} W_{1i} \\ W_{2i} \\ W_{3i} \end{bmatrix}$ span W_i. Then $D = \mathrm{diag}(D_1, D_2, D_3)$ stabilizes $W_1 \oplus W_2 \oplus W_3$ if and only if there exists a block diagonal matrix $E = \mathrm{diag}(E_1, E_2, E_3)$, $E_i \in \mathrm{GL}(2n, q)$, such that

$$\begin{bmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{bmatrix} \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix} \begin{bmatrix} E_1 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & E_3 \end{bmatrix}. \tag{5}$$

Note that each direct sum decomposition $W_1 \oplus W_2 \oplus W_3$, $\dim(W_i) = 2n$, has $6 \cdot |\mathrm{GL}(2n, q)|^3$ such matrix representations. (The factor 6 takes care of the orders of the three summands.) So the question becomes to bound the probability for a random invertible matrix to have a non-scalar D and E satisfying Equation 5.

First, note that Equation 5 holds if and only if $D_i W_{i,j} = W_{i,j} E_j$ for $i, j \in [3]$.

▷ Claim 20. When $q = \Omega(1)$, we have $\Pr[\forall i, j \in [3], \mathrm{rk}(W_{i,j}) = 2n] \geq 1 - \frac{20}{q}$.

Proof. Let us work in the setting when W is a random matrix, not necessarily the one above. Then $\Pr[\mathrm{rk}(W) = 6n] \geq 1 - \frac{2}{q}$. For any $i, j \in [3]$, $\Pr[\mathrm{rk}(W_{i,j}) < 2n] \leq \frac{2}{q}$, so $\Pr[\exists i, j \in [3], \mathrm{rk}(W_{i,j}) < 2n] \leq \frac{18}{q}$. It follows that $\Pr[\exists i, j \in [3], \mathrm{rk}(W_{i,j}) < 2n \mid \mathrm{rk}(W) = 6n] = \Pr[\exists i, j \in [3], \mathrm{rk}(W_{i,j}) < 2n \wedge \mathrm{rk}(W) = 6n] / \Pr[\mathrm{rk}(W) = 6n] \leq \frac{18/q}{1-2/q} = \frac{18}{q-2} \leq \frac{20}{q}$, where the last inequality uses that $q = \Omega(1)$. ◁

So we assume that $\mathrm{rk}(W_{i,j}) = 2n$ for all $i, j \in [3]$ in the following, with a loss of probability $\leq \frac{20}{q}$.

For $i \in [3]$, by $D_i W_{ii} = W_{ii} E_i$, we have $D_i = W_{ii} E_i W_{ii}^{-1}$. For $i \neq j$, by $(W_{jj} E_j W_{jj}^{-1}) W_{ji} = D_j W_{ji} = W_{ji} E_i$, we have $E_j = W_{jj}^{-1} W_{ji} E_i W_{ji}^{-1} W_{jj}$. Again for $i \neq j$, we have $W_{ii} E_i W_{ii}^{-1} W_{ij} = D_i W_{ij} = W_{ij} E_j = W_{ij} W_{jj}^{-1} W_{ji} E_i W_{ji}^{-1} W_{jj}$. It follows that

$$\forall i, j \in [3], i \neq j, E_i W_{ii}^{-1} W_{ij} W_{jj}^{-1} W_{ji} = W_{ii}^{-1} W_{ij} W_{jj}^{-1} W_{ji} E_i.$$

In particular, E_3 commutes with $X = W_{33}^{-1} W_{32} W_{22}^{-1} W_{23}$ and $Y = W_{33}^{-1} W_{31} W_{11}^{-1} W_{13}$. Since W_{ij} are independent random invertible matrices, X and Y are independent random invertible matrices. We now resort to the following classical result.

▶ **Theorem 21** ([41], cf. also [23, 40]). *Let X and Y be two random matrices in $\mathrm{SL}(n, q)$. Then the probability of X and Y not generating $\mathrm{SL}(n, q)$ is $\leq \frac{1}{q^{\Omega(n)}}$.*

Back to our setting, the above theorem implies that the group G generated by random X and Y from $\mathrm{GL}(2n,q)$ contains $\mathrm{SL}(2n,q)$ with probability $\geq 1-\frac{1}{q^{\Omega(n)}}$. It follows that E_3 belongs to the centralizer of G, so E_3 must be a scalar matrix. Then note that D_i's and other E_i's are all conjugates of E_3. So we have $\forall i \in [3]$, $D_i = E_i = \lambda I_{2n}$ for some $\lambda \neq 0 \in \mathbb{F}_q$.

Summarizing the above, we have

$$\begin{aligned}
& \Pr[S \text{ non-scalar for } V_1, \ldots, V_6] \\
\leq\ & \Pr[S \text{ non-scalar for } V_i \wedge \mathbb{F}_q^{6n} = U_1 \oplus U_2 \oplus U_3 = W_1 \oplus W_2 \oplus W_3] + \frac{4}{q} \\
\leq\ & \Pr[S \text{ non-scalar for } V_i \mid \mathbb{F}_q^{6n} = U_1 \oplus U_2 \oplus U_3 = W_1 \oplus W_2 \oplus W_3] + \frac{4}{q} \\
\leq\ & \Pr[D \text{ non-scalar for } W \wedge \forall i,j \in [3], \mathrm{rk}(W_{ij}) = 2n] + \frac{20}{q} + \frac{4}{q} \\
\leq\ & \Pr[D \text{ non-scalar for } W \mid \forall i,j \in [3], \mathrm{rk}(W_{ij}) = 2n] + \frac{24}{q} \\
\leq\ & \frac{1}{q^{\Omega(n)}} + \frac{24}{q} \\
\leq\ & \frac{1}{q^{\Omega(1)}},
\end{aligned}$$

when $q = n^{\Omega(1)}$. This concludes the proof of Proposition 15. ◀

6 Application to p-Group Isomorphism, using constructive Baer and Lazard Correspondences

The applications to p-Group Isomorphism rely on the following well-known connections between alternating bilinear maps and Lie algebras on the one hand, and p-groups of "small" class on the other. We present these connections here, partly for audiences not from computational group theory, and partly because we will need to address some computational aspects of these procedures. We begin with some preliminaries.

6.1 Preliminaries

TI-completeness. As the proof of Theorem P in Section 6.3.1 uses a result on TI-completeness from [32], here we recall the definition of TI; see Definition 6 for the d-Tensor Isomorphism problem.

▶ **Definition 22** (dTI, TI). *For any field $\mathbb{F}$, $d\mathsf{TI}_{\mathbb{F}}$ denotes the class of problems that are polynomial-time Turing (Cook) reducible to d-Tensor Isomorphism over $\mathbb{F}$. Also let $\mathsf{TI}_{\mathbb{F}} = \bigcup_{d\geq 1} d\mathsf{TI}_{\mathbb{F}}$.*

The relationship between TI over different fields remains an intriguing open question [32], but here we will only need TI over $\mathbb{F}_p$. One of the the main results of [32] is that $\mathsf{TI} = d\mathsf{TI}$ for any fixed $d \geq 3$.

Algebras and their algorithmic representations. A Lie algebra $\mathcal{A}$ consists of a vector space V and a bilinear map $[,] : V \times V \to V$ that is alternating ($[v,v] = 0$ for all $v \in V$; this is equivalent to skew-symmetry $[u,v] = -[v,u]$ in characteristic not 2) and satisfies the Jacobi identity $[x,[y,z]] + [z,[x,y]] + [y,[z,x]] = 0$. The Jacobi identity is essentially the "derivative" of associativity.

After choosing an ordered basis $(b_1, \dots, b_n)$ where $b_i \in \mathbb{F}^n$ of $V \cong \mathbb{F}^n$, this bilinear map $[,]$ can be represented by an $n \times n \times n$ 3-way array $\mathtt{A}$, such that $[b_i, b_j] = \sum_{k\in[n]} \mathtt{A}(i,j,k)b_k$. This is the structure constant representation of $\mathcal{A}$. Algorithms for Lie algebras have been studied intensively in this model, e. g., [21, 38].

It is also natural to consider matrix spaces that are closed under commutator. More specifically, let $\mathcal{A} \leq \mathrm{M}(n, \mathbb{F})$ be a matrix space. If $\mathcal{A}$ is closed under commutator, that is, for any $A, B \in \mathcal{A}$, $[A, B] = AB - BA \in \mathcal{A}$, then $\mathcal{A}$ is a matrix Lie algebra with the product being the commutator. (Protip: one way to remember the Jacobi identity is to derive it as the natural identity among nested commutators of three matrices.) Algorithms for matrix Lie algebras have also been studied, e. g., [24, 36, 38].

6.2 Constructive Baer Correspondence and Theorems A and B

Let us review Baer's Correspondence [7], which connects alternating bilinear maps with p-groups of class 2 and exponent p. Let P be a p-group of class 2 and exponent p, $p > 2$. Suppose the commutator subgroup $[P, P] \cong \mathbb{Z}_p^m$ and $P/[P, P] \cong \mathbb{Z}_p^n$. Then the commutator map $[,] : P/[P, P] \times P/[P, P] \to [P, P]$ is an alternating bilinear map. Conversely, let $\phi : \mathbb{Z}_p^n \times \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ be an alternating bilinear map. Then a p-group of class 2 and exponent p, denoted as P_ϕ can be defined as follows. The group elements are from $\mathbb{Z}_p^n \times \mathbb{Z}_p^m$, and the group product $\cdot$ is defined as

$$(u, v) \cdot (u', v') = (u + u', v + v' + \frac{1}{2}\phi(u, u')).$$

We say that $(A, B) \in \mathrm{GL}(n, p) \times \mathrm{GL}(m, p)$ is a pseudo-autometry of ϕ, if $\phi = B \circ \phi \circ A$. Wilson [71] elucidated the structure of $\mathrm{Aut}(P_\phi)$ in terms of the pseudo-autometry group of ϕ, that we denote $\Psi\mathrm{Aut}(\phi)$. Here we recall the consequence of Wilson's result that we need for counting group isomorphisms.

▶ **Proposition 23** (Wilson [71, Prop. 3.8], see [15, Prop. 2.4] for notation closer to ours)**.** *For $\phi : \mathbb{Z}_p^n \times \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ an alternating bilinear map,*

$$|\mathrm{Aut}(P_\phi)| = |\Psi\mathrm{Aut}(\phi)|p^{nm},$$

where $\Psi\mathrm{Aut}(\phi)$ denotes the pseudo-autometry group of ϕ.

We then state a lemma which can be viewed as a constructive version of Baer's Correspondence, communicated to us by James B. Wilson.

▶ **Lemma 24** (Constructive version of Baer's Correspondence for matrix groups)**.** *Let p be an odd prime. Over the finite field $\mathbb{F} = \mathbb{F}_{p^e}$, ALTERNATING MATRIX SPACE ISOMETRY is equivalent to GROUP ISOMORPHISM for matrix groups over $\mathbb{F}$ that are p-groups of class 2 and exponent p. More precisely, there are functions computable in time* $\mathrm{poly}(n, m, \log |\mathbb{F}|)$*:*

- *G: $\Lambda(n, \mathbb{F})^m \to \mathrm{M}(n + m + 1, \mathbb{F})^{n+m}$ and*
- *Alt: $\mathrm{M}(n, \mathbb{F})^m \to \Lambda(m, \mathbb{F})^{O(m^2)}$*

such that: (1) for an alternating bilinear map $\mathbf{A}$, the group generated by $G(\mathbf{A})$ is the Baer group corresponding to $\mathbf{A}$, (2) G and Alt are mutually inverse, in the sense that the group generated by $G(Alt(M_1, \dots, M_m))$ is isomorphic to the group generated by $M_1, \dots, M_m$, and conversely $Alt(G(\mathbf{A}))$ is pseudo-isometric to $\mathbf{A}$.

Proof. First, let G be a p-group of class 2 and exponent p given by m generating matrices of size $n \times n$ over $\mathbb{F}$. Then from the generating matrices of G, we first compute a generating set of $[G, G]$, by just computing all the commutators of the given generators. We can then remove those redundant elements from this generating set in time $\mathrm{poly}(\log |[G,G]|, \log |\mathbb{F}|)$, using Luks' result on computing with solvable matrix groups [51]. We then compute a set of representatives of a non-redundant generating set of $G/[G,G]$, again using Luks's aforementioned result. From these data we can compute an alternating bilinear map representing the commutator map of G in time $\mathrm{poly}(n, m, \log |\mathbb{F}|)$.

Conversely, let an alternating bilinear map be given by $\mathbf{A} = (A_1, \ldots, A_m) \in \Lambda(n, \mathbb{F})^m$. From $\mathbf{A}$, for $i \in [n]$, construct $B_i = [A_1 e_i, \ldots, A_m e_i] \in \mathrm{M}(n \times m, \mathbb{F})$, where e_i is the ith standard basis vector of $\mathbb{F}^n$. That is, the jth column of B_i is the ith column of A_j. Then for $i \in [n]$, construct

$$\tilde{B_i} = \begin{bmatrix} 1 & e_i^t & 0 \\ 0 & I_n & B_i \\ 0 & 0 & I_m \end{bmatrix} \in \mathrm{GL}(1+n+m, \mathbb{F}),$$

where $e_i \in \mathbb{F}^n$, and for $j \in [m]$, construct

$$\tilde{C_j} = \begin{bmatrix} 1 & 0 & e_j^t \\ 0 & I_n & 0 \\ 0 & 0 & I_m \end{bmatrix} \in \mathrm{GL}(1+n+m, \mathbb{F}),$$

where $e_j \in \mathbb{F}^m$. Let $G(\mathbf{A})$ be the matrix group generated by $\tilde{B_i}$ and $\tilde{C_j}$. Then it can be verified easily that, $G(\mathbf{A})$ is isomorphic to the Baer group corresponding to the alternating bilinear map defined by $\mathbf{A}$. In particular, $[G,G] \cong \mathbb{F}^m \cong \mathbb{Z}_p^{em}$ (isomorphism of abelian groups), and $G/[G,G] \cong \mathbb{F}^n \cong \mathbb{Z}_p^{en}$. This construction can be done in time $\mathrm{poly}(n, m, \log |\mathbb{F}|)$. ◀

Given the above lemma, we can present search- and counting-to-decision reductions for testing isomorphism of a class of p-groups, proving Theorems A and B.

Proof of Theorem A. The search-to-decision reduction follows from Theorem A′, using the $q^{O(n+m)}$-time algorithm, with the constructive version of Baer's Correspondence in the model of matrix groups over finite fields (Lemma 24).

In more detail, given Lemma 24 we can follow the procedure in the proof of Theorem A′. For the given p-groups, we compute their commutator maps. Then whenever we need to feed the decision oracle, we transform from the alternating bilinear map to a generating set of a p-group of class 2 and exponent p with this bilinear map as the commutator map. After getting the desired pseudo-isometry for the alternating bilinear maps, we can easily recover an isomorphism between the originally given p-groups. ◀

Proof of Theorem B. For the counting-to-decision reduction, we basically follow the above routine, but with a twist, because of the minor distinction between alternating matrix space isometry, and alternating bilinear map pseudo-isometry. Let us briefly explain this issue. Suppose from an alternating bilinear map $\phi : \mathbb{Z}_p^n \times \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ we constructed the p-group P_ϕ of class 2 and exponent p; by Proposition 23 $|\mathrm{Aut}(P_\phi)| = p^{nm} |\Psi\mathrm{Aut}(\phi)|$, so by multiplying the result by p^{nm}, it is necessary and sufficient to count the psuedo-autometries of ϕ.

Towards that end, let $(C_1, \ldots, C_m) \in \Lambda(n, p)$ be a matrix representation of ϕ. If C_i's are linearly independent, then for a pseudo-autometry $(A, B) \in \mathrm{GL}(n, p) \times \mathrm{GL}(m, p)$, given A there exists a unique B that makes (A, B) a pseudo-autometry. If C_i's are not linearly independent, say the linear span of C_i's is of dimension m', then the number of B such that (A, B) is a pseudo-autometry (assuming there are any) is $|\mathrm{M}((m-m') \times m', p)||\mathrm{GL}(m-m', p)| = p^{m'(m-m')}|\mathrm{GL}(m-m', p)|$. To see this, suppose that we have taken linear combinations of the C_i so that $C_1, \ldots, C_m'$ are linearly independent and $C_{m'+1}, C_{m'+2}, \ldots, C_m$ are zero. Then without changing the C_i, we may take any invertible linear combination among $C_{m'+1}, \dots, C_m$ (a copy of $\mathrm{GL}(m-m',p)$), and we may add any linear combination of the last $m-m'$ matrices to the first m' matrices (a copy of $\mathrm{M}((m-m') \times m', p)$). The counting to decision reduction for Alternating Matrix Space Isometry computes the number of $A \in \mathrm{GL}(n,p)$ so that there exists some $B \in \mathrm{GL}(m,p)$ such that (A,B) is a pseudo-autometry. So it needs to be multiplied by a factor of $p^{m'(m-m')}|\mathrm{GL}(m-m',p)|$. ◀

6.3 Constructive Lazard's Correspondence and Theorem P

The Lazard Correspondence [46] is a correspondence between certain classes of groups and Lie algebras, which extends the usual correspondence between Lie groups and Lie algebras (say, over $\mathbb{R}$) to some groups and Lie algebras in positive characteristic. Here we state just enough to give a sense of it; for further details and exposition we refer to Khukhro's book [43] and Naik's thesis [60]. While Naik's thesis is quite long, it also includes a reader's guide, and collects many results scattered across the literature or well-known to the experts in one place, building the theory from the ground up and with many examples.

Recall that a *Lie ring* is an abelian group L equipped with a bilinear map $[,]$, called the Lie bracket, which is (1) alternating ($[x,x]=0$ for all $x \in L$) and (2) satisfies the Jacobi identity $[x,[y,z]]+[y,[z,x]]+[z,[x,y]]=0$ for all $x,y,z \in L$ (in some sense the "derivative" of the associativity equation). Let $L^1 = L$, and $L^{i+1} = [L, L^i]$, which is the subgroup (of the underlying additive group) generated by all elements of the form $[x,y]$ for $x \in L, y \in L^i$. Then L is *nilpotent* if $L^{c+1} = 0$ for some finite c; the smallest such c is the *nilpotency class*. (Lie algebras are just Lie rings over a field.)

The correspondence between Lie algebras and Lie groups over $\mathbb{R}$ uses the Baker–Campbell–Hausdorff (BCH) formula to convert between a Lie algebra and a Lie group, so we start there. The BCH formula is the solution to the problem that for non-commuting matrices X, Y, $e^X e^Y \neq e^{X+Y}$ in general (where the matrix exponential here is defined using the power series for e^x). Rather, using commutators $[A,B] = AB - BA$, we have

$$\exp(X)\exp(Y) = \exp\left(X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}\left([X,[X,Y]] - [Y,[X,Y]]\right) + \cdots\right),$$

where the remaining terms are iterated commutators that all involve at least 4 Xs and Ys, and successive terms involve more and more. Applying the exponential function to a Lie algebra in characteristic zero yields a Lie group. The BCH formula can be inverted, giving the correspondence in the other direction.

In a nilpotent Lie algebra, the BCH formula has only finitely many nonzero terms, so issues of convergence disappear and we may consider applying the correspondence over finite fields or rings; the only remaining obstacle is that the denominators appearing in the formula must be units in the ring. It turns out that the correspondence continues to work in characteristic p so long as one does not need to use the p-th term of the BCH formula (which includes division by p), and the latter is avoided whenever a nilpotent group has class strictly less than p, or even when all subgroups generated by at most 3 elements have class strictly less than p. While the correspondence does apply more generally, here we only state the version for finite groups. For any fixed nilpotency class c, computing the Lazard Correspondence is efficient in theory; for how to compute it in practice when the groups are given by polycyclic presentations, see [20].

Let $\mathbf{Grp}_{p,n,c}$ denote the set of finite groups of order p^n and class c, and let $\mathbf{Lie}_{p,n,c}$ denote the set of Lie rings of order p^n and class c. We note that for nilpotency class 2, the Baer Correspondence is the same as the Lazard Correspondence.

▶ **Theorem 25** (Lazard Correspondence for finite groups [46], see, e. g., [43, Ch. 9 & 10] or [60, Ch. 6]). *For any prime p and any $1 \leq c < p$, there are functions* $\mathbf{log}\colon \mathbf{Grp}_{p,n,c} \leftrightarrow \mathbf{Lie}_{p,n,c} : \mathbf{exp}$ *such that (1)* $\mathbf{log}$ *and* $\mathbf{exp}$ *are inverses of one another, (2) two groups $G, H \in \mathbf{Grp}_{p,n,c}$ are isomorphic if and only if* $\mathbf{log}(G)$ *and* $\mathbf{log}(H)$ *are isomorphic, and (3) if G has exponent p, then the underlying abelian group of* $\mathbf{log}(G)$ *has exponent p. More strongly,* $\mathbf{log}$ *is an isomorphism of categories* $\mathbf{Grp}_{p,n,c} \cong \mathbf{Lie}_{p,n,c}$.

Part (3) can be found as a special case of [60, Lemma 6.1.2].

For p-groups given by $d \times d$ matrices over the finite field $\mathbb{F}_{p^e}$, we will need one additional fact about the correspondence, namely that it also results in a Lie algebra of $d \times d$ matrices. (Being able to bound the dimension of the Lie algebra and work with it in a simple linear-algebraic way seems crucial for our reduction to work efficiently.) In fact, the BCH Correspondence is *easier* to see for matrix groups using the matrix exponential and matrix logarithm; most of the work for BCH and Lazard is to get the correspondence to work even *without* the matrices. In some sense, this is thus the "original" setting of this correspondence. Though it is surely not new, we could not find a convenient reference for this fact about matrix groups over finite fields, so we state it formally here.

▶ **Proposition 26** (cf. [43, Exercise 10.6]). *Let $G \leq \mathrm{GL}(d, \mathbb{F}_{p^e})$ be a finite p-subgroup of exponent p, consisting of $d \times d$ matrices over a finite field of characteristic p. Then* $\mathbf{log}(G)$ *(from the Lazard Correspondence) can be realized as a finite Lie subalgebra of $de \times de$ matrices over $\mathbb{F}_p$. Given a generating set for G of m matrices, a generating set for* $\mathbf{log}(G)$ *can be constructed in* $\mathrm{poly}(d, n, e \log p)$ *time.*

Khukhro [43] gives the characteristic zero analogue of this result (minus the straightforward complexity analysis) for the full group of upper unitriangular matrices as Exercise 10.6. One way to see Proposition 26 is to use the characteristic zero result, apply the fact that these isomorphisms are in fact equivalence of categories (and thus hold for subgroups/subalgebras as well), and note that the same formulae in characteristic zero apply in characteristic p so long as one never needs to divide by p. We now sketch the argument.

Proof sketch. First we use the standard embedding of $\mathrm{GL}(d, \mathbb{F}_{p^e})$ into $\mathrm{GL}(de, \mathbb{F}_p)$ (replace each element by an $e \times e$ block which is the left regular representation of $\mathbb{F}_{p^e}$ acting on itself as an e-dimensional $\mathbb{F}_p$-vector space), to realize G as a subgroup of $\mathrm{GL}(de, \mathbb{F}_p)$. G is conjugate in $\mathrm{GL}(de, \mathbb{F}_p)$ to a group of upper unitriangular matrices (upper triangular with all 1s on the diagonal); this is a standard fact that can be seen in several ways, for example, by noting that the group U of all upper unitriangular matrices in $\mathrm{GL}(de, \mathbb{F}_p)$ is a Sylow p-subgroup, and applying Sylow's Theorem. (Note that we do not need to do this conjugation algorithmically, though it is possible to do so [27, 36, 64]; this is only for the proof.) Thus we may write every $g \in G$ as $1 + n$, where the sum here is the ordinary sum of matrices, 1 denotes the identity matrix, and n is strictly upper triangular. To see that we can truncate the Taylor series for logarithm before the p-th term (thus avoiding needing to divide by p), note that $(1+n)^p = 1$ since G is exponent p. We have $(1+n)^p = 1^p + \binom{p}{1}n + \binom{p}{2}n^2 + \cdots + \binom{p}{p-1}n^{p-1} + n^p$. Since these are matrices over a field of characteristic p, and $p | \binom{p}{i}$ for all $1 \leq i \leq p-1$, all the intermediate terms vanish and we have that $(1+n)^p = 1^p + n^p$. Thus $1 = (1+n)^p = 1 + n^p$, so we get that $n^p = 0$. Thus, in the the Taylor series for the logarithm

$$\log(1+n) = n - \frac{n^2}{2} + \frac{n^3}{3} - \cdots$$

the last nonzero term is $n^{p-1}/(p-1)$, so we may use this Taylor series even over $\mathbb{F}_{p^e}$.

The main things to check are that the set $\log(G) := \{\log(1+n) : 1+n \in G\}$ is closed under scalar multiplication, matrix addition, and matrix commutator $[X, Y] = XY - YX$. Suppose g_1, g_2 are matrices in G, and write them as $g_i = 1 + n_i$ $(i = 1, 2)$, as above. We recall that, because $n_i^p = 0$ from above, the power series for both log and exp work to compute the matrix logarithm and exponential over $\mathbb{F}_{p^e}$, respectively, and that the usual rules of logarithms are satisfied for a single matrix A: whenever $A \in M_{de}(\mathbb{F}_p)$ satisfies $A^p = 0$, we have $\log \exp A = A$, $\exp \log(1 + A) = 1 + A$, $\exp(nA) = (\exp A)^n$ for $n \in \mathbb{Z}$, and $\log((1 + A)^n) = n \log(1 + A)$.

- Scalar multiplication: For $\alpha \in \mathbb{F}_p$, we show that $\alpha \log(1 + n_1)$ is in $\log(G)$. This is easy to show, as it follows directly from the rules of logarithms just mentioned: $\alpha \log(1 + n_1) = \log((1 + n_1)^\alpha)$ where on the right-hand side we treat α as an integer in the range $[0, p - 1]$. (This is the only point where we are using that we are working over $\mathbb{F}_p$ now rather than $\mathbb{F}_{p^e}$.)
- Addition: Let $x_i = \log(1 + n_i)$ for $i = 1, 2$. We want to show that $x_1 + x_2$ is in $\log(G)$, or equivalently that $\exp(x_1 + x_2) \in G$. This follows from the first inverse BCH formula h_1, which satisfies $\exp(\hat{x}_1 + \hat{x}_2) = h_1(\exp(\hat{x}_1), \exp(\hat{x}_2))$ for $\hat{x}_i$ in the free nilpotent-of-class-c $\mathbb{F}_p$-Lie algebra, and then we may apply the homomorphism from the latter algebra to the subalgebra of $\mathrm{M}_n(\mathbb{F}_p)$ generated by the n_i to see that the same formula works. (We note, because a reviewer asked, that here we do not need this entire subalgebra to be in $\{g - 1 : g \in G\}$; the use of that subalgebra is just convenient for talking about algebra homomorphisms in the proof. Rather, it suffices that the preceding equation holds for these particular elements n_i, which are by definition of the form $g_i - 1$ for some matrices $g_i \in G$.)
- Commutator: $[\log(1 + n_1), \log(1 + n_2)]$. A similar argument as in the previous case works, using the second inverse BCH formula h_2, which satisfies $\exp([\hat{x}_1, \hat{x}_2]) = h_2(\exp(\hat{x}_1), \exp(\hat{x}_2))$.

Equivalently, we may note that the derivation of the inverse BCH formulas in [43] uses a free nilpotent associative algebra as an ambient setting in which both the group (or rather, n such that $1 + n$ is in the group) and the corresponding Lie algebra live; in our case, we may replace the ambient free nilpotent associative algebra with the algebra of $de \times de$ strictly upper-triangular matrices over $\mathbb{F}_p$, and all the derivations remain the same, *mutatis mutandis*. See, for example, [43, p. 105, "Another remark..."]. ◀

6.3.1 Class reduction in p-group isomorphism testing

Proposition 26 now allows us to prove Theorem P.

Proof of Theorem P. By the Lazard Correspondence (reproduced as Theorem 25) two p-groups of exponent p and class $c < p$ are isomorphic if and only if their corresponding $\mathbb{F}_p$-Lie algebras are. By Proposition 26, we can construct a generating set for the corresponding $\mathbb{F}_p$-Lie algebra by applying the power series for logarithm to the generating matrices of G. This Lie algebra is thus a subalgebra of $ne \times ne$ matrices over $\mathbb{F}_p$, so we can generate a basis for the entire Lie algebra (using the linear-algebra version of breadth-first search; its dimension is $\leq (ne)^2$) and compute its structure constants in time polynomial in n, m, and $e \log p$. Then use [28] to reduce isomorphism of Lie algebras to 3-Tensor Isomorphism, and then use the fact that isomorphism of p-groups of exponent p and class 2 given by a matrix generating set over $\mathbb{F}_p$ is TI-complete [32] to reduce to the latter problem. ◀

7 Conclusion

In this paper, we gave first-of-their-kind results around search-to-decision, counting-to-decision, and reductions to hard instances in the context of Group Isomorphism. We focused on p-groups of class 2 (or more generally small class) and exponent p, as these are widely believed to be the hardest cases of GpI. They also have the closest connection with tensors.

We view this paper as the second in a planned series, focusing on isomorphism problems for tensors, groups, polynomials, and related structures. Although Graph Isomorphism (GI) is perhaps the most well-studied isomorphism problem in computational complexity – even going back to Cook's and Levin's initial investigations into NP (see [1, Sec. 1]) – it has long been considered to be solvable in practice [55,56], and Babai's recent quasi-polynomial-time breakthrough is one of the theoretical gems of the last several decades [3]. However, several isomorphism problems for tensors, groups, and polynomials seem to be much harder to solve, both in practice – they've been suggested as difficult enough to support cryptography [39,61] – and in theory: the best known worst-case upper bounds are barely improved from brute force (e. g., [49,66]). As these problems arise in a variety of areas, from multivariate cryptography and machine learning, to quantum information and computational algebra, getting a better understanding of their complexity is an important goal with many potential applications.

In the first paper in this series [32], we showed that numerous such isomorphism problems from many research areas are equivalent under polynomial-time reductions, creating bridges between different disciplines. The Tensor Isomorphism (TI) problem turns out to occupy a central position among these problems, leading us to define the complexity class TI, consisting of those problems polynomial-time reducible to the Tensor Isomorphism problem. The gadgets and TI-completeness result from that first paper in some cases opened the door, and in other cases are used as subroutines, in the main results of the current paper.

Finally, we list here some additional questions that we find interesting and approachable. One question is whether our tensor-based methods here can be extended or combined with other methods to get analogous results in wider classes of groups; for isomorphism algorithms, something along these lines was proposed by Brooksbank, Grochow, Li, Wilson, & Qiao [12], but there are many interesting open questions in this direction.

Getting the results of this paper to work in the Cayley table model would also be interesting from the complexity-theoretic perspective; the necessary ingredients are discussed in Remark 2.

Lastly, we mention that extending the results of the present paper, [28], and [32] to rings beyond fields would be very interesting. In particular, working with tensors over $\mathbb{Z}/p^k\mathbb{Z}$ is an important step towards extending the results of this paper to p-groups of class 2 without restricting them to exponent p. (This is particularly important when $p = 2$, as groups of exponent 2 are abelian, so the hardest instances of 2-groups, rather than "p-groups of class 2 and exponent p" with $p = 2$, are often taken to be 2-groups of class 2 and exponent *four*.)

It seems conceivable that many of our arguments could extend to tensors over local rings – those with a unique maximal ideal – as many of our arguments are rank-based, and rank still has nice properties over local rings (e.g. Nakayama's Lemma). In particular, if R is a ring and $\mathfrak{m}$ a maximal ideal, then $R/\mathfrak{m}$ is a field; in a local ring, there is a unique maximal ideal, so the field $R/\mathfrak{m}$ is canonically associated to R, and one can talk cleanly about rank and dimension of R-modules considered over the field $R/\mathfrak{m}$. Besides $\mathbb{Z}/p^k\mathbb{Z}$, another local ring of interest is the ring $\mathbb{F}[[t]]$ of power series in one variable over a field $\mathbb{F}$; a tensor over $\mathbb{F}[[t]]$ is essentially a 1-parameter family of tensors over $\mathbb{F}$, so studying tensor problems over $\mathbb{F}[[t]]$ could have applications to border rank and geometric complexity theory.

References

1 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Inf. Comput.*, 256:2–8, 2017. doi:10.1016/j.ic.2017.04.004.

2 Vikraman Arvind and Jacobo Torán. Isomorphism testing: Perspective and open problems. *Bulletin of the EATCS*, 86:66–84, 2005.

3 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 684–697, 2016. arXiv:1512.03547 [cs.DS] version 2. doi:10.1145/2897518.2897542.

4 László Babai, Paolo Codenotti, Joshua A. Grochow, and Youming Qiao. Code equivalence and group isomorphism. In *Proceedings of the Twenty-Second Annual ACM–SIAM Symposium on Discrete Algorithms (SODA11)*, pages 1395–1408, Philadelphia, PA, 2011. SIAM. doi:10.1137/1.9781611973082.107.

5 László Babai, Paolo Codenotti, and Youming Qiao. Polynomial-time isomorphism test for groups with no abelian normal subgroups - (extended abstract). In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Proceedings, Part I*, pages 51–62, 2012. doi:10.1007/978-3-642-31594-7_5.

6 László Babai and Youming Qiao. Polynomial-time isomorphism test for groups with Abelian Sylow towers. In *29th STACS*, pages 453–464. Springer LNCS 6651, 2012. doi:10.4230/LIPIcs.STACS.2012.453.

7 Reinhold Baer. Groups with abelian central quotient group. *Trans. AMS*, 44(3):357–386, 1938. doi:10.1090/S0002-9947-1938-1501972-1.

8 Mihir Bellare and Shafi Goldwasser. The complexity of decision versus search. *SIAM J. Comput.*, 23(1):97–119, 1994. doi:10.1137/S0097539792228289.

9 Hans Ulrich Besche and Bettina Eick. Construction of finite groups. *J. Symb. Comput.*, 27(4):387–404, 1999. doi:10.1006/jsco.1998.0258.

10 Hans Ulrich Besche, Bettina Eick, and E.A. O'Brien. A millennium project: Constructing small groups. *Intern. J. Alg. and Comput*, 12:623–644, 2002. doi:10.1142/S0218196702001115.

11 Anton Betten, Michael Braun, Harald Fripertinger, Adalbert Kerber, Axel Kohnert, and Alfred Wassermann. *Error-correcting linear codes: Classification by isometry and applications*, volume 18. Springer Science and Business Media, 2006.

12 Peter A. Brooksbank, Joshua A. Grochow, Yinan Li, Youming Qiao, and James B. Wilson. Incorporating Weisfeiler–Leman into algorithms for group isomorphism. arXiv:1905.02518 [cs.CC], 2019.

13 Peter A. Brooksbank, Yinan Li, Youming Qiao, and James B. Wilson. Improved algorithms for alternating matrix space isometry: From theory to practice. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ESA.2020.26.

14 Peter A. Brooksbank and Eugene M. Luks. Testing isomorphism of modules. *J. Algebra*, 320(11):4020–4029, 2008. doi:10.1016/j.jalgebra.2008.07.014.

15 Peter A. Brooksbank, Joshua Maglione, and James B. Wilson. A fast isomorphism test for groups whose Lie algebra has genus 2. *J. Algebra*, 473:545–590, 2017. doi:10.1016/j.jalgebra.2016.12.007.

16 Peter A. Brooksbank and E. A. O'Brien. Constructing the group preserving a system of forms. *Internat. J. Algebra Comput.*, 18(2):227–241, 2008. doi:10.1142/S021819670800441X.

17 John J. Cannon and Derek F. Holt. Automorphism group computation and isomorphism testing in finite groups. *J. Symbolic Comput.*, 35(3):241–267, 2003. doi:10.1016/S0747-7171(02)00133-5.

18 Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *J. ACM*, 56(3):Art. 14, 57, 2009. doi:10.1145/1516512.1516516.

19 Alexander Chistov, Gábor Ivanyos, and Marek Karpinski. Polynomial time algorithms for modules over finite dimensional algebras. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, ISSAC '97, pages 68–74. ACM, 1997. doi:10.1145/258726.258751.

20 Serena Cicalò, Willem A. de Graaf, and Michael Vaughan-Lee. An effective version of the Lazard correspondence. *J. Algebra*, 352(1):430–450, 2012. doi:10.1016/j.jalgebra.2011.11.031.

21 W.A. de Graaf. *Lie Algebras: Theory and Algorithms*, volume 56 of *North-Holland Mathematical Library*. Elsevier Science, 2000.

22 Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Trans. Algorithms*, 10(4):Art. 21, 32, 2014. doi:10.1145/2635812.

23 Sean Eberhard and Stefan-C. Virchow. Random generation of the special linear group. *Transactions of the American Mathematical Society*, page 1, 2020. doi:10.1090/tran/8009.

24 Wayne Eberly and Mark Giesbrecht. Efficient decomposition of associative algebras over finite fields. *Journal of Symbolic Computation*, 29(3):441–458, 2000. doi:10.1006/jsco.1999.0308.

25 Bettina Eick, C. R. Leedham-Green, and E. A. O'Brien. Constructing automorphism groups of p-groups. *Comm. Algebra*, 30(5):2271–2295, 2002. doi:10.1081/AGB-120003468.

26 V. Felsch and J. Neubüser. On a programme for the determination of the automorphism group of a finite group. In Pergamon J. Leech, editor, *Computational Problems in Abstract Algebra (Proceedings of a Conference on Computational Problems in Algebra, Oxford, 1967)*, pages 59–60, Oxford, 1970.

27 Katalin Friedl and Lajos Rónyai. Polynomial time solutions of some problems in computational algebra. In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 153–162. ACM, 1985. doi:10.1145/22145.22162.

28 Vyacheslav Futorny, Joshua A. Grochow, and Vladimir V. Sergeichuk. Wildness for tensors. *Lin. Alg. Appl.*, 566:212–244, 2019. doi:10.1016/j.laa.2018.12.022.

29 Joshua A. Grochow. Answer to "what is the hardest instance for the group isomorphism problem?". Theoretical Computer Science Stack Exchange. URL: https://cstheory.stackexchange.com/a/42551/129.

30 Joshua A. Grochow and Youming Qiao. Polynomial-time isomorphism test of groups that are tame extensions - (extended abstract). In *Algorithms and Computation - 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings*, pages 578–589, 2015. doi:10.1007/978-3-662-48971-0_49.

31 Joshua A. Grochow and Youming Qiao. Algorithms for group isomorphism via group extensions and cohomology. *SIAM J. Comput.*, 46(4):1153–1216, 2017. Preliminary version in IEEE Conference on Computational Complexity (CCC) 2014 (DOI:10.1109/CCC.2014.19). Also available as arXiv:1309.1776 [cs.DS] and ECCC Technical Report TR13-123. doi:10.1137/15M1009767.

32 Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: Tensor Isomorphism-completeness. In *ITCS*, page to appear, 2021. arXiv:1907.00309.

33 Martin Grohe and Pascal Schweitzer. The graph isomorphism problem. *Commun. ACM*, 63(11):128–134, 2020. doi:10.1145/3372123.

34 Xiaoyu He and Youming Qiao. On the Baer–Lovász–Tutte construction of groups from graphs: isomorphism types and homomorphism notions. arXiv:2003.07200 [math.CO], 2020.

35 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. System Sci.*, 62(2):367–375, 2001. Special issue on the Fourteenth Annual IEEE Conference on Computational Complexity (Atlanta, GA, 1999). doi:10.1006/jcss.2000.1727.

36 Gábor Ivanyos. Fast randomized algorithms for the structure of matrix algebras over finite fields. In *Proceedings of the 2000 international symposium on Symbolic and algebraic computation*, pages 175–183. ACM, 2000. doi:10.1145/345542.345620.

37 Gábor Ivanyos, Marek Karpinski, and Nitin Saxena. Deterministic polynomial time algorithms for matrix completion problems. *SIAM J. Comput.*, 39(8):3736–3751, 2010. doi:10.1137/090781231.

38 Gábor Ivanyos and Lajos Rónyai. Computations in associative and Lie algebras. In *Some tapas of computer algebra*, pages 91–120. Springer, 1999. doi:10.1007/978-3-662-03891-8_5.

39 Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I*, volume 11891 of *Lecture Notes in Computer Science*, pages 251–281. Springer, 2019. Preprint arXiv:1906.04330 [cs.CR]. doi:10.1007/978-3-030-36030-6_11.

40 William M. Kantor. Some topics in asymptotic group theory. *Groups, Combinatorics and Geometry (Durham*, pages 403–421, 1990.

41 William M Kantor and Alexander Lubotzky. The probability of generating a finite classical group. *Geometriae Dedicata*, 36(1):67–87, 1990.

42 Neeraj Kayal and Timur Nezhmetdinov. Factoring groups efficiently. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 585–596. Springer, 2009. Preprint ECCC Tech. Report TR08-074. doi:10.1007/978-3-642-02927-1_49.

43 E. I. Khukhro. *p-automorphisms of finite p-groups*, volume 246 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1998. doi:10.1017/CBO9780511526008.

44 Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1993. doi:10.1007/978-1-4612-0333-9.

45 Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. doi:10.1137/07070111X.

46 Michel Lazard. Sur les groupes nilpotents et les anneaux de Lie. *Ann. Sci. Ecole Norm. Sup. (3)*, 71:101–190, 1954. doi:10.24033/asens.1021.

47 François Le Gall. Efficient isomorphism testing for a class of group extensions. In *Proc. 26th STACS*, pages 625–636, 2009. doi:10.4230/LIPIcs.STACS.2009.1830.

48 Mark L. Lewis and James B. Wilson. Isomorphism in expanding families of indistinguishable groups. *Groups Complex. Cryptol.*, 4(1):73–110, 2012. doi:10.1515/gcc-2012-0008.

49 Yinan Li and Youming Qiao. Linear algebraic analogues of the graph isomorphism problem and the Erdős–Rényi model. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 463–474. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.49.

50 Richard J. Lipton, Lawrence Snyder, and Yechezkel Zalcstein. The complexity of word and isomorphism problems for finite groups. Yale University Department of Computer Science Research Report # 91, 1977. URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/a053246.pdf.

51 Eugene M. Luks. Computing in solvable matrix groups. In *FOCS 1992, 33rd Annual Symposium on Foundations of Computer Science*, pages 111–120. IEEE Computer Society, 1992. doi:10.1109/SFCS.1992.267813.

52 Eugene M. Luks. Permutation groups and polynomial-time computation. In *Groups and computation (New Brunswick, NJ, 1991)*, volume 11 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 139–175. Amer. Math. Soc., Providence, RI, 1993.

53 Eugene M. Luks. Hypergraph isomorphism and structural equivalence of boolean functions. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 652–658, 1999. doi:10.1145/301250.301427.

54 Rudolf Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–136, 1979.

55 Brendan D. McKay. Practical graph isomorphism. *Congr. Numer.*, pages 45–87, 1980.

56 Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60(0):94–112, 2014. doi:10.1016/j.jsc.2013.09.003.

57 Alan H. Mekler. Stability of nilpotent groups of class 2 and prime exponent. *The Journal of Symbolic Logic*, 46(4):781–788, 1981.

58 Gary L. Miller. On the $n^{\log n}$ isomorphism technique (a preliminary report). In *STOC*, pages 51–58. ACM, 1978. doi:10.1145/800133.804331.

59 Takunari Miyazaki. Luks's reduction of graph isomorphism to code equivalence. Comment to E. W. Clark, https://groups.google.com/forum/#!msg/sci.math.research/puZxGj9HXKI/CeyH2yyyNFUJ, 1996.

60 Vipul Naik. *Lazard correspondence up to isoclinism*. PhD thesis, The University of Chicago, 2013. URL: https://vipulnaik.com/thesis/.

61 Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, pages 33–48, 1996. doi:10.1007/3-540-68339-9_4.

62 Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Trans. Inf. Theory*, 43(5):1602–1604, 1997. doi:10.1109/18.623157.

63 Youming Qiao, Jayalal M. N. Sarma, and Bangsheng Tang. On isomorphism testing of groups with normal Hall subgroups. In *Proc. 28th STACS*, pages 567–578, 2011. doi:10.4230/LIPIcs.STACS.2011.567.

64 Lajos Rónyai. Computing the structure of finite algebras. *J. Symb. Comput.*, 9(3):355–373, 1990. doi:10.1016/S0747-7171(08)80017-X.

65 David J. Rosenbaum. Bidirectional collision detection and faster deterministic isomorphism testing. arXiv preprint arXiv:1304.3935 [cs.DS], 2013.

66 David J. Rosenbaum. Breaking the $n^{\log n}$ barrier for solvable-group isomorphism. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1054–1073. SIAM, 2013. Preprint arXiv:1205.0642 [cs.DS]. doi:10.1137/1.9781611973105.76.

67 Nicolas Sendrier and Dimitris E. Simos. The hardness of code equivalence over $\mathbb{F}_q$ and its application to code-based cryptography. In *International Workshop on Post-Quantum Cryptography*, pages 203–216. Springer, 2013.

68 Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. doi:10.1137/0220053.

69 Leslie G. Valiant. Relative complexity of checking and evaluating. *Information Processing Lett.*, 5(1):20–23, 1976/77. doi:10.1016/0020-0190(76)90097-1.

70 James Wilson. 2014 conference on *Groups, Computation, and Geometry* at Colorado State University, co-organized by P. Brooksbank, A. Hulpke, T. Penttila, J. Wilson, and W. Kantor. Personal communication, 2014.

71 James B. Wilson. Decomposing p-groups via Jordan algebras. *J. Algebra*, 322(8):2642–2679, 2009. doi:10.1016/j.jalgebra.2009.07.029.

72 James B. Wilson. Finding direct product decompositions in polynomial time. arXiv:1005.0548 [math.GR], 2010.

73 James B. Wilson. Existence, algorithms, and asymptotics of direct product decompositions, I. *Groups Complex. Cryptol.*, 4(1):33–72, 2012. doi:10.1515/gcc-2012-0007.

74 V. N. Zemlyachenko, N. M. Korneenko, and R. I. Tyshkevich. Graph isomorphism problem. *J. Soviet Math.*, 29(4):1426–1481, May 1985. doi:10.1007/BF02104746.

Branching Programs with Bounded Repetitions and Flow Formulas

Anastasia Sofronova
St. Petersburg Department of Steklov Mathematical Institute of
Russian Academy of Sciences, Russia

Dmitry Sokolov
St. Petersburg Department of Steklov Mathematical Institute of
Russian Academy of Sciences, Russia
St. Petersburg State University, Russia

Abstract

Restricted branching programs capture various complexity measures like space in Turing machines or length of proofs in proof systems. In this paper, we focus on the application in the proof complexity that was discovered by Lovasz et al. [14] who showed the equivalence between regular Resolution and read-once branching programs for "unsatisfied clause search problem" (Search_φ). This connection is widely used, in particular, in the recent breakthrough result about the Clique problem in regular Resolution by Atserias et al. [5].

We study the branching programs with bounded repetitions, so-called $(1, +k)$-BPs (Sieling [21]) in application to the Search_φ problem. On the one hand, it is a natural generalization of read-once branching programs. On the other hand, this model gives a powerful proof system that can efficiently certify the unsatisfiability of a wide class of formulas that is hard for Resolution (Knop [13]).

We deal with Search_φ that is "relatively easy" compared to all known hard examples for the $(1, +k)$-BPs. We introduce the first technique for proving exponential lower bounds for the $(1, +k)$-BPs on Search_φ. To do it we combine a well-known technique for proving lower bounds on the size of branching programs [12, 21, 22] with the modification of the "closure" technique [1, 3]. In contrast with most Resolution lower bounds, our technique uses not only "local" properties of the formula, but also a "global" structure. Our hard examples are based on the Flow formulas introduced in [3].

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases proof complexity, branching programs, bounded repetitions, lower bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.17

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2021/028/

Funding *Anastasia Sofronova*: The research presented in Sections 3 and 4 is supported by Russian Science Foundation (project 18-71-10042).

Acknowledgements The authors would like to thank Dmitry Itsykson, Artur Riazanov for fruitful discussions and comments, Edward Hirsch for comments on the draft, Alexander Knop for a statement of the problem. The authors would also like to thank anonymous reviewers for their valuable comments.

1 Introduction

Branching program is a computational model that generalizes decision tree in the most natural way: the underlying graph of computation can be an arbitrary directed acyclic graph. This is one of the most fundamental models in theoretical computer science: it captures the space complexity of many versions of restricted and unrestricted Turing machines, various proof systems may be described in terms of this model, etc.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 17; pp. 17:1–17:25
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A Shannon's style counting argument says that there is a boolean function such that any branching program that computes this function requires an exponential size. However for an explicit function, we are still far from a superpolynomial lower bound, and the best known result is $\frac{n^2}{\log^2 n}$ due to Nechiporuk [16].

For some applications, it is enough to deal with the restricted models of branching programs and many such models were considered. One of the most popular restrictions is the read-once model of branching programs [15] where any input bit may be queried at most once during each computation. This model corresponds to the *eraser Turing machines*. Exponential lower bounds for this model were proven in [24, 26]. For capturing more general machines some natural generalization of read-once branching programs were studied. And one of the most important models among these generalizations is the model with bounded repetitions aka $(1, +k)$-BP that was described in [21]. In this model, we allow our branching programs to requery variables, but on each computation only k input bits may be queried more than one time. There are two natural points of view on this model:

- *syntactic:* if we apply the restriction on *every* path;
- *semantic:* if we apply the restriction on *consistent* paths

(for formal definition see section 2.1). The semantic version is more powerful and may capture strong Turing machine models (for details see [12]).

Exponential lower bounds on $(1, +k)$-BP were shown in [12, 20–22] for various parameters k. Lower bounds from [12] hold for $k = \Omega\left(\frac{n}{\log n}\right)$ and the lower bound from [10] holds even for $k = \Omega(n)$, where n is the number of input bits. We refer the reader to the books [11, 25] with the detailed description of results related to branching programs.

Lower bounds for $(1, +k)$-BP described above are given for "complicated" functions (usually it is characteristic functions of an error-correcting code with additional properties). In particular, these functions are complicated in terms of the certificate complexity. Unfortunately, for some applications, it is not enough. Following [14], we have a connection between proof systems and branching programs in application to the "unsatisfied clause search problem". Hence for the lower bounds in proof complexity we want to deal with this search problem, which is an "easy" problem. Namely, it can be described by a small collection of certificates. In this paper we introduce a technique for proving such lower bounds on the semantic $(1, +k)$-BP where $k = \mathcal{O}(\log n / \log\log n)$ where n is the number of variables.

1.1 Search Problem and Proof Systems

Consider a **search problem**, defined by a relation $S \subseteq \mathcal{I} \times \mathcal{O}$ for some finite sets $\mathcal{I}$ and $\mathcal{O}$. On input $x \in \mathcal{I}$ the search problem is to find some output in $S(x) := \{o \in \mathcal{O} \mid (x, o) \in S\}$. In this paper we study the "unsatisfied clause search problem" for a CNF formula.

▶ **Definition 1.** *A* ***unsatisfied search problem*** Search_φ *for an unsatisfiable CNF formula* $\varphi := \bigwedge_{i \in I} C_i$ *on n variables is defined as follows:*

- *input: an n-variable assignment $z \in \{0, 1\}^n$;*
- *output: an element $i \in I$ such that clause C_i of φ is falsified by z.*

Informally speaking, we may think that if we can solve the Search_φ problem in some computational model $\mathfrak{C}$, then the description of $C \in \mathfrak{C}$ that solves Search_φ is a "certificate of unsatisfiability" of a formula φ. So we may think of this model as a proof system. We do not want to formalize this statement for a general computational model, but we prove the formal statement for the branching programs (see section 5).

The proof system that is defined by the read-once branching programs is equivalent to regular Resolution [14]. This connection is widely used in proof complexity. As an example, we can consider first lower bounds on the regular Resolution and Resolution proofs of the Weak Pigeonhole Principle [17, 18], recent breakthrough result: a lower bound on the regular Resolution proofs of the Clique formulas [5].

The connection between regular Resolution and branching programs makes it interesting to consider some less restricted models of branching programs in application to the Search_φ problems. Some of these models were considered in [13]. In this paper we focus on $(1, +k)$-BPs. Despite the success in proving lower bounds on the Resolution (and hence read-once programs) the lower bounds for $(1, +k)$-BP on the Search_φ are an open question even for $k = 1$.

Previous Techniques

The behaviour of branching programs on functions differs from the behaviour on the Search_φ problem. For example, the unrestricted programs may solve Search_φ for any φ in linear size (we can implement a simple algorithm that checks clauses of φ step by step). Informally speaking, as we said above, the lower bounds for functions on $(1, +k)$-BP [12, 20–22] heavily used the fact that there is no efficient description of these functions in terms of certificates. However Search_φ is defined by a small set of certificates. Considering this difference, it is unclear how to use the classical techniques for our problem.

Another issue is that $(1, +k)$-BP is much stronger than general Resolution on some classes of formulas [13] even for small constant k and syntactic model. This is a crucial observation and it means that we cannot directly apply general techniques for proving lower bounds in proof complexity like [3, 6] etc., since these techniques cannot distinguish between considered classes of formulas and other hard examples for Resolution. Hence if we want to prove lower bound for $(1, +k)$-BP on Search_φ we need some additional arguments in comparison to Resolution lower bounds.

1.2 Our Results

The main result is an exponential lower bound on the size of $(1, +k)$-BPs in application to Search_φ for $k = \mathcal{O}\left(\frac{\log n}{\log \log n}\right)$.

▶ **Theorem 2.** *For $n \in \mathbb{N}$ and $k_0 := k_0(n)$ there is an unsatisfiable formula φ on n variables of size $n^{\mathcal{O}(k_0)}$ such that any semantic $(1, +k_0)$-BP solving Search_φ requires size $\exp\left[\Omega\left(\frac{n}{2^{\mathcal{O}(k_0)}}\right)\right]$.*

We also show that $(1, +k)$-BPs define a proof system in terms of Cook–Reckhow definition [7] $(1, +k)$-BP-PS (see section 5). That is a generalization of the result from [13], where it was shown only for ℓ-CNF where ℓ is an absolute constant.

▶ **Theorem 3.** *A syntactic $(1, +k)$-BP-PS is a proof system in terms of Cook–Reckhow definition for any constant $k \in \mathbb{N}$.*

1.3 Technique

The key ingredients for the lower bound are:

- *garlands:* aka (s, ℓ)-chains, that is a standard technique for proving lower bounds on the branching programs [11, 12, 20–22];
- *closure:* a technique that allows to make large partial restriction and keep the search problem hard for branching programs (and proof systems) [1, 3];

- *amplification:* a trick from [2] that makes formula hard for regular Resolution (and read-once branching programs) and help us to force the branching program to use the repetitions in a very structured way;
- *Flow-Cut:* the famous Theorem [8] that shows the duality between the maximum flow and the minimum cut, that we use to extend partial assignments to total assignments with good properties.

Let us introduce a general sketch of the proof. In section 4.1 we define an unsatisfiable formula Flow_G [3] that states: in graph G we have a source of a flow but there is no sink. We require graph G to be an algebraic expander, but, in fact, we need two properties:

- G is a combinatorial expander; namely, each set of vertices of size at most $r = \Omega\left(\frac{n}{\log n}\right)$ has a lot of neighbours (this is a "local" property, since we care only about small enough sets);
- max-balanced-cut of the graph G is large enough (this is a "global" property of the graph G), where "balanced" means that each piece has size at least $\Omega(r)$.

It is not clear how to show the lower bound for this formula itself and we amplify Flow_G formulas by using the trick from [2]. Denote the result of amplification by φ.

1. For the sake of contradiction we assume that we have a small $(1, +k)$-BP solving Search_φ. We generate a big family of paths and, using the upper bound on the size of our program, we find some paths in the program that form a "garland" structure (see section 4). This idea is similar to the idea from [12].
2. These paths correspond to some assignments and we keep our formula "hard" under these assignments. To do it we use a modification of the "closure" technique [3] (an easier version of this iterative modification was used in [23]). Here we use a combinatorial expansion of the graph G.
3. By using the fact that we deal with an amplified version of Flow_G we show that from the end point of paths that form the garland we cannot reach any leaf that is marked by one of the clauses from some set $T \subseteq \varphi$. Here we use the fact that k is small enough.
4. To conclude the proof, we use the Max-Flow Min-Cut Theorem (and global properties of our graph) to show that there should be some path from the garland to some clause from the set T.

   See section 4 for more details.

2 Preliminaries

Let X be a set of boolean variables. For a variable $x \in X$ we denote $x^1 := x$ and $x^0 := \neg x$. We say that $\alpha \colon X \to \{0, 1, *, ?\}$ is a **generalized partial assignment** and α **assigns** or **touches** $x \in X$ iff $\alpha(x) \in \{0, 1, ?\}$. And an assignment γ is an **instance** of α iff:

- $\alpha(x) \in \{0, 1, *\}$ implies $\gamma(x) = \alpha(x)$;
- $\alpha(x) =?$ implies $\gamma(x) \in \{0, 1\}$.

If α and β are two partial assignments to variables from the set X, we say that a generalized partial assignment $\alpha \uplus \beta \colon X \to \{0, 1, *, ?\}$ is a **joint assignment** iff:

- if $\alpha(x) = a$ and $\beta(x) \in \{a, *\}$, then $\alpha \uplus \beta(x) = a$;
- if $\beta(x) = a$ and $\alpha(x) \in \{a, *\}$, then $\alpha \uplus \beta(x) = a$;
- if $\alpha(x) = a$ and $\beta(x) = 1 - a$, then $\alpha \uplus \beta(x) =?$;
- if $\alpha(x) = \beta(x) = *$, then $\alpha \uplus \beta(x) = *$,

where $a \in \{0, 1\}$.

We will also use the famous Max-Flow Min-Cut Theorem.

▶ **Theorem 4** (Max-Flow Min-Cut [8]). *Let $G := (V, E)$. For any $s, t \in V$ the maximum value of an s-t flow is equal to the minimum capacity over all s-t cuts.*

2.1 Branching Programs

Let $X := \{x_1, \dots, x_n\}$ be a set of propositional variables and $\mathcal{O}$ be a finite set. A **branching program** is a directed acyclic graph with one source. Every vertex of the graph is labeled by a variable from X, or by an element of the set $\mathcal{O}$ with respect to the following properties:

- if a vertex is labeled by $o \in \mathcal{O}$, then it is a sink;
- if a vertex is labeled by a variable, then it has exactly two outgoing edges: one edge is labeled by 0 and the other one is labeled by 1.

Every branching program B defines a function $f_B\colon \{0,1\}^n \to \mathcal{O}$. We assume that every input $z \in \{0,1\}^n$ induces a path from source to sink in a natural way. If this path ends in a vertex with a label $o \in \mathcal{O}$ then we define $f_B(z) := o$.

We say that B is a **branching program for the relation $\boldsymbol{S \subseteq \{0,1\} \times \mathcal{O}}$** iff f_B is consistent with S: namely if $f_B(z) = o$ then $(z, o) \in S$.

Let D be a branching program and v is a node in it. The **subprogram** of D with the root v we denote by $D(v)$ and define as a subgraph of D that is reachable from v. Also for a partial assignment ρ we define a branching program $D|_\rho$ as the following transformation applied to D:

- for each variable y to which ρ assigns a value a, contract edges $y = a$ and delete edges $y = \neg a$;
- delete all vertices that are unreachable from the root.

These operations only decrease the size of the program.

If p is a consistent path in a branching program, we denote a partial assignment that corresponds to this path by τ_p.

Let us also define some classical restrictions of the general branching programs.

▶ **Definition 5.** *Let B be a branching program. We say that B is a* ***(syntactic) read-once branching program*** *or* 1-BP *iff on every path from the source to a sink we can see each variable at most once.*

We say that B is a **$\boldsymbol{(1, +k)}$-BP** *iff on every path p from the source to a sink there is a set of variables X_p of size at most k such that all other variables appear in p at most once. And we can twist this definition a little bit and say that B is a* ***semantic*** **$\boldsymbol{(1, +k)}$-BP** *iff on every consistent path from p source to sink there is a set of variables X_p of size at most k such that all other variables appear in p at most once.*

If a branching program B computes a boolean function, we say that it is **satisfiable** iff f_B is not identically zero.

▶ **Theorem 6** (Savický [19]). *There is an algorithm to check a satisfiability of a syntactic* $(1, +k)$-BP *in time* $\mathcal{O}\left[\left(\frac{4en}{k}\right)^k sn\right]$.

The following algorithm also will be useful for us.

▶ **Theorem 7** (Savický [19]). *The test whether an input branching program is a syntactic* $(1, +k)$-BP *can be done in time* $\mathcal{O}\left[\left(\frac{3en}{k+1}\right)^{k+1} s\right]$.

The next observation is natural and extremely useful for proving lower bounds.

▶ **Lemma 8.** *Let D be a $(1,+k)$-BP for Search_φ, p be a consistent path from the root to some node v. If p has a variable x queried more than one time on it then $D(v)|_{\tau_p}$ is a $(1,+(k-1))$-BP for the $\mathsf{Search}_{\varphi|_{\tau_p}}$. The result holds for both: semantic and syntactic models.*

Proof. A program $D(v)|_{\tau_p}$ is a program for the $\mathsf{Search}_{\varphi|_{\tau_p}}$ by the correctness of the program D. Consider a path s in D from v to some leaf. Let X_s be a set of variables that are queried more than one time on s. If $|X_s| = k$ and $x \notin X_s$, the path ps has at least $k+1$ variables that are queried more that one time. This is a contradiction. If $|X_s| = k$ and $x \in X_s$, note that in $D(v)|_{\tau_p}$ we contract all edges that correspond to the x variable and hence we transform this path into a path with at most $k-1$ repetitions. ◀

3 Expanders

We are given a graph $G := (V,E)$. For two subsets of vertices A, B we write $E(A,B)$ to denote the set of pairs (v,e) where $v \in A$, e is an edge that is incident to v and e connects v with some vertex in B. We will think about it as about set of edges between A and B, but if A and B intersect we count edges within intersection twice. We also use a shortcut notations $E(S) := E(S,V)$ and $\overline{S} := V \setminus S$. If the graph we consider is unclear from the context we specify it as a subscript: $E_G(A,B)$.

▶ Remark 9. Assuming that G is Δ-regular graph this definition allows us to use natural equalities:

- $|E(S)| = \Delta|S|$;
- $|E(A,\overline{A})| = \Delta|A| - |E(A,A)|$.

We write $\mathrm{N}_G(v)$ to denote the set of **neighbours** of v in the graph G. We extend this notion to sets and denote by $\mathrm{N}_G(S) := \{v \mid \exists u \in S,\ (u,v) \in E\}$ the **neighbourhood** of a set of vertices $S \subseteq V$.

A graph $G := (V,E)$ is an **(n,Δ,α)-algebraic expander** (or just **expander**), if:

- $|V| = n$;
- the degree of any vertex $v \in V$ equals Δ;
- the absolute value of the second largest eigenvalue of the adjacency matrix of G is at most $\alpha\Delta$.

▶ **Lemma 10** (Mixing Lemma [4]). *Let $G := (V,E)$ be an (n,Δ,α)-expander. For any two subsets $A, B \subseteq V$ the following holds:*

$$\left| |E(A,B)| - \frac{\Delta|A||B|}{n} \right| \le \alpha\Delta\sqrt{|A||B|}.$$

We also need *combinatorial* edge expansion. We say that $G := (V,E)$ satisfies **(r,β)-(edge) expansion property** for some $r, \beta > 0$, if for all $S \subseteq V$ of size at most r holds $E(S,\overline{S}) \ge \beta\Delta|S|$. The Mixing Lemma says that any expander graph satisfies expansion property for suitable parameters.

▶ **Corollary 11.** *If $G := (V,E)$ is an (n,Δ,α)-expander, then for any $0 < \beta < 1-\alpha$ the graph G satisfy $((1-\alpha-\beta)n,\beta)$-expansion property.*

Proof. Consider some $A \subseteq V$ of size at most $(1-\alpha-\beta)n$. Note that $|E(A,\overline{A})| = \Delta|A| - |E(A,A)|$. By Mixing Lemma:

$$|E(A,A)| \le \frac{\Delta|A|^2}{n} + \alpha\Delta|A| = \Delta|A|\left(\frac{|A|}{n} + \alpha\right) \le \Delta|A|(1-\beta).$$

Hence $|E(A,\overline{A})| \ge \beta\Delta|A|$ by Remark 9. ◀

The "vertex analog" of the next proposition is well known in the literature (for example [9]). We turn it into edge version.

▶ **Proposition 12.** *Let $G := (V, E)$ be a graph of degree Δ. If G satisfies (r, β)-expansion property then for any set $S \subseteq V$ of size $k \leq r$ there is an enumeration $v_1, v_2, \dots, v_k \in S$ and a sequence $R_1, \dots, R_k \subseteq E(S)$ such that:*

- $R_i = E(\{v_i\}, V \setminus \{v_1, v_2, \dots, v_i\})$;
- $|R_i| \geq \beta\Delta$.

Proof. We create this sequence in reversed order. Since $|S| \leq r$, it holds that $|E(S, \overline{S})| \geq \beta\Delta|S|$ and there is a vertex $v_k \in S$ such that $|E(\{v_k\}, \overline{S})| \geq \beta\Delta$. Let $R_k := |E(\{v_k\}, \overline{S})|$, and repeat the process for $S \setminus \{v_k\}$. ◀

4 Lower Bounds for $(1, +k)$-BP

In this section, we will prove the following theorem:

▶ **Theorem 13** (2). *For $n \in \mathbb{N}$ and $k_0 := k_0(n)$ there is an unsatisfiable formula φ on n variables of size $n^{\mathcal{O}(k_0)}$ such that any semantic $(1, +k_0)$-BP solving* Search$_\varphi$ *requires size* $\exp\left[\Omega\left(\frac{n}{2^{\mathcal{O}(k_0)}}\right)\right]$.

Let us describe the main ideas used in the proof. To prove this Theorem we would like to construct an exponentially big set of paths, which cannot be compactly "glued" together in $(1, +k)$-BP, correctly solving Search$_\varphi$.

To give a detailed plan we need an auxiliary definition.

▶ **Definition 14.** *A **ℓ-garland** in a branching program is a pair of paths (a, b) from the root such that $a := v_0 a_1 v_1 a_2 v_2 a_3 \dots a_\ell v_\ell$ and $b := v_0 b_1 v_1 b_2 v_2 b_3 \dots b_\ell v_\ell$ where a_i, b_i are possibly empty paths and paths $v_j a_{j+1} v_{j+1}$ and $v_j b_{j+1} v_{j+1}$ are different for all $0 \leq j < \ell$ (see Fig. 1).*

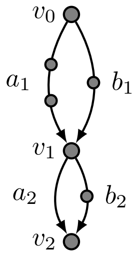


Figure 1 2-garland.

Let us consider the detailed plan.

1. By induction on k we want to show that Search$_{\varphi|_\rho}$ is hard for $(1, +k)$-BP even after some "good" restriction ρ.
2. For the sake of contradiction we assume that we have a small $(1, +k)$-BP solving Search$_{\varphi|_\rho}$. In the section 4.2.1 we generate a family of paths starting from the root of the program and find in this family a $(k + 1)$-garland (see Fig. 1). This idea is similar to [12].
3. To argue that we can find a garland we generate exponentially many paths by walking from root (section 4.2). During this process, we have to make sure that on these paths our branching program cannot determine an answer (that would mean that we cannot walk anymore). To avoid it we use the "closure" technique that is motivated by technique from [1,3] and avoid "local contradictions". And hence we have to choose the formula φ very carefully, but we still have some freedom.

4. If we found a repetition while constructing a garland, we use Lemma 8 and apply induction hypothesis. This is a place where we use that formula φ is still hard even after the restriction.
5. In the section 4.2.2 we combine different parts of garland and argue about the reachability of certain leaves. We have to make sure that the paths we consider are consistent and that when we reach the endpoint of the garland, formula φ remains hard. We achieve it by using the following properties.
   - We have already removed repetitions from the garland by using Lemma 8 and induction hypothesis.
   - To show that combinations of different parts of the garland give us consistent paths we equip the closure technique by the notion of "strongly satisfied" (see Section 4.1.1) constraints. This is the second place that requires specific properties of the formula φ.

   At the end of this section, we will have a set of clauses $\mathfrak{C} \subseteq \varphi$ such that leaves marked by elements of this set should be unreachable from the endpoint of the garland.
6. For the last part (section 4.2.3) we consider an arbitrary path r in our garland and note that $\varphi \setminus \mathfrak{C}$ is a satisfiable formula even under the restriction τ_r. It is hard to show this property for the formulas that encode natural combinatorial principles. We use the trick from [2] to change the formula φ to make sure that $\mathfrak{C}$ is large enough.

   Here we use the global structure of our formula φ (in our case we use the Max-Flow Min-Cut Theorem) to satisfy all clauses in $\varphi \setminus \mathfrak{C}$.

We start with defining the hard formulas on a suitable expander graph.

4.1 Hard Formulas

Let $G := (V, E)$ be a directed graph. Each edge $e \in E$ has the corresponding variable x_e, where $x_e = 1$ indicates that a flow of size 1 is going through an edge e. Let u be an arbitrary, but fixed vertex of the graph.

The formula $\mathsf{Flow}_{G,u}$ consists of the following constraints written in CNF for all $v \in V$:

$$\sum_{e \in E : \mathrm{st}(e)=v} x_e - \sum_{e \in E : \mathrm{en}(e)=v} x_e \geq c(v),$$

where $e = (\mathrm{st}(e), \mathrm{en}(e))$ and $c \colon V \to \{0, 1\}$ is a labeling function:

- $c(v) = 0$, for all $v \in V \setminus \{u\}$;
- $c(u) = 1$.

This formulas states: for all vertices in the graph the flow is non-negative, and at least for one vertex it is strictly positive. It is easy to see that $\mathsf{Flow}_{G,u}$ is unsatisfiable. We omit index u since in our applications it is an arbitrary vertex.

We use the most naive CNF encoding of these constraints. We represent each constraint separately. Consider a vertex $v \in V$ and a set of edges $E_v := \{e_1, e_2, \ldots, e_s\} \subseteq E$ that are incident to v. Let $\rho_v \colon E_v \to \{0, 1\}$ be an assignment that violates the constraint in v. In this case we add to the formula a clause C:

$$x_{e_1}^{1-\rho(x_{e_1})} \vee x_{e_2}^{1-\rho(x_{e_2})} \vee \cdots \vee x_{e_s}^{1-\rho(x_{e_s})},$$

and we also say that this assignment has a **gap**:

$$g(\rho_v) := c(v) - \sum_{e \in E : \mathrm{st}(e)=v} \rho_v(x_e) + \sum_{e \in E : \mathrm{en}(e)=v} \rho_v(x_e).$$

For our purpose we consider Flow_G based on expanders. To be precise, we start with a graph G that is an (n, Δ, α)-expander, where $\Delta = \Theta(\log n)$ and α is some fixed constant, and replace each undirected edge by two directed edges (we say that these edges are **dual**). The exact value of Δ depends on a value of k.

▶ Remark 15. We consider only **proper** partial assignments ρ that satisfy the following property for all pairs of dual edges (e, e'):

- $\rho(x_e) \in \{0, 1\}$ iff $\rho(x_{e'}) \in \{0, 1\}$;
- if $\rho(x_e) = 1$ then $\rho(x_{e'}) = 0$.

We also identify $\mathrm{supp}(\rho)$ with an undirected set of edges that are assigned by ρ.

To make the formula somewhat "confusing" for $(1, +k)$-BP, we would like to add more variables to clauses. These variables do not really affect the physical meaning of the formula, but make it hard for $(1, +k)$-BP to extract additional information from repetitions on paths. This transformation is sensitive to the exact CNF encoding of the constraints that is written above.

▶ **Definition 16.** *Let $G := (V, E)$ be an undirected graph and $\mathcal{C}_v$ be a subset of clauses corresponding to vertex v in Flow_G. Let $\eta_v^k : \mathcal{C}_v \to \binom{E}{k}$ be a mapping, and $\eta^k := \{\eta_v^k \mid v \in V\}$ be a family of such mappings. We define $\mathsf{Flow}_G^{\eta^k}$ the following way:*

- *for each $v \in V$ we consider each $C \in \mathcal{C}_v$;*
- *we take $\eta_v^k(C) = \{e_1, \dots, e_k\}$, which is a set of k edges;*
- *we replace C by 2^{2k} clauses of the form:*

$$C \vee \bigvee_i x_{s_i}^{a_i} \vee x_{s'_i}^{a'_i}$$

enumerated by $a_i, a'_i \in \{0, 1\}$, where $i \in [k]$ and s_i, s'_i are directed copies of the edge e_i.

As described in the plan, at some point in the proof we would like to construct an assignment that leaves certain clauses (to which a certain set of variables was added) unsatisfied. For our purpose, we would like those clauses to "strongly unsatisfy" the condition in their vertices.

Let us describe the construction of η^k. Assume that $\Delta \geq 50 \cdot k \log n$. For each $v \in V$ we define η_v independently. We will be interested in adding variables to clauses which correspond to large incoming flow.

1. Let us consider a set of clauses $\mathcal{C}$ that corresponds to v and a proper partial assignment on edges incident to v with gap equal to $\frac{\Delta}{4} + 1$.
2. Note that $|\mathcal{C}| \geq \binom{\Delta}{\Delta/4} \geq 4^{\Delta/4} \geq n^{4k}$. The first inequality holds since we can choose arbitrary $\Delta/4 + 1$ incoming edges to obtain the desired gap and set all other incident edges to zero.
3. There are at most $\binom{n\Delta}{k} \leq \left(\frac{n\Delta e}{k}\right)^k \leq n^{2k}$ different sets of k edges. Hence we can choose a subset of $B \subseteq \mathcal{C}$ and define η_v^k to be a bijection between B and all possible choices of sets of k edges.

Note that the existence of $(1, +k)$-BP of size S solving $\mathsf{Search}_{\mathsf{Flow}_G^{\eta^k}}$ (for any η^k) implies the existence of $(1, +k)$-BP of size S solving $\mathsf{Search}_{\mathsf{Flow}_G}$.

4.1.1 Locally Consistent Assignments

We need a notion of "good assignments", i.e. assignments that reduce Flow_G formulas to smaller, but "equally hard" instances.

Let $G := (V, E)$ be a graph. A proper assignment ρ δ-satisfies a set of vertices $U \subseteq V$ iff for all $v \in U$ the following holds:

- ρ assigns all edges that are incident to v;
- ρ satisfies the constraint for v;
- $\sum_{e \in E : \mathrm{st}(e) = v} \rho(x_e) \geq \delta \cdot \Delta$.

We also say that a proper assignment ρ is **(r, δ, β)-locally consistent** iff there is a set of vertices V_ρ of size at most r such that:

- ρ δ-satisfies V_ρ;
- $(V \setminus V_\rho, E \setminus \mathrm{supp}(\rho))$ satisfies (r, β)-expansion property.

▶ Remark 17. If ρ is an (r, δ, β)-locally consistent assignment for some $\beta > 0$, then V_ρ is uniquely defined.

Proof. For the sake of contradiction assume that there are two candidates A, B. Wlog $A \setminus B \neq \emptyset$. Pick an arbitrary vertex $v \in A \setminus B$. Since A satisfies the required properties, ρ assigns all edges that are incident to v, which contradicts the fact that $(V \setminus B, E \setminus \mathrm{supp}(\rho))$ satisfies (r, β)-expansion property. ◀

4.2 Proof of Theorem 2

Let G be an (n, Δ, α)-expander and η^{k_0+1} be a mapping defined in section 4.1. In this section we prove an exponential lower bound on $\mathsf{Search}_{\mathsf{Flow}_G^{\eta^{k_0+1}}}$ for $(1, +k_0)$-BP. We assume that n is large enough.

Let us fix some parameters:

- $\Delta := 100 k_0 \log n$ and $\Delta > 200$;
- $\alpha := 0.01$ is the second eigenvalue of the normalized adjacency matrix of G;
- $r := \frac{n}{\Delta}$ and $\beta := 0.96$ is the "combinatorial expansion" of the graph G;
- $\beta' := 0.95$ is an expansion parameter that we try to maintain after removing some vertices and edges from G;
- $\nu_k := \left(\frac{1}{4}(\beta - \beta')\right)^{k+3}$ is a scaling factor that indicates the fraction of edges that we want to assign in our partial assignment.

Note that $r \ll (1-\beta-\alpha)n = 0.03 \cdot n$ and hence by Corollary 11 G satisfies (r, β)-expansion property, hence we can use all combinatorial expansion properties and tools.

To formulate the induction hypothesis we need one more definition. Let $M \subseteq E$ and ρ is a proper assignment. We say that ρ is **γ-minimal local consistent extension** or ***(mlce)*** on M iff:

- ρ is $(r, 0.6, \gamma)$-locally consistent assignment;
- $\mathrm{supp}(\rho) = M \cup E(V_\rho)$;
- $|E(V_\rho, \overline{V_\rho}) \setminus M| < \gamma \Delta |V_\rho|$.

Informally we may think about it in the following way: after we assign edges from M somehow, ρ should assign also V_ρ as a "minimal" set of vertices to take care of in order to be locally consistent.

Let $\varphi := \mathsf{Flow}_G^{\eta^{k_0+1}}$. By induction on $k \leq k_0$ we show the following statement. For all sets of edges $M \subseteq E$ of size at most $\nu_k \Delta r$ and all β'-mlce ρ on M any $(1, +k)$-BP for $\mathsf{Search}_{\varphi|_\rho}$ has size at least $2^{\frac{\nu_k}{4(k+1)^2} \Delta r}$.

Fix some M, ρ, $0 \leq k \leq k_0$ and for the sake of contradiction assume that we have a $(1, +k)$-BP D of size $2^{\frac{\nu_k}{4(k+1)^2} \Delta r}$ for $\mathsf{Search}_{\varphi|_\rho}$.

4.2.1 Construction of the Garland

To fulfill our plan of the proof, described at the beginning of the section, we start constructing the garland by obtaining an exponentially big set of paths with the corresponding assignments. Let us remind that $|M| \le \nu_k \Delta r$ and ρ is β'-mlce on M.

We say that triple (p, U_p, σ_p) is **γ-good** iff:

- p is a path from the root of the branching program;
- U is a subset of edges such that corresponding variables are queried on p, so-called "branching variables";
- σ_p is a partial assignment such that:
  - σ_p extends $\rho \cup \tau_p$;
  - σ_p is a γ-mlce on $M \cup U_p$;
  - σ_p 0.8-satisfies $V_{\sigma_p} \setminus V_\rho$,

  where τ_p is an assignment that corresponds to p.

We maintain the set of β'-good triples $\mathcal{P}$ and an auxiliary set $\mathcal{S}$ of triples that appear in the set $\mathcal{P}$ at some moment during the process. In the beginning of our construction $\mathcal{P} \coloneqq \{(\emptyset, \emptyset, \rho)\}$ and $\mathcal{S} \coloneqq \mathcal{P}$.

We repeat the following process while we have at least one triple $(p, U_p, \sigma_p) \in \mathcal{P}$ such that $|U_p| \le \nu_k \Delta r$.

Consider the triple described above. Let v be the end of p and x_e be the variable asked in v.

1. If x_e was queried on p we stop the process. In this case we return "Repetition" and we remember the path p.
2. Erase the triple (p, U_p, σ_p) from $\mathcal{P}$.
3. If $\sigma_p(x_e) \in \{0, 1\}$, then we continue along the edge $x_e = \sigma_p(x_e)$. Consider a path p' that is the extension of p along this edge, $U_{p'} \coloneqq U_p$ and $\sigma_{p'} \coloneqq \sigma_p$. Put $(p', U_{p'}, \sigma_{p'})$ into $\mathcal{P}$ and $\mathcal{S}$ and repeat the process from the beginning.
4. If $\sigma_p(x_e) = *$, then it is a "branching node", and we call this step a **branching step**.
   a. Let p' be a path obtained by continuing p along the edge $x_e = 0$, and p'' be a path obtained by continuing p along the edge $x_e = 1$.
   b. $U_{p'} \coloneqq U_p \cup e$, $U_{p''} \coloneqq U_p \cup e$.
   c. $\tau' \coloneqq \sigma_p \cup \{x_e = 0, x_{e'} = 0\}$, $\tau'' \coloneqq \sigma_p \cup \{x_e = 1, x_{e'} = 0\}$, where $x_{e'}$ is a dual edge.
   d. $(p', U_{p'}, \tau')$ is $(\beta' - 0.01)$-good triple. We extend an assignment τ' to make this triple β'-good. For the formal statement see Lemma 18. Here we describe an idea. Let $R \subseteq E$ be a set of edges that are unassigned by τ' (or τ''), and $B \subseteq V \setminus V_{\sigma_p}$ be the maximal set of vertices that satisfies:
      - $|B| \le r$;
      - $|E(B, \overline{B}) \cap R| \le \beta' \Delta |B|$.

      Let κ be an assignment on variables that correspond to edges in the set $E(B) \setminus \mathrm{supp}(\tau')$ such that $\tau' \cup \kappa$ 0.8-satisfies the constraints for all $v \in B$. This assignment κ always exists (and moreover it is independent of the value of x_e, but we do not use this fact).
   e. We denote $\sigma_{p'} \coloneqq \tau' \cup \kappa$, $\sigma_{p''} \coloneqq \tau'' \cup \kappa$ and put $(p', U_{p'}, \sigma_{p'})$ and $(p'', U_{p''}, \sigma_{p''})$ into $\mathcal{P}$ and into $\mathcal{S}$.

To conclude the construction we want to show the following claims.

- **Repetition case.** In the first case of the proof (if we have a repetition) we can reduce the problem to a lower bound on $(1, +(k-1))$-BP.
- **Correctness.** The branching step can be done and triples $(p', U_{p'}, \sigma_{p'})$ and $(p'', U_{p''}, \sigma_{p''})$ satisfy the required properties.
- **Garland extraction.** Among these paths we can find an k-garland (a, b) and a locally consistent extension of ρ.

4.2.1.1 Correctness

We show that if we have a triple (p, U_p, τ_p) which is β'-good then after processing it with our algorithm we also put in our sets β'-good triples. Let us formulate the general Lemma that helps us with it.

▶ **Lemma 18.** *Let (p, U_p, σ_p) and (q, U_q, σ_q) be* 0.9*-good triples. Then there is an assignment κ such that:*

- *for any γ that is an instance of $\sigma_p \uplus \sigma_q$ an assignment $\gamma \cup \kappa$ is a β'-mlce on* $\mathrm{supp}(\sigma_p) \cup \mathrm{supp}(\sigma_q)$*;*
- $|\,\mathrm{supp}(\gamma \cup \kappa)| \leq \nu_{k-1}\Delta r$*.*

Moreover if $p = q$ then triple $(p, U_p, \sigma_p \cup \kappa)$ is β'-good.

Proof. The proof was motivated by the closure technique developed in [1,3]. For the full version of the proof see Appendix A. ◀

If the branching step was not done, then we do not change U and τ, and we extend the path p according to the assignment τ hence the triple remains β'-good. We are left with the branching step. Note that $(p', U_{p'}, \tau')$ is 0.9-good and we apply Lemma 18 to a pair composed of two identical triples $(p', U_{p'}, \tau')$ and obtain κ that satisfies the required properties.

4.2.1.2 Repetition case

First let us note that if there is a repetition, then $k > 0$. Suppose we found a repetition while considering a triple (p, U_p, σ_p). The size of $M \cup U_p$ is at most $2\nu_k\Delta r$ and σ_p is β'-mlce on $M \cup U_p$. Let v be an end node of p. The program $D(v)|_{\sigma_p}$ is a $(1, +(k-1))$-BP for $\mathsf{Search}_{\mathsf{Flow}_G^{\eta^k}|_{\sigma_p}}$ by Lemma 8. Thus by induction hypothesis we have a lower bound of $2^{\frac{\nu_{k-1}}{4k^2}\Delta r} \geq 2^{\frac{\nu_k}{4(k+1)^2}\Delta r}$ on the size of $D(v)|_{\sigma_p}$ and in this case we are done.

4.2.1.3 Garland extraction

The following Lemma gives us a pair of triples $(p, U_p, \sigma_p), (q, U_p, \sigma_q) \in \mathcal{P}$ such that (p, q) forms a $(k+1)$-garland.

▶ **Lemma 19.** *There are $(p, U_p, \sigma_p), (q, U_q, \sigma_q) \in \mathcal{S}$ such that (p, q) forms a $(k+1)$-garland.*

Proof. For the proof see Appendix B. ◀

To continue the proof we need some additional property that we can "avoid repetitions" in this garland. We say that there is a **repetition in a garland** $p = v_0 p_1 v_1 p_2 v_2 p_3 \ldots p_{k_0+1} v_{k_0+1}$ and $q = v_0 q_1 v_1 q_2 v_2 q_3 \ldots q_{k_0+1} v_{k_0+1}$ iff there is **path in the garland**, i.e. path r of the form $v_0 r_1 v_1 r_2 v_2 r_3 \ldots r_{k_0+1} v_{k_0+1}$, such that some variable is queried more than one time on it, where $r_i \in \{p_i, q_i\}$.

Consider a path r in our garland (p, q) that contains a repetition and $r' \subseteq r$ the largest initial segment of r without repetitions. Let v be its end node. We apply Lemma 18 to triples $(p, U_p, \sigma_p), (q, U_q, \sigma_q)$, which gives us assignment κ, and choose a instance γ of $\sigma_p \uplus \sigma_q$ that is consistent with $\tau_{r'}$. Moreover, $|\,\mathrm{supp}(\gamma \cup \kappa)| \leq \nu_{k-1}\Delta r$, and $\gamma \cup \kappa$ is a β'-mlce on $\mathrm{supp}(\sigma_p) \cup \mathrm{supp}(\sigma_q)$. Hence by Lemma 8 we can use the induction hypothesis for $(1, +k-1)$-BP $D(v)|_{\tau_{r'}}$ and formula $\varphi|_{\gamma \cup \kappa}$. The size of $D(v)|_{\tau_{r'}}$ is at least $2^{\frac{\nu_{k-1}}{4k^2}\Delta r} \geq 2^{\frac{\nu_k}{4(k+1)^2}\Delta r}$.

For the rest of the proof we can assume that on any path r of the form described above there are no repetitions.

4.2.2 Unreachable Leaves

Let us summarize what we have from the previous section. We created a pair of triples: (p, U_p, σ_p) and (q, U_q, σ_q) such that:

- (p, q) forms $(k_0 + 1)$-garland:
  - $p = v_0 p_1 v_1 p_2 v_2 p_3 \dots p_{k_0+1} v_{k_0+1}$;
  - $q = v_0 q_1 v_1 q_2 v_2 q_3 \dots q_{k_0+1} v_{k_0+1}$;
- (p, U_p, σ_p) and (q, U_q, σ_q) are β'-good;
- there are no repetitions on any path in the garland (p, q).

We use Lemma 18 for (p, U_p, σ_p) and (q, U_q, σ_q) and get an assignment κ. Let us fix an assignment γ that is an instance of $\sigma_p \uplus \sigma_q$ consistent with:

- τ_p;
- values in σ_q that do not contradict τ_p

and denote $\zeta := \gamma \cup \kappa$. Note that, by construction:

- $|\zeta| \le \nu_{k-1} \Delta r$;
- $|\zeta|$ is $(r, 0.6, \beta')$-locally consistent.

In this section we describe a set of clauses that should be unreachable from the vertex v_{k_0+1}. Note that on each segment of a garland $(v_i p_i v_{i+1}, v_i q_i v_{i+1})$ we query at least one variable in both assignments τ_p and τ_q and get the different values. Denote any variable that satisfies this property by x_i.

We remind that $\varphi := \mathsf{Flow}_G^{\eta^{k_0+1}}$. Let $\mathfrak{D}, \mathfrak{C}$ be the subsets of clauses:

$$\mathfrak{D} := \{D \in \mathsf{Flow}_G \mid \text{for every } e \text{ that corresponds to some } x_i : e \in \eta^{k_0+1}(D)\}.$$

and

$$\mathfrak{C} := \{C \in \varphi \mid C \text{ is obtained from some } D \in \mathfrak{D} \text{ by the amplification trick}\}.$$

For the sake of contradiction suppose that there is a path s from v_{k_0+1} such that:

- s is a consistent path and τ_s is consistent with ζ and hence ps is also consistent;
- s ends in a clause $C \in \mathfrak{C}$.

Consider a family of paths $r_i := v_0 p_1 v_1 p_2 v_2 p_3 \dots p_{i-1} v_{i-1} q_i v_i p_{i+1} q_{i+1} p_{i+2} \dots p_{k_0+1} v_{k_0+1}$, where $i \in [k_0 + 1]$. All paths r_i are consistent since there are no repetitions in the garland (p, q). Hence if r_i is inconsistent with s then on s we requery some variable x'_i from the segment $v_{i-1} q_i v_i$ and get an inconsistent value.

By construction, τ_s is consistent with ζ, and $\zeta := \gamma \cup \kappa$, where γ is an instance of $\sigma_p \uplus \sigma_q$. If x'_i appeared in $v_{i-1} q_i v_i$, but not in $v_{i-1} p_i v_i$ (note that it cannot appear in any other segment of the garland, since there are no repetitions on the garland), then $(\sigma_p \uplus \sigma_q)(x'_i) \in \{\sigma_q(x'_i), ?\}$ and $\tau_p(x'_i) = *$ thus $\gamma(x'_i) = \sigma_q(x'_i)$ by the choice of γ. It follows that $\zeta(x'_i) = \sigma_q(x'_i)$ as well, and since τ_s is consistent with ζ, we cannot obtain an inconsistent with τ_{q_i} value for x'_i while requerying it. Hence x'_i had appeared in $v_{i-1} p_i v_i$ as well, and on s we requeried a variable from $v_{i-1} p_i v_i$ in consistent way. Moreover if all paths from some set $\{r_i\}_{i \in L}$ where $L \subseteq [k_0 + 1]$ are inconsistent with s we requery at least $|L|$ variables from the path p on the path s. Hence at least one of the paths r_{i_0} is consistent with s, where $i_0 \in [k_0 + 1]$ (or on the path ps we requery at least $k_0 + 1$ variables).

▶ Remark 20. This is the only place there we use the property that there are no repetitions on the garland.

Consider two paths ps and $r_{i_0}s$:

- these paths are consistent;
- $\tau_{ps}(x_{i_0}) \neq \tau_{r_{i_0}s}(x_{i_0})$.

These properties imply that clause C is not a legal answer for at least one these paths, and we have a contradiction with the assumption that there is a consistent path from v_{k_0+1} to this clause. That gives us the desired description of leaves that should be unreachable for v_{k_0+1}.

To conclude the proof it remains to show that there should be a path from v_{k_0+1} to at least one leaf marked by a clause $C \in \mathfrak{C}$. We do it in the next section.

4.2.3 Directing the Flow

Let us remind that we deal with $\varphi := \mathsf{Flow}_G^{\eta^{k_0+1}}$. To show that there is a path consistent with ζ from v_{k_0+1} to a leaf with a label $C \in \mathfrak{C}$ we show that $(\varphi \setminus \mathfrak{C})|_\zeta$ is satisfiable and hence there should be an extension of ζ that violates only clauses from $\mathfrak{C}$.

▶ Remark 21. If we do not care about assignment ζ, the statement is trivial, since φ is so-called minimally unsatisfiable formula (that becomes satisfiable after removing any clause). But ζ transforms our formula to "heavily unsatisfiable" formula, since ζ 0.6-satisfies a lot of vertices (that was the crucial property that we used to create a garland).

Note that by construction of η^{k_0+1} for each $v \in V$ there exists a clause $D \in \mathfrak{D}$ that had originated from the constraint for v. For each v, we pick any such clause and denote it by D_v. We divide the rest of the proof into two parts.

1. "Local part". We find a carefully chosen large enough set of vertices $U \in V$ and an assignment $\tau \supseteq \zeta$ such that there is a set $V_\tau \supseteq (U \cup V_\zeta)$:
   - $(V \setminus V_\tau, E \setminus \operatorname{supp}(\tau))$ satisfies (r, β')-expansion property;
   - for all $v \in U$ the assignment τ violates D_v and hence τ assigns all edges incident to v;
   - for all $v \in V_\tau \setminus U$ the assignment τ satisfies constraint for v.

   For this part we use the simplified version of technique used for the garland creation.
2. "Global part". By using Max-Flow Min-Cut Theorem we show that τ can be extended to total assignment that satisfies constraints for vertices whose constraints are neither satisfied nor falsified by τ yet.

Since we satisfy all the constraints of $(\mathsf{Flow}_G \setminus \mathfrak{D})_\zeta$ this assignment also satisfies all constraints in $(\varphi \setminus \mathfrak{C})|_\zeta$ by the construction of the formula φ (clauses of φ are the weakened versions of the clauses Flow_G).

Before we proceed with the proof let us define the "overflow".

▶ **Definition 22.** *The **overflow** introduced by a locally consistent assignment σ is:*

$$\mathsf{of}_\sigma := 1 + \sum_{v \in V_\sigma} \left(\sum_{e \in E : \mathrm{st}(e) = v} x_e - \sum_{e \in E : \mathrm{en}(e) = v} x_e - c(v) \right).$$

Note that $\mathsf{of}_\zeta \leq |\zeta| + 1 \leq \nu_{k-1} \Delta r + 1$.

4.2.3.1 Local part

We start with the local part of the proof. In the beginning of our construction $U_0 := \emptyset$, $\tau_0 := \zeta$, $V_{\tau_0} := V_\zeta$ and $i := 0$.

We repeat the following process while $\mathsf{of}_{\tau_i} > 0$.

1. Choose a vertex u_i that is untouched by τ_i.
2. Let ρ_{u_i} be an assignment to edges that are incident to u_i such that D_{u_i} is unsatisfied by ρ_{u_i}.
3. $\tau' := \tau_i \cup \rho_{u_i}$. Since u_i is untouched by τ_i there is no intersection between ρ_{u_i} and τ_i.
4. Let $H_i \subseteq V \setminus V_{\tau_i}$ be the maximal set of vertices that satisfies:
   - $|H_i| \leq r$;
   - $|E(H_i, \overline{H_i} \setminus \{u_i\}) \setminus \mathrm{supp}(\tau_i)| \leq \beta' \Delta |H_i|$.

   Let κ_i be an assignment on variables that correspond to edges in the set $E(H) \setminus \mathrm{supp}(\tau')$ such that for all $v \in H_i$:

$$\sum_{e \in E:\mathrm{st}(e)=v} (\tau' \cup \kappa_i)(x_e) - \sum_{e \in E:\mathrm{en}(e)=v} (\tau' \cup \kappa_i)(x_e) = c(v).$$

5. $U_{i+1} := U_i \cup \{u_i\}$, $\tau_{i+1} := \tau' \cup \kappa_i$ and $V_{\tau_{i+1}} := V_{\tau_i} \cup H_i \cup \{u_i\}$.
6. $i := i + 1$.

Let ℓ be a number of iterations in this process. Let $U := U_\ell$ and $\tau := \tau_\ell$.

At first we give an upper bound on ℓ. Since for all i an assignment κ_i exactly satisfies vertices in H, inclusion of H into V_τ does not change the overflow. Assignment ρ_{u_i} violates $D_{u_i} \in \mathfrak{D}$ and by definition of η^{k_0+1}:

$$-\frac{\Delta}{4} - 1 \leq \sum_{e \in E:\mathrm{st}(e)=u_i} \rho_{u_i}(x_e) - \sum_{e \in E:\mathrm{en}(e)=u_i} \rho_{u_i}(x_e) \leq -\frac{\Delta}{4}.$$

Hence on each iteration $\mathsf{of}_{\tau_{i+1}} \leq \mathsf{of}_{\tau_i} - \frac{\Delta}{4}$ and $|U| \leq \frac{4|\zeta|}{\Delta}$ and $-\frac{\Delta}{4} - 1 \leq \mathsf{of}_\tau \leq 0$.

▶ **Lemma 23.** *For all $i \leq \ell$:*

- κ_i *exists;*
- $|V_{\tau_i}| \leq \frac{1}{(\beta-\beta')\Delta}(\mathrm{supp}(\zeta) + \Delta|U_i|)$ *and hence* $|\tau_i| \leq \frac{2}{(\beta-\beta')}(|\mathrm{supp}(\zeta)| + \Delta|U_i|)$*;*
- $(V \setminus V_{\tau_i}, E \setminus \mathrm{supp}(\tau_i))$ *satisfies* (r, β')*-expansion property.*

Proof. This Lemma may be considered as simplified version of Lemma 18. For the proof see Appendix A. ◀

To conclude the construction note that $\tau_i \leq \frac{10}{4}\nu_{k-2}\Delta r \leq \frac{\Delta}{4} r$ for all $i \leq \ell$ and we always can find the vertex untouched by τ_i.

▶ Remark 24. This is the only place where we use that $r \leq \frac{n}{\Delta}$.

4.2.3.2 Global part

Let $B := V_\tau \setminus V_\zeta$. For the vertex $v \in V$ the **overflow of v** is defined in the following way:

$$\mathsf{of}(v) := -\sum_{\substack{e \in \mathrm{supp}(\tau) \\ \mathrm{st}(e)=u}} \tau(x_e) + \sum_{\substack{e \in \mathrm{supp}(\tau) \\ \mathrm{en}(e)=u}} \tau(x_e) + c(v).$$

We want to create an auxiliary graph. Let $F^+ := \{v \in V \setminus V_\tau \mid \mathsf{of}(v) > 0\}$ and $F^- := \{v \in V \setminus V_\tau \mid \mathsf{of}(v) < 0\}$. See Fig. 2.

We define a graph $G' := (V', E')$ on vertices $V' := (V \setminus V_\tau) \cup \{s\} \cup \{t\}$, where s is a source and t is a sink. Edges E' include four groups:

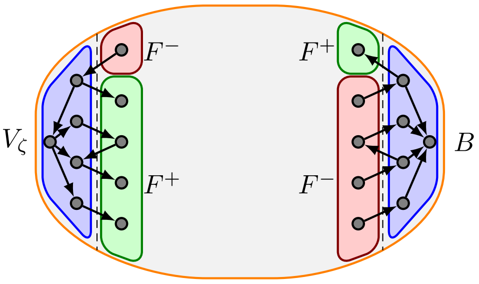


Figure 2 Set after assignment.

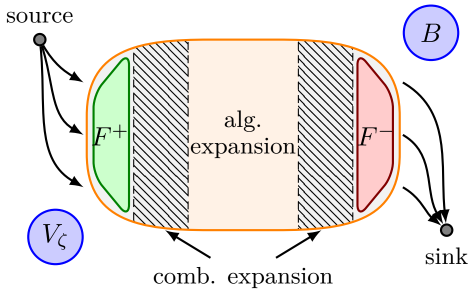


Figure 3 Graph G' with cuts.

- $E \setminus \mathrm{supp}(\tau)$;
- we connect s with all $v \in F^+$ by $\mathsf{of}(v)$ number of edges;
- we connect t with all $v \in F^-$ by $-\mathsf{of}(v)$ number of edges;
- if $\mathsf{of}_\tau < 0$ we choose an arbitrary set of vertices $S \in V \setminus V_\tau$ of size $|\mathsf{of}_\tau|$ and connect all $v \in S$ with s by one more edge.

See Fig. 3.

▶ Remark 25. **1.** $\mathsf{deg}(s) = \mathsf{deg}(t)$;
2. If $A \subseteq V'$ then $E(\{s\}, A) \leq \frac{\Delta}{4} + 1 + \sum_{v \in A} \mathsf{of}(v)$ and $E(\{t\}, A) = -\sum_{v \in A} \mathsf{of}(v)$.

Proof. The first property follows from the construction of τ and the second one follows from definition of G'. ◀

Let $f := \mathsf{deg}(s)$. To conclude the proof we want to show that there is an s-t flow in G' of size f (assuming that capacity of each edge is 1) and that if this flow exists, then we have an extension of τ that satisfies $\mathsf{Flow}_G \setminus \mathfrak{D}$. As we mention above together these facts imply that $(\mathsf{Flow}_G \setminus \mathfrak{D})|_\tau$ is satisfiable hence $(\mathsf{Flow}_G \setminus \mathfrak{D})|_\zeta$ is satisfiable and $(\varphi \setminus \mathfrak{C})|_\zeta$ is also satisfiable hence there is a path from v_{k_0+1} to a leaf marked by some $C \in \mathfrak{C}$ which is a contradiction with an existence of a garland and an assumption about size of the branching program.

We start with the second part. Suppose that we have a flow of size f. Fix the flow that achieves this value. We define a total proper assignment $\sigma \supseteq \tau$ in the natural way. Consider an edge $e \in E' \cup E$ and $a = (u, v), a' = (v, u)$ its directed copies. If there is a flow on the edge e:

- from u to v then $x_a = 1$ and $x_{a'} = 0$;
- from v to u then $x_a = 0$ and $x_{a'} = 1$.

otherwise we set $x_a = 0$ and $x_{a'} = 0$.

Note that $f = \mathsf{deg}(s)$ hence we use all edges that connect s with other vertices to push the flow. That implies for all $v \in V \setminus V_\tau$:

$$\sum_{\substack{e \in \mathrm{supp}(\sigma) \setminus \mathrm{supp}(\tau) \\ \mathrm{st}(e) = v}} \sigma(x_e) + \sum_{\substack{e \in \mathrm{supp}(\sigma) \setminus \mathrm{supp}(\tau) \\ \mathrm{en}(e) = v}} \sigma(x_e) = |E(s, v)| = \mathsf{of}(v)$$

and hence

$$\sum_{e \in E : \mathrm{st}(e) = v} \sigma(x_e) + \sum_{e \in E : \mathrm{en}(e) = v} \sigma(x_e) = c(v).$$

and constraints for all vertices in $V \setminus V_\tau$ are satisfied, but τ itself satisfied all constraints in $\mathsf{Flow}_G \setminus \mathfrak{D}$ that correspond to vertices in V_τ. Altogether it says that σ satisfies all constraints in $\mathsf{Flow}_G \setminus \mathfrak{D}$ as desired.

It remains to show that we have an s-t flow of size f in G'. To do it we use the Max-Flow Min-Cut Theorem and show that minimal s-t cut has size f. Consider such a cut (S, T), where S, T are disjoint subsets of V' such that $s \in S$ and $t \in T$. We consider two cases:

- either S or T is small enough, then we use the (r, β')-expansion property that we have after removing $\mathrm{supp}(\tau)$ and V_τ from G;
- S and T are large enough, then we use the Mixing Lemma to show that even removing $\mathrm{supp}(\tau)$ from G cannot destroy balanced cuts.

see Fig. 3.

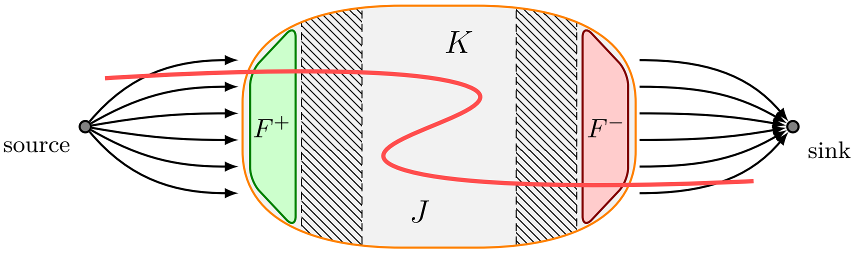


Figure 4 Graph s-t cut.

Consider an arbitrary s-t cut $S \cup T$. Let $J \coloneqq S \setminus \{s\}$ and $K \coloneqq T \setminus \{t\}$ (see Fig. 4). Consider the following cases.

1. If $J = \emptyset$ or $K = \emptyset$ then size of (S, T) cut equals $\mathsf{deg}(s)$ or $\mathsf{deg}(t)$ respectively and we are done.
2. $0 < |J| \le r$ or $0 < |K| \le r$. Wlog assume that $|J| \le r$. Note that:
$$E_{G'}(S, T) = E_{G'}(\{s\}, K) + E_{G'}(\{t\}, J) + E_{G'}(J, K).$$
$E_{G'}(\{s\}, K) = \sum_{v \in F^+ \cap K} \mathsf{of}(v)$, so by Remark 25 to give a lower bound on the size of cut it is enough to show that $E_{G'}(J, K) \ge \frac{\Delta}{4} + 1 + \sum_{v \in F^+ \cap J} \mathsf{of}(v)$. But $(V \setminus V_\tau, E \setminus \mathrm{supp}(\tau))$ satisfies (r, β')-expansion property. Hence
   - for all $v \in V \setminus V_\tau$: $|\mathsf{of}(v)| \le 0.1 \cdot \Delta$;
   - $|E_{G'}(J, K)| \ge 0.9 \cdot \Delta |J|$,

   that implies that $|E_{G'}(J, K)| - \frac{\Delta}{4} - 1 \ge 2 \sum_{v \in F^+ \cap J} \mathsf{of}(v)$.
3. $|J| > r, |K| > r$. Wlog assume that $|J| \le |K|$. By Mixing Lemma:
$$|E_G(J, \overline{J})| = \Delta|J| - E_G(J, J) \ge \Delta|J| - \frac{\Delta}{n}|J|^2 - \alpha\Delta|J| \ge \Delta|J| - |J| - \alpha\Delta|J| \ge 0.9 \cdot \Delta r,$$
and
$$|E_{G'}(J, K)| \ge |E_G(J, \overline{J})| - |\,\mathrm{supp}(\tau)| \ge 0.6 \cdot \Delta r.$$
On the other hand:
$$f = \sum_{v \in F^+} \mathsf{of}(v) =$$
$$\sum_{v \in F^+} \left(- \sum_{\substack{e \in \mathrm{supp}(\tau) \\ \mathrm{st}(e) = u}} \tau(x_e) + \sum_{\substack{e \in \mathrm{supp}(\tau) \\ \mathrm{en}(e) = u}} \tau(x_e) + c(v) \right) \le$$
$$|\,\mathrm{supp}(\tau)| \le \frac{\Delta}{4} r.$$

Hence in all cases (S, T) has size at least f which by Max-Flow Min-Cut Theorem implies the existence of flow in G' of size at least f. That as mentioned above implies the desired lower bound on the size of branching program.

5 Cook–Reckhow Proof Systems

In this section we illustrate that syntactic $(1, +k)$-BP give us a proof system in terms of Cook–Reckhow. This result is a generalization of the same result for formulas of bounded width [13]. We start with the most general definition of a proof system for a language of unsatisfiable formulas.

▶ **Definition 26** (Cook, Reckhow [7]). *A* ***proof system*** *is a polynomial-time algorithm* $\Pi(\varphi, w)$ *that satisfies two properties:*

correctness: if there is some $w \in \{0,1\}^*$ *such that* $\Pi(\varphi, w) = 1$ *then* φ *is unsatisfiable;*

soundness: if φ *is an unsatisfiable boolean formula then there is a string* $w \in \{0,1\}^*$ *such that* $\Pi(\varphi, w) = 1$.

We say that w *is a* ***witness of unsatisfiability*** *of* φ.

$(1, +k)$-BP can be used to define a natural proof system. We assume that the witness of unsatisfiability of a CNF formula φ is a description of a $(1, +k)$-BP that solves Search_φ problem; denote it by $(1, +k)$-BP-PS. This definition is equivalent to the definition of $(1, +k)$-BP-PS from [13] but we erase some technicalities.

For our purpose we need to show that there is a polynomial-time algorithm that checks whether a given description is a $(1, +k)$-BP and that it solves Search_φ.

▶ **Lemma 27.** *There is an algorithm that for given syntactic* $(1, +k)$-BP *of size* s *and boolean CNF formula* φ *with* m *clauses and* n *variables checks whether this program solves* Search_φ *in time* $\mathcal{O}\left[\left(\frac{4en}{k}\right)^k sn^2m\right]$.

We defer the proof of this Lemma to section 5.1.

▶ **Theorem 5.1** (3). *A syntactic* $(1, +k)$-BP-PS *is a proof system in terms of Cook–Reckhow definition for any constant* $k \in \mathbb{N}$.

Proof. Given a description of a branching program B we can use an algorithm from Theorem 7 to check whether it is $(1, +k)$-BP. After that we can use an algorithm from Lemma 27 to check whether this program solves Search_φ problem. If k is an absolute constant both algorithms work in time $\mathsf{poly}(|\varphi|, |B|)$. ◀

5.1 Proof of Lemma 27

Fix some syntactic $(1, +k)$-BP B and some CNF formula $\varphi := \bigvee_{i=1}^{m} C_i$. Leaves of B are marked by clauses of φ. We construct an auxiliary branching programs B_i that are obtained by replacing the labels C_i of sinks by 1 and other labels by 0.

The clause C is a solution of the Search_φ problem for an assignment z iff $C(z) = 0$. Hence B makes a mistake on the assignment z iff the path that corresponds to z ends in sink marked by C and $C(z) = 1$. But it means that B makes a mistake iff there is a variable x_i such that an assignment $x_i \leftarrow z_i$ satisfies C and there is a path in B from source to sink labeled by C consistent with an assignment $x_i \leftarrow z_i$.

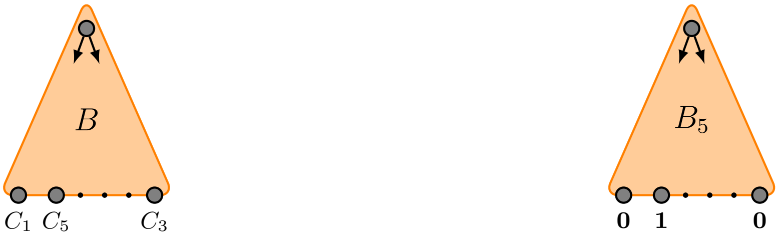


Figure 5 Construction of B_i.

The last observation gives useful criteria of correctness. We have path in B from source to sink that is labeled by C_i consistent with some assignment $x_j \leftarrow a$ iff $B_i|_{x_j \leftarrow a}$ is satisfiable. We are ready to describe an algorithm:

1. enumerate all clauses $C_i \in \varphi$;
2. enumerate variables $x_j \in C_i$ and consider a constant a such that $C_i|_{x_j \leftarrow a} = 1$;
3. check whether $B_i|_{x_j \leftarrow a}$ is satisfiable, if yes return "NO";
4. if B passes all tests then return "B is correct".

The correctness of this algorithm follows from previous observation. And we run satisfiability algorithm at most nm times hence the running time is at most $\mathcal{O}\left[\left(\frac{4en}{k}\right)^k sn^2m\right]$.

6 Open Problems

In conclusion we want to mention some open problems. We start with the obvious ones.

1. Find a formula that is hard for $(1, +k)$-BP where $k := n^{\varepsilon}$.
2. Find a formula that is hard for read-twice branching programs (programs that on any path may read each variable at most twice).

Another problems are more technical, but in our opinion the solution of these problems may lead to new techniques for proving lower bounds.

1. Find a "natural" formula that is hard for $(1, +k)$-BP for any $k > 0$. The main problem with the current bound is that we amplify our formula by an η function. This is an artificial trick that prevents generalization of our main Theorem.
2. More difficult question: can we prove a lower bound on random Δ-CNF formulas? This is a canonical example of the hard formulas. Typically, only the "local" structure is used for proving lower bounds on these formulas, which is one of the important barriers for proving lower bounds on these formulas in $\mathbf{AC}_0$-Frege proof system.

References

1 Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM J. Comput.*, 34(1):67–88, 2004. doi:10.1137/S0097539701389944.

2 Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. *Theory Comput.*, 3(1):81–102, 2007. doi:10.4086/toc.2007.v003a005.

3 Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. *Proceedings of the Steklov Institute of Mathematics*, 242:18–35, 2003. Available at http://people.cs.uchicago.edu/~razborov/files/misha.pdf. Preliminary version in *FOCS '01*.

4 Noga Alon and Fan R. K. Chung. Explicit construction of linear sized tolerant networks. *Discret. Math.*, 72(1-3):15–19, 1988. doi:10.1016/0012-365X(88)90189-6.

5 Albert Atserias, Ilario Bonacina, Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Alexander A. Razborov. Clique is hard on average for regular resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 866–877. ACM, 2018. doi:10.1145/3188745.3188856.

6 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow – resolution made simple. *J. ACM*, 48(2):149–169, 2001. doi:10.1145/375827.375835.

7 Stephen Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979. URL: https://projecteuclid.org:443/euclid.jsl/1183740343.

8 L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi:10.4153/CJM-1956-045-5.

9 Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. Optimal sherali-adams gaps from pairwise independence. In Irit Dinur, Klaus Jansen, Joseph Naor, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 125–139, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

10 Stasys Jukna. Expanders and time-restricted branching programs. *Theor. Comput. Sci.*, 409(3):471–476, 2008. doi:10.1016/j.tcs.2008.09.012.

11 Stasys Jukna. *Boolean Function Complexity — Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-24508-4.

12 Stasys Jukna and Alexander A. Razborov. Neither reading few bits twice nor reading illegally helps much. *Discret. Appl. Math.*, 85(3):223–238, 1998. doi:10.1016/S0166-218X(98)00042-0.

13 Alexander Knop. IPS-like Proof Systems Based on Binary Decision Diagrams. *Electron. Colloquium Comput. Complex.*, 24:179, 2017. URL: https://eccc.weizmann.ac.il/report/2017/179.

14 László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson. Search problems in the decision tree model. *SIAM J. Discret. Math.*, 8(1):119–132, 1995. doi:10.1137/S0895480192233867.

15 William Masek. A fast algorithm for the string editing problem and decision graph complexity. *Master Thesis, Massachusetts Institute of Technology*, 1976.

16 E. I. Nechiporuk. On a boolean function. *Dokl. Akad. Nauk SSSR*, 169:765–766, 1966.

17 Toniann Pitassi and Ran Raz. Regular resolution lower bounds for the weak pigeonhole principle. *Comb.*, 24(3):503–524, 2004. doi:10.1007/s00493-004-0030-y.

18 Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004. doi:10.1145/972639.972640.

19 Petr Savický. A probabilistic nonequivalence test for syntactic (1,+k)-branching programs. *Electron. Colloquium Comput. Complex.*, 5(51), 1998. URL: http://eccc.hpi-web.de/eccc-reports/1998/TR98-051/index.html.

20 Petr Savický and Stanislav Žák. A lower bound on branching programs reading some bits twice. *Theor. Comput. Sci.*, 172(1-2):293–301, 1997. doi:10.1016/S0304-3975(96)00183-1.

21 Detlef Sieling. New lower bounds and hierarchy results for restricted branching programs. *J. Comput. Syst. Sci.*, 53(1):79–87, 1996. doi:10.1006/jcss.1996.0050.

22 Detlef Sieling and Ingo Wegener. New lower bounds and hierarchy results for restricted branching programs. In Ernst W. Mayr, Gunther Schmidt, and Gottfried Tinhofer, editors, *Graph-Theoretic Concepts in Computer Science, 20th International Workshop, WG '94, Herrsching, Germany, June 16-18, 1994, Proceedings*, volume 903 of *Lecture Notes in Computer Science*, pages 359–370. Springer, 1994. doi:10.1007/3-540-59071-4_61.

23 Dmitry Sokolov. (semi)algebraic proofs over ±1 variables. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 78–90. ACM, 2020. doi:10.1145/3357713.3384288.

24 Ingo Wegener. On the complexity of branching programs and decision trees for clique functions. *J. ACM*, 35(2):461–471, 1988. doi:10.1145/42282.46161.

25 Ingo Wegener. *Branching Programs and Binary Decision Diagrams.* SIAM, 2000. URL: http://ls2-www.cs.uni-dortmund.de/monographs/bdd/.

26 Stanislav Žák. An exponential lower bound for real-time branching programs. *Information and Control*, 71(1):87–94, 1986. doi:10.1016/S0019-9958(86)80018-3.

A Missed Lemmas

A.1 Lemma 18

At first we prove an auxiliary Lemma.

▶ **Lemma 28.** *If $G := (V, E)$ satisfies (r, a)-expansion property, $M \subseteq E$, and $S \subseteq V$ of size at most r, such that $|E(S, \overline{S}) \setminus M| \le b\Delta|S|$ then $|S| \le \frac{|M|}{(a-b)\Delta}$.*

Proof. The size of S is at most r, hence:

$$b\Delta|S| \ge |E(S, \overline{S}) \setminus M| \ge a\Delta|S| - |M|.$$

Thus $|S| \le \frac{|M|}{(a-b)\Delta}$. ◀

▶ **Lemma A.1** (18). *Let (p, U_p, σ_p) and (q, U_q, σ_q) be 0.9-good triples. Then there is an assignment κ such that:*

- *for any γ that is an instance of $\sigma_p \uplus \sigma_q$ an assignment $\gamma \cup \kappa$ is a β'-mlce on $\mathrm{supp}(\sigma_p) \cup \mathrm{supp}(\sigma_q)$;*
- *$|\,\mathrm{supp}(\gamma \cup \kappa)| \le \nu_{k-1}\Delta r$.*

Moreover if $p = q$ then triple $(p, U_p, \sigma_p \cup \kappa)$ is β'-good.

Proof. Let $S := V_{\sigma_p} \cup V_{\sigma_q}$, $E_\sigma := \mathrm{supp}(\sigma_p) \cup \mathrm{supp}(\sigma_q)$ and $B \subseteq V \setminus S$ be the maximal set of vertices that satisfies:

- $|B| \le r$;
- $|E(B, \overline{B}) \setminus E_\sigma| \le \beta'\Delta|B|$.

At first we give an upper bound on the size of set B.

Partial assignment σ_p is 0.9-mlce on $M \cup U_p$. $\beta'\Delta|V_{\sigma_p}| \ge |E(V_{\sigma_p}, \overline{V}_{\sigma_p}) \setminus (M \cup U_p)|$ and by Lemma 28

$$|V_{\sigma_p}| \le \frac{|M \cup U_p|}{(\beta - \beta')\Delta} \le 2\frac{\nu_k}{(\beta - \beta')}r \le \frac{1}{2}\nu_{k-1}r.$$

By analogy the same holds for V_{σ_q}.

The equality $E(B, \overline{B}) \cap E_\sigma = E(B, S) \cup (E(B) \cap |M \cup U_p \cup U_q|)$ together with $|E(B, \overline{B}) \setminus E_\sigma| \le \beta'\Delta|B|$ implies:

$$(1 - \beta')\Delta|B| - |M \cup U_p \cup U_q| \le |E(B, S)|$$

By Mixing Lemma:

$$|E(B, S)| \le \frac{\Delta}{n}|B||S| + \alpha\Delta\sqrt{|S||B|}.$$

For the sake of contradiction assume that $|B| \ge |S|$ thus:

$$|E(B, S)| \le \frac{\Delta}{n}|B||S| + \alpha\Delta|B|.$$

Altogether:

$$(1-\beta')\Delta|B| \leq \frac{\Delta r}{n}\nu_{k-1}|B| + \alpha\Delta|B| + 3\nu_k\Delta r \leq 2\alpha\Delta|B|,$$

that contradicts the choice of α and β', hence $|B| \leq |S| \leq \nu_{k-1}r$.

At first we show that $(V \setminus (S \cup B), E \setminus (E_\sigma \cup E(B)))$ satisfies (r, β')-expansion property. By contradiction, suppose that there is a set $B' \subseteq V \setminus (S \cup B)$ of size at most r such that $|E(B', \overline{B'}) \setminus (E_\sigma \cup E(B))| < \beta'\Delta|B'|$.

Again by Lemma 28 we conclude that:

$$|B'| \leq \frac{|M \cup U_p \cup U_q| + \Delta|S \cup B|}{(\beta - \beta')\Delta} \leq \nu_{k-1}r + \frac{1}{2}r \leq \frac{3}{4}r.$$

But it implies that $|B \cup B'| \leq r$, moreover:

$$\begin{aligned}&|E(B \cup B', \overline{B \cup B'}) \setminus E_\sigma| \leq \\ &\qquad \beta'\Delta|B| + \beta'\Delta|B'| = \\ &\qquad\qquad \beta'\Delta|B \cup B'|. && B \text{ and } B' \text{ are disjoint}\end{aligned}$$

That contradicts the choice of B.

Now we find a proper assignment κ on the $E(B) \setminus E_\sigma$ such that for all $v \in B$:

$$\sum_{e \in E:\mathrm{st}(e)=v} x_e \geq 0.8 \cdot \Delta.$$

Since σ_p is an $(r, 0.6, 0.9)$-locally consistent assignment, then $(V \setminus V_{\sigma_p}, E \setminus \mathrm{supp}(\sigma_p))$ satisfies $(r, 0.9)$-expansion property. By analogy we have the same property for σ_q that implies: $(V \setminus S, E \setminus E_\sigma)$ satisfies $(r, 0.8)$-expansion property. Indeed, consider a set $C \subseteq V \setminus S$ of size at most r:

$$\begin{aligned}|E(C, \overline{C}) \setminus E_\sigma| &= |E(C, \overline{C})| - |E(C, \overline{C}) \cap E_\sigma| \\ &\geq |E(C, \overline{C})| - |E(C, \overline{C}) \cap \mathrm{supp}(\sigma_p)| - |E(C, \overline{C}) \cap \mathrm{supp}(\sigma_q)| \\ &= |E(C, \overline{C}) \setminus \mathrm{supp}(\sigma_p)| - 0.1 \cdot \Delta|C| \\ &\geq 0.8 \cdot \Delta|C|.\end{aligned}$$

By Proposition 12 there is an enumeration of vertices in B: $v_1, v_2, \ldots, v_{|B|} \in B$ and a sequence $R_1, \ldots, R_{|B|} \subseteq (E(B) \setminus E_\sigma)$ such that:

- $R_i = E(\{v_i\}, V \setminus \{v_1, v_2, \ldots, v_i\}) \setminus E_\sigma$;
- $|R_i| \geq 0.8\Delta$.

We define κ in the following way:

- for an $e \in R_i$ we assign corresponding variables to direct the flow outside of the vertex v_i (i.e. if e' is a directed copy of e that goes outside of v_i we set $x_{e'}$ to 1 and set the dual edge to 0);
- for all loops inside the set B we assign corresponding variables to 0.

Let γ be an instance of $\sigma_p \uplus \sigma_q$, $\zeta := \gamma \cup \kappa$ and $V_\zeta := S \cup B$. We have already shown that the graph $(V \setminus V_\zeta, E \setminus \mathrm{supp}(\zeta))$ satisfies (r, β')-expansion property. We want to show that vertices in V_ζ are 0.6-satisfied by ζ. Consider four cases.

1. $v \in V_\rho$. Both assignments σ_p and σ_q extend an assignment ρ hence γ agreed with both assignments on edges incident to V_ρ. Thus γ 0.6-satisfies v.

2. $v \in V_{\sigma_p} \setminus V_\rho$. Let E_v be a set of edges that are incident to v. At least $0.8 \cdot \Delta$ of those edges carry outgoing flow from v in σ_p. Denote those edges as E_{σ_p}.
   If $v \notin V_{\sigma_q}$ then σ_q may assign at most $0.1 \cdot \Delta$ edges in E_v. That means that in γ at least $0.7 \cdot \Delta$ edges from E_{σ_p} still carry outgoing flow from v.
   If $v \in V_{\sigma_q}$ then σ_p and σ_q both 0.8-satisfy v. Let $E_{\sigma_q} \subseteq E_v$ be the set of edges that carry outgoing flow from v in σ_q. Then $E_{\sigma_p} \cap E_{\sigma_q} \geq 0.6 \cdot \Delta$, and all those edges carry outgoing flow from v in γ.
   Note that if $\sigma_p = \sigma_q$, then we 0.8-satisfy v.
3. $v \in V_{\sigma_p} \setminus V_\rho$. By analogy with the previous case.
4. $v \in B$. We direct the flow on at least $0.8 \cdot \Delta$ edges from E_v outside of v hence κ 0.8-satisfies v.

By construction $V_\zeta := V_{\sigma_p} \cup V_{\sigma_q} \cup B$ hence $|V_\zeta| \leq \nu_{k-1} r$ and $|\operatorname{supp}(\zeta)| \leq \nu_{k-1} r$. In order to check that ζ is β'-mlce note that:

$$|E(B, \overline{B}) \setminus E_\sigma| \leq \beta' \Delta |B| \leq \beta' \Delta |V_\zeta|,$$

but

$$|E(B, \overline{B}) \setminus E_\sigma| = |E(V_\zeta, \overline{V_\zeta}) \setminus E_\sigma|$$

since σ_p and σ_q together assign all edges that are incident to $V_{\sigma_p} \cup V_{\sigma_q}$. Thus:

$$|E(V_\zeta, \overline{V_\zeta}) \setminus E_\sigma| \leq \beta' \Delta |V_\zeta|$$

that concludes the proof.

In case of $(p, U_p, \sigma_p) = (q, U_q, \sigma_q)$ it remains to show that ζ is β'-mlce on $M \cup U_p$. Again we note that:

$$|E(B, \overline{B}) \setminus E_\sigma| \leq \beta' \Delta |B|,$$

and also

$$|E(V_{\sigma_p}, \overline{V}_{\sigma_p}) \setminus E_\sigma| \leq \beta' \Delta |V_{\sigma_p}|,$$

hence

$$|E(V_{\sigma_p} \cup B, \overline{V_{\sigma_p} \cup B}) \setminus E_\sigma| \leq \beta' \Delta (|V_{\sigma_p}| + |B|) \leq \beta' \Delta |V_{\sigma_p} \cup B|,$$

where the last inequality holds since B and V_{σ_p} are disjoint, that concludes the proof. ◀

A.2 Lemma 23

▶ **Lemma A.2** (23). *For all $i \leq \ell$:*

- *κ_i exists;*
- *$|V_{\tau_i}| \leq \frac{1}{(\beta-\beta')\Delta}(\operatorname{supp}(\zeta) + \Delta|U_i|)$ and hence $|\tau_i| \leq \frac{2}{(\beta-\beta')}(|\operatorname{supp}(\zeta)| + \Delta|U_i|)$;*
- *$(V \setminus V_{\tau_i}, E \setminus \operatorname{supp}(\tau_i))$ satisfies (r, β')-expansion property.*

Proof. We show by induction on i that:

- $(V \setminus V_{\tau_i}, E \setminus \operatorname{supp}(\tau_i))$ satisfies (r, β')-expansion property;
- $|E(V_{\tau_i}, \overline{V}_{\tau_i}) \setminus (\operatorname{supp}(\zeta) \cup E(U_i))| < \beta' \Delta |V_{\tau_i}|$;
- $|V_{\tau_i}| \leq \frac{1}{(\beta-\beta')\Delta}(\operatorname{supp}(\zeta) + \Delta|U_i|)$ and hence $|\tau_i| \leq \frac{2}{(\beta-\beta')}(|\operatorname{supp}(\zeta)| + \Delta|U_i|)$.

Assignment τ_0 is ζ and ζ is $(r, 0.6, \beta')$-locally consistent, in particular, $(V \setminus V_\zeta, E \setminus \operatorname{supp}(\zeta))$ satisfies (r, β')-expansion property and $E(V_\zeta, \overline{V}_\zeta) \setminus \operatorname{supp}(\zeta)) = \emptyset$.

By definition of H_i:

$$\beta'\Delta|H_i| > |E(H_i, \overline{H}_i \setminus \{u_i\}) \setminus \operatorname{supp}(\tau_i)| \ge |E(H_i, \overline{H}_i) \setminus (\operatorname{supp}(\tau_i) \cup E(H_i, \{u_i\}))|$$

and by Lemma 28

$$|H_i| \le \frac{|\operatorname{supp}(\tau_i) \cup E(H_i, u_i)|}{(\beta - \beta')\Delta} \le \frac{|\operatorname{supp}(\tau_i) \cup E(H_i, u_i)|}{(\beta - \beta')\Delta} \le \frac{1}{2}\nu_{k-2}(r+1).$$

Hence $|H_i \cup V_{\tau_i} \cup \{u_i\}| \le r$ that together with:

$$\begin{aligned}
&|E(H_i \cup V_{\tau_i} \cup \{u_i\}, \overline{H_i \cup V_{\tau_i} \cup \{u_i\}}) \setminus (\operatorname{supp}(\zeta) \cup E(U_{i+1}))| \le \\
&|E(V_{\tau_i}, \overline{H_i \cup V_{\tau_i}}) \setminus (\operatorname{supp}(\zeta) \cup E(U_{i+1}))| + |E(H_i, \overline{H}_i) \setminus (\operatorname{supp}(\zeta) \cup E(U_{i+1}) \cup E(V_{\tau_i})| \le \\
&\beta'\Delta|V_{\tau_i}| + \beta'\Delta|H_i| \le \\
&\beta'\Delta|H_i \cup V_{\tau_i}| \le \\
&\beta'\Delta|H_i \cup V_{\tau_i} \cup \{u_i\}|
\end{aligned}$$

implies $|V_{\tau_{i+1}}| = |H_i \cup V_{\tau_i} \cup \{u_i\}| \le \frac{1}{(\beta-\beta')\Delta}(|\operatorname{supp}(\zeta)| + \Delta|U_{i+1}|)$ by Lemma 28. Also $|\tau_{i+1}| \le \frac{2}{(\beta-\beta')}(|\operatorname{supp}(\zeta)| + \Delta|U_{i+1}|)$ since by construction τ_{i+1} assigns only edges in $\operatorname{supp}(\zeta) \cup E(U_i \cup V_{\tau_{i+1}})$.

Now we show that a graph $(V \setminus V_{\tau_{i+1}}, E \setminus \operatorname{supp}(\tau_{i+1}))$ satisfies (r, β')-expansion property. For the sake of contradiction assume that there is a set $S \subseteq V \setminus V_{\tau_{i+1}}$ of size at most r such that: $E(S, \overline{S}) \setminus \operatorname{supp}(\tau_{i+1}) \le \beta'\Delta|B|$.

By Lemma 28 $|S| \le \frac{|\operatorname{supp}(\tau_{i+1})|}{(\beta-\beta')\Delta} \le \frac{1}{2}\nu_{k-2}(r+1)$. Hence $|H_i \cup S| \le r$ that together with:

$$\begin{aligned}
&E(H_i \cup S, \overline{H_i \cup S} \setminus \{u_i\}) \setminus \operatorname{supp}(\tau_i)| \le \\
&E(H_i, \overline{H_i \cup S} \setminus \{u_i\}) \setminus \operatorname{supp}(\tau_i)| + E(S, \overline{H_i \cup S} \setminus \{u_i\}) \setminus \operatorname{supp}(\tau_i)| \le \\
&\beta'\Delta|H_i| + \beta'\Delta|S| = \\
&\beta'\Delta|H_i \cup S|
\end{aligned}$$

contradicts the choice of H_i.

To conclude the proof we have to show the existence of κ_i. Note that $(V \setminus V_{\tau_i}, E \setminus \operatorname{supp}(\tau_i))$ satisfies (r, β')-expansion property. Consider an arbitrary set $B \subseteq V \setminus (V_{\tau_i} \cup \{u_i\})$ of size at most r:

$$|E(B, \overline{B}) \setminus (\operatorname{supp}(\tau_i) \cup E(\{u_i\}))| \ge \beta'\Delta|B| - E(B, \{u\}).$$

By Mixing Lemma:

$$|E(B, \{u\})| \le \frac{\Delta}{n}|B| + \alpha\Delta\sqrt{B} \le 0.05 \cdot \Delta|B|,$$

and hence

$$|E(B, \overline{B}) \setminus (\operatorname{supp}(\tau_i) \cup E(\{u_i\}))| \ge 0.9 \cdot \Delta|B|$$

and graph $(V \setminus V_{\tau_i} \setminus \{u_i\}, E \setminus \operatorname{supp}(\tau_i))$ satisfies $(r, 0.9)$-expansion property.

By Proposition 12 there is an enumeration of vertices in H_i: $v_1, v_2, \ldots, v_{|H_i|} \in H_i$ and a sequence $R_1, \ldots, R_{|H_i|} \subseteq E(H_i) \setminus (\operatorname{supp}(\tau_i) \cup E(\{u_i\}))$ such that:

- $R_k = E(\{v_k\}, V \setminus \{v_1, v_2, \ldots, v_k\}) \setminus (\mathrm{supp}(\tau_i) \cup E(\{u_i\}))$;
- $|R_i| \geq 0.9 \cdot \Delta$.

We define κ_i for vertices $v_1, \ldots, v_{H_i}$ step by step, such that κ_i on $E(v_k)$ satisfies the constraint:

$$\sum_{e \in E:\mathrm{st}(e)=v_k} (\tau' \cup \kappa_i)(x_e) - \sum_{e \in E:\mathrm{en}(e)=v_k} (\tau' \cup \kappa_i)(x_e) = c(v_k).$$

Since we have an access to the $0.9 \cdot \Delta$ edges and others are already assigned, we can always choose the right values (loops are always assigned to zero). ◀

B Garland in the Paths

▶ **Lemma B.1** (19). *There are $(p, U_p, \sigma_p), (q, U_q, \sigma_q) \in \mathcal{S}$ such that (p, q) forms a $(k+1)$-garland.*

Proof. Note that we can describe elements in $\mathcal{P}$ by a sequence of bits of size $s := \nu_k \Delta r$. Each bit of this sequence describes an assignment for an edge e that we choose on "branching step". From the construction it follows that different sequences generate different paths in the branching program and hence different elements of $\mathcal{P}$.

Let $s_k := \lfloor \frac{s}{k+1} \rfloor$. We construct our garland by the iterative algorithm. After i-th iteration we have a set S_i of sequences of size is_k such that any two of the corresponding paths form i-garland and all paths end in the same node. The size of S_i will be at least $\exp\left[s_k - \frac{i}{2k} s_k\right]$ for all $1 \leq i \leq k+1$.

1. For $i = 1$ consider all possible strings of length s_k and paths that correspond to them. The branching program has size at most $2^{\frac{s_k}{2k}}$, hence there exists a node such that at least $2^{\frac{s_k(2k-1)}{2k}}$ paths end there. The set S_1 consists of all corresponding sequences.
2. For the step i, $2 \leq i \leq k+1$, we consider all sequences in S_{i-1}. Let v be the end node of all paths corresponding to sequences in the set. To each sequence $s \in S_{i-1}$ we append a string u_s of s_k bits in such a way that for any pair $r, r' \in S_{i-1}$ paths that corresponds to ru_r and $r'u_{r'}$ differ at some node after v. Since $2^{s_k} \geq |S_{i-1}|$, it is possible to do this. For the resulting sequences, we consider the set of the corresponding paths. The set of paths has size at least $2^{\frac{s_k(2k-i+1)}{2k}}$, and the size of the program is at most $2^{\frac{s_k}{2k}}$. Hence there exists a node such that $2^{\frac{s_k(2k-i)}{2k}}$ paths end there. Let S_i be the set of sequences corresponding to those paths.

After $k+1$ steps we have a set S_{k+1}, $|S_{k+1}| \geq 2$, such that any two sequences in it correspond to a $(k+1)$-garland. ◀

A Majority Lemma for Randomised Query Complexity

Mika Göös ✉
School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland

Gilbert Maystre ✉
School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland

Abstract

We show that computing the majority of n copies of a boolean function g has randomised query complexity $\mathrm{R}(\textsc{Maj} \circ g^n) = \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g))$. In fact, we show that to obtain a similar result for any composed function $f \circ g^n$, it suffices to prove a sufficiently strong form of the result only in the special case $g = \textsc{GapOr}$.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Query Complexity, Composition, Majority

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.18

Acknowledgements We thank Thomas Watson and anonymous reviewers for helpful comments.

1 Introduction

In boolean function complexity theory, a typical *direct sum problem* asks: For a given boolean function $g\colon \{0,1\}^m \to \{0,1\}$, how much harder is it to compute g on n separate inputs, that is, computing $g^n(x^1, \dots, x^n) \coloneqq (g(x^1), \dots, g(x^n))$, compared to computing g on a single input? For randomised query complexity, a complete answer was recently obtained by Blais and Brody [7] (improving on [17, 6]). They showed that the most obvious way to compute g^n is optimal: Evaluate each copy of g separately with a "reduced" error probability $\ll 1/n$ so that, by a union bound, the n-bit output will be correct with high probability. More precisely, their result states (we assume $n \geq 3$ for simplicity of notation throughout the paper)

$$\forall g : \quad \mathrm{R}(g^n) \;=\; \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g)). \qquad \text{(Direct sum)}$$

Here we used standard notation: $\mathrm{R}(g) \coloneqq \mathrm{R}_{1/3}(g)$ where $\mathrm{R}_\epsilon(g)$ denotes the ϵ-error query complexity of g, that is, the least number of queries a randomised algorithm (decision tree) must make to the input bits $x_i \in \{0,1\}$ of an unknown input $x \in \{0,1\}^m$ in order to output $g(x)$ with probability at least $1 - \epsilon$ (where the probability is over the internal randomness of the algorithm). Similarly, $\overline{\mathrm{R}}_\epsilon(g)$ denotes the ϵ-error *expected* query complexity of g where we measure the expected (rather than worst-case) number of queries made by the algorithm. See Section 2 for precise definitions.

How far can we push the direct sum result? What if, instead of all the n output bits of g^n, we only wanted to compute their parity? In other words, what is the randomised query complexity of the composed function $\textsc{Xor} \circ g^n$? Do we still have to compute each g with reduced error? Brody et al. [8] provided an affirmative answer:

$$\forall g : \quad \mathrm{R}(\textsc{Xor} \circ g^n) \;=\; \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g)). \qquad (\textsc{Xor}\ \text{Lemma})$$

More generally, we can ask the following question.

▶ **Problem 1.** *For which n-bit outer functions f (assume $\mathrm{R}(f) = \Theta(n)$ for simplicity) and inner functions g does the composed function $f \circ g^n$ necessitate error reduction?*

There is no conjectured characterisation for when error reduction is necessary. To showcase the subtlety of this question, we mention that $f = \textsc{Or}$, despite having a highly "sensitive" input $x = 0^n$, never necessitates error reduction. By now, there are many proofs [12, 20, 22, 15, 5] showing that $\mathrm{R}(\textsc{Or} \circ g^n) = O(n \cdot \mathrm{R}(g))$ for every g.

Our goal in this paper is to make further progress on Problem 1.

1.1 Our results

Our main result is to prove tight bounds for composing with the n-bit majority function $\textsc{Maj}$. This in particular confirms a conjecture made in [7, 5].

▶ **Theorem 2** ($\textsc{Maj}$ lemma). $\mathrm{R}(\textsc{Maj} \circ g^n) = \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g))$ *for every partial function* g.

Previously, Ben-David et al. [5] proved Theorem 2 in the special case $g = \textsc{GapOr}$. Here $\textsc{GapOr} = \textsc{GapOr}_m$ is the m-bit partial function defined by $\textsc{GapOr}(x) = \textsc{Or}(x)$ on inputs of Hamming weight $|x| \in \{0, m/2\}$ and is undefined otherwise. This is a particularly clean example of a function whose query complexity behaves as (assuming $m \geq \log(1/\epsilon)$)

$$\overline{\mathrm{R}}_\epsilon(\textsc{GapOr}) \;=\; \Theta(\log(1/\epsilon)).$$

We prove Theorem 2 by a direct *reduction* to this previous result! Our more general result says, informally, that error reduction is necessary for any composed function $f \circ g^n$ if it is necessary in the special case $g = \textsc{GapOr}$. Our key conceptual insight is to formulate a sense in which every g can be "simulated" by $\textsc{GapOr}$. There is, however, a slight technical caveat. For the reduction to work, we need to assume that the lower bound for $f \circ \textsc{GapOr}^n$ holds not only against randomised decision trees but also against a more powerful model called *ϵ-approximate nonnegative degree* $\deg^+_\epsilon$ (aka conical junta degree, partition bound), which we will recall in Section 2.

▶ **Theorem 3** (Reduction to $\textsc{GapOr}$). *If a function f satisfies* $\deg^+_\epsilon(f \circ h^n) \geq \Omega(n \log n)$ *for some constant* $\epsilon > 0$ *and for both* $h \in \{\textsc{GapOr}_{\log n}, \neg\textsc{GapOr}_{\log n}\}$, *then*

$$\forall g: \quad \mathrm{R}(f \circ g^n) \;=\; \Omega(n \cdot \overline{\mathrm{R}}_{1/n}(g)).$$

Theorem 2 follows immediately by combining Theorem 3 with [5, Theorem 4], which proved the required nonnegative degree lower bound for $\textsc{Maj} \circ \textsc{GapOr}^n$ (we only note that their proof works equally well for $\neg\textsc{GapOr}$ in place of $\textsc{GapOr}$). In fact, the nonnegative degree lower bound holds more generally for any $(2n+1)$-bit outer function that agrees with $\textsc{Maj}$ on inputs of weight n and $n+1$. For example, $\textsc{Xor}$ is such a function, and hence the $\textsc{Xor}$ lemma of Brody et al. [8] can be recovered using Theorem 3. However, the original proof in [8] is much simpler than ours, and moreover, the result of [8] actually characterises $\overline{\mathrm{R}}_\epsilon(\textsc{Xor} \circ g^n)$ for all $\epsilon > 0$ while we focus on the bounded-error case $\epsilon = 1/3$.

Our goal for the rest of the paper is to prove Theorem 3.

Optimality? We note that our choice of $\textsc{GapOr}$ in Theorem 3 is optimal at least in the sense that it cannot be replaced with the more symmetric alternative $\textsc{GapMaj}$, which is defined by $\textsc{GapMaj}_m(x) = \textsc{Maj}_m(x)$ on inputs of weight $|x| \in \{m/3, 2m/3\}$ and undefined otherwise. There are known examples of *partial* f (but no known *total* ones) for which $\textsc{GapOr}$ does not need error reduction while $\textsc{GapMaj}$ does [5, Section 4]. We suspect however that other aspects of Theorem 3 can be improved; see Subsection 1.4 for open problems.

1.2 Techniques: Leaf Lemma

Our main technical contribution, which might be of independent interest, is what we call Leaf Lemma. It states that every boolean function g admits a balanced input distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$, where μ^i is a distribution supported on $g^{-1}(i)$, and a "hard side" $b \in \{0,1\}$ satisfying the following: If we run a decision tree of shallow depth $\ll \overline{\mathrm{R}}_\epsilon(g)$ on a random input $x \sim \mu$ then we will typically reach a leaf ℓ making *one-sided error*, that is, if the leaf ℓ is reached by $x \sim \mu^b$ with probability p, then ℓ is also reached by $x \sim \mu^{1-b}$ with probability at least $\epsilon \cdot p$. Interestingly, this property is inherently one-sided and the choice of the hard side b depends on the function g. For example, GapOr and ¬GapOr have distinct hard sides. See our proof overview in Section 3 for more details.

1.3 Other related work

Complexity of composition. A major theme in boolean function complexity theory is to understand the complexity of the composition $f \circ g^n$ in terms of the complexities of its two constituent functions. It has been long known that many well-studied complexity measures behave *multiplicatively* under composition. For example, deterministic query complexity satisfies $\mathrm{D}(f \circ g^n) = \mathrm{D}(f)\,\mathrm{D}(g)$ [24], quantum query complexity satisfies $\mathrm{Q}(f \circ g^n) = \Theta(\mathrm{Q}(f)\,\mathrm{Q}(g))$ [23, 21], and yet more examples (degree, certificate complexity, sensitivity) are discussed in [25]. An interesting exception to this rule is randomised query complexity, where we can have two types of counter-examples.

- *Super-multiplicative:* There are functions f and g such that $\mathrm{R}(f \circ g^n) \geq \omega(\mathrm{R}(f)\,\mathrm{R}(g))$. For example, this happens whenever f necessitates error reduction for $g =$ GapOr.
- *Sub-multiplicative:* Recent work [13, 3] has found surprising examples of *partial* f and g such that $\mathrm{R}(f \circ g^n) \leq o(\mathrm{R}(f)\,\mathrm{R}(g))$.

It is still open to quantify the extent to which multiplicativity can fail. For example, it has not been ruled out that $\mathrm{R}(f \circ g^n) \geq \mathrm{R}(f)\,\mathrm{R}(g)/\mathrm{poly}(\log n)$ for all partial functions. It is also possible that a strict multiplicative lower bound holds for all *total* functions. This latter question is known as the *randomised composition conjecture* (for total functions) and it has been studied in a long line of work [6, 1, 13, 2, 3, 4].

Noisy decision trees. Necessity of error reduction is closely related to the model of "noisy decision trees" [12, 11, 10, 15]. In this model, the goal is to compute a boolean function f given *noisy query access* to its input bits. A single query to an input variable x_i returns its correct value with probability $2/3$ (say) and the opposite value $1 - x_i$ with probability $1/3$. This model is effectively equivalent to computing $f \circ \textsc{GapMaj}^n$ in the standard query model. With this interpretation, one of the results of [12] states that $\mathrm{R}(\textsc{Maj} \circ \textsc{GapMaj}^n) = \Theta(n \log n)$. We note that this is weaker (in two respects) than the result $\deg^+_\epsilon(\textsc{Maj} \circ \textsc{GapOr}^n) = \Theta(n \log n)$ from [5], which we used to derive our main result (although see Problem 4 below).

1.4 Open problems

How optimal is Theorem 3? We suspect that our assumption about nonnegative degree is an artifact of our proof and can be relaxed as follows.

▶ **Problem 4.** *Show that the hypothesis in Theorem 3 can be weakened to* $\mathrm{R}(f \circ h^n) \geq \Omega(n \log n)$.

Whether we need to assume hardness for both GapOr and its negation, we do not know.

▶ **Problem 5.** *Are there examples of f with $\mathrm{R}(f \circ \textsc{GapOr}^n) \geq \omega(\mathrm{R}(f \circ \neg\textsc{GapOr}^n))$?*

Theorem 3 could be useful in showing tight composition results for yet more outer functions. For example, consider the well-studied partial function $\textsc{SqrtGapMaj}_n$ (often called simply *the* gap majority function) defined as $\textsc{Maj}_n$ but restricted to inputs of Hamming weight $|x| \notin n/2 \pm \sqrt{n}$.

▶ **Problem 6.** *Show $\mathrm{R}(\textsc{SqrtGapMaj} \circ g^n) = \Theta(n \cdot \overline{\mathrm{R}}_{1/n}(g))$ for every g.*

2 Query complexity basics

We study *partial* boolean functions $f\colon \{0,1\}^n \to \{0,1,*\}$. The *domain* of the function is $\mathrm{dom}(f) \coloneqq f^{-1}(\{0,1\})$ and the inputs $f^{-1}(*)$ are *undefined*. We say f is *total* if $\mathrm{dom}(f) = \{0,1\}^n$. For partial functions f and g, their *composition* $f \circ g^n$ is defined by $(f \circ g^n)(x^1,\ldots,x^n) \coloneqq f(g(x^1),\ldots,g(x^n))$ if $x^i \in \mathrm{dom}(g)$ for all $i \in [n]$; otherwise $(f \circ g^n)(x^1,\ldots,x^n) \coloneqq *$. Standard references for boolean function complexity are [9, 18].

Decision trees. A *(deterministic) decision tree* t is an algorithm for computing a boolean function on an unknown input $x \in \{0,1\}^n$. The algorithm repeatedly queries the input variables $x_i \in \{0,1\}$ in some order (which can depend on outcomes of queries made so far) until eventually producing an output $t(x)$. Such an algorithm can be represented as a binary tree, with internal nodes labelled with variables x_i, outgoing edges of the internal nodes labelled with query outcomes ($x_i = 0$ and $x_i = 1$), and leaves labelled with output values. Each input x determines a unique root-to-leaf path, obtained by following the query outcomes consistent with x. The most important cost measure of t is its *depth*, denoted $\mathrm{depth}(t)$, which is the longest root-to-leaf path in the tree and equals $\max_x q(t,x)$ where $q(t,x)$ denotes the number of queries made by t on input x.

A *randomised decision tree* T is a distribution over deterministic decision trees $t \sim T$. We say T computes $f\colon \{0,1\}^n \to \{0,1,*\}$ with error ϵ if for every $x \in \mathrm{dom}(f)$ we have $\mathbb{P}_{t\sim T}[t(x) = f(x)] \geq 1-\epsilon$. There are two cost measures for T: the *(worst-case) depth* is the maximum depth of any decision tree in the support of T; the *expected depth* is $\max_x \mathbb{E}_{t\sim T}[q(t,x)]$. The *$\epsilon$-error query complexity of f*, denoted $\mathrm{R}_\epsilon(f)$, is the least depth of a randomised decision tree that computes f with error ϵ. The *ϵ-error expected query complexity*, denoted $\overline{\mathrm{R}}_\epsilon(f)$, is defined analogously.

Error reduction. It is well known that the error probability of an algorithm (computing a boolean-valued function) can be reduced from any constant $1/2-\delta$, where $\delta > 0$, to any other constant $\epsilon > 0$ by repeating the algorithm constantly many times (in fact, $O(\log(1/\epsilon)/\delta^2)$ many) and outputting the majority answer. Hence we often set $\epsilon \coloneqq 1/3$ and omit ϵ from notation. In this bounded-error regime, we have $\overline{\mathrm{R}}(f) \leq \mathrm{R}(f) \leq O(\overline{\mathrm{R}}(f))$ where the second inequality follows by truncating executions that query many more bits than the expectation. For vanishing $\epsilon = o(1)$ (as $n \to \infty$), it is possible that $\overline{\mathrm{R}}_\epsilon(f) \leq o(\mathrm{R}_\epsilon(f))$. For example, consider the partial $2n$-bit function f where the task is to distinguish inputs of the form $x0^n$ from inputs of the form $0^n x$ with the promise that $|x| = n/2$. We have $\overline{\mathrm{R}}_{1/n}(f) = O(1)$ while $\mathrm{R}_{1/n}(f) = \Theta(\log n)$. In this small-error regime, the following fine-grained error reduction calculation will be useful.

▷ Claim 7. $\overline{\mathrm{R}}_{\epsilon^k}(f) \leq 4k \cdot \overline{\mathrm{R}}_\epsilon(f)$ for every $k \geq 1$ and $\epsilon \leq 1/16$.

Proof. Suppose T computes f with error ϵ and consider the algorithm T' that runs T $4k-1$ times and outputs the majority answer. Then T' errs iff at least $2k$ of the runs err. This happens with probability at most $\sum_{i=2k}^{4k-1}\binom{4k-1}{i}\epsilon^i(1-\epsilon)^{4k-1-i} \leq 2^{4k}\epsilon^{2k} \leq \epsilon^k$. ◁

Leaf indicators. Let t be a decision tree with n-bit inputs. We denote by $\mathcal{L}(t)$ the set of its leaves and by $\ell^t_x \in \mathcal{L}(t)$ the unique leaf reached on input x. We often identify a leaf $\ell \in \mathcal{L}(t)$ with its associated *leaf indicator* function $\ell\colon \{0,1\}^n \to \{0,1\}$ defined by $\ell(x) := 1$ iff input x reaches leaf ℓ. Thus each ℓ is simply a conjunction of at most $\operatorname{depth}(t)$ literals (x_i or $\bar{x}_i$) determined by the unique root-to-ℓ path in t. If t outputs boolean values, we let $\mathcal{A}(t) \subseteq \mathcal{L}(t)$ denote the set of *accepting* leaves, that is, those that output 1. Since the leaf indicators have pairwise disjoint supports, we can write the function computed by t as

$$t(x) \;=\; \textstyle\sum_{\ell\in\mathcal{A}(t)} \ell(x). \tag{1}$$

Nonnegative degree. Let $p\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$ be a nonnegative function. We say p is a *nonnegative d-junta* if it depends on at most d of its variables. For example, if t is a depth-d decision tree, then each $\ell \in \mathcal{L}(t)$ is a nonnegative d-junta. More generally, we say that p is a *conical junta* of degree d if it can be written as a conical combination of nonnegative d-juntas, that is, $p(x) = \sum_i a_i q_i(x)$ where $a_i \geq 0$ are nonnegative scalars and the q_i are nonnegative d-juntas. For example, the function computed by t is a degree-d conical junta, as given by the expression (1). The *nonnegative degree* of p, denoted $\deg^+(p)$, is the least d such that p is a degree-d conical junta.

Let $f\colon \{0,1\}^n \to \{0,1,*\}$ be a partial function. We say that p ϵ-approximates f if $p(x) \in f(x) \pm \epsilon$ for every $x \in \operatorname{dom}(f)$. The *$\epsilon$-approximate nonnegative degree* of f, denoted $\deg^+_\epsilon(f)$, is the least degree of a conical junta that ϵ-approximates f. For example, if T is a depth-d randomised ϵ-error decision tree for f, then there exists a degree-d conical junta p_T that ϵ-approximates f, namely,

$$p_T(x) \;:=\; \mathbb{E}_{t\sim T}[t(x)] \;\in\; f(x) \pm \epsilon.$$

This shows that $\deg^+_\epsilon(f) \leq \mathrm{R}_\epsilon(f)$. The gap betweeen $\deg^+_{1/3}(f)$ and $\mathrm{R}(f)$ can be huge for partial functions. For example, consider the n-bit UniqueOr defined by $\textsc{UniqueOr}(x) = \textsc{Or}(x)$ for inputs of weight $|x| \in \{0,1\}$ and undefined othtwerwise. Then $\deg^+(\textsc{UniqueOr}) = 1$ (computed by $\sum_i x_i$) while $\mathrm{R}(\textsc{UniqueOr}) = \Theta(n)$. For total functions, the gap is at most polynomial [9].

Nonnegative degree has been studied under many names: (one-sided) partition bound [16], WAPP query complexity [14, 5], and query complexity "in expectation" [19].

3 Proof overview

Here we outline the proof of Theorem 3. We phrase the proof in the contrapositive: Supposing that T is a randomised decision tree computing $f \circ g^n$ of shallow depth $\ll n \cdot \overline{\mathrm{R}}_{1/n}(g)$ we construct an approximate conical junta for $f \circ \textsc{GapOr}^n$ (or $f \circ \neg\textsc{GapOr}^n$) of degree $\ll n \log n$.

Our overview is in two parts.

(§3.1) We first formulate our main technical lemma called Leaf Lemma and its generalisation Multileaf Lemma. They describe what typical leaves of T look like: they are *noisy*, meaning that they make noticeable errors in predicting the outputs of many copies of g. The proofs of these lemmas will occupy the remaining sections of this paper.

(§3.2) Then we use Multileaf Lemma to prove Theorem 3. A notable component of this part of the proof is showing how the acceptance probabilities of noisy leaves can be "simulated" by low-degree conical juntas in the domain of $f \circ \textsc{GapOr}^n$.

3.1 Statement of Leaf Lemma

Example. We build up to the statement of Leaf Lemma by first considering the prototypical example $g = \textsc{GapOr}_m$. Define two distributions μ^0 and μ^1 so that μ^i is uniform over $\textsc{GapOr}_m^{-1}(i)$. Namely, μ^0 places probability 1 on the input 0^m and μ^1 is uniform over x of weight $|x| = m/2$. Suppose t is a deterministic decision tree of shallow depth $d \ll m$ trying to compute $\textsc{GapOr}_m$. For a leaf $\ell \in \mathcal{L}(t)$ and any input distribution μ we write for short

$$\ell(\mu) \;\coloneqq\; \mathbb{E}_{x\sim\mu}[\ell(x)] \;=\; \mathbb{P}_{x\sim\mu}[\ell(x)=1].$$

What do the typical leaves look like when we run t on a random input $x \sim \mu^i$ for $i \in \{0,1\}$?

- *Easy side $i = 1$.* The tree will query a 1-bit after about 2 queries in expectation. Such leaves ℓ are safe to output 1 as they know $\textsc{GapOr}(x) = 1$ for certain: $\ell(\mu^0) = 0$ and $\ell(\mu^1) > 0$.
- *Hard side $i = 0$.* Here every query returns 0 and we reach a leaf ℓ reading d many 0s. Although the leaf ℓ can be quite confident that the input x was sampled from μ^0 rather than μ^1, some uncertainty remains: $\ell(\mu^0) = 1$ and $\ell(\mu^1) \geq \epsilon$ for $\epsilon \coloneqq 2^{-\Omega(d)}$.

In both cases, we have $\ell(\mu^1) \geq \epsilon \cdot \ell(\mu^0)$ and we say that ℓ is *(one-sidedly) noisy.* We now formalise how every g gives rise to such noisy leaves.

General case. Fix a partial function $g\colon \{0,1\}^m \to \{0,1,*\}$. Let $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ be a balanced distribution where μ^i is supported on $g^{-1}(i)$. For a leaf ℓ over m bits, a "hard side" $b \in \{0,1\}$, and an error parameter $\epsilon \geq 0$, we define

$$\ell \text{ is } (\epsilon,\mu,b)\text{-}\textit{noisy} \quad \overset{\text{def}}{\iff} \quad \ell(\mu^{1-b}) \;\geq\; \epsilon \cdot \ell(\mu^b).$$

Our Leaf Lemma says that every partial function g admits a hard distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ such that if we run a shallow decision tree t on a random input $x \sim \mu$, the leaf reached ℓ_x^t will typically be noisy. For simplicity of notation, for small quantities $a, b \in [0,1]$, we write $a \ll b$ (resp. $a \lll b$) to mean $a \leq cb$ (resp. $a^c \leq b$) for a sufficiently small constant $c > 0$.

▶ **Leaf Lemma.** *For every partial g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over* $\mathrm{dom}(g)$ *and a hard side $b \in \{0,1\}$ such that for every deterministic tree t and $i \in \{0,1\}$:*

$$\frac{\mathbb{E}_{x\sim\mu^i}[q(t,x)]}{\overline{\mathrm{R}}_\epsilon(g)} \;\lll\; \delta \quad\implies\quad \mathbb{P}_{x\sim\mu^i}[\,\ell_x^t \text{ is } (\epsilon,\mu,b)\text{-noisy}\,] \;\geq\; 1-\delta.$$

Leaf Lemma is our main technical contribution. The proof appears in Section 4. To whet the reader's appetite, we highlight two interesting challenges that make the lemma non-trivial.

(C1) *Which side is hard?* We need to somehow tease out a hard side for an arbitrary g and this can even depend on the choice of μ. For example, consider $g(b,x) \coloneqq b \oplus \textsc{GapOr}(x)$ where $b \in \{0,1\}$. Rather than μ assigning b at random, the distribution can fix b to either 0 or 1, which reduces g to either $\textsc{GapOr}$ or $\neg\textsc{GapOr}$ (two functions with distinct hard sides).

(C2) *Behaviour of typical leaves.* The existence of μ is often proved using various minimax theorems (we use one due to Blais and Brody [7]). These theorems typically guarantee that any shallow decision tree incurs error at least ϵ on average relative to μ. This does not rule out the following bad scenario: the tree could make error $1/2$ on 2ϵ fraction of the leaves reached and no error on $1-2\epsilon$ fraction of the leaves – here the typical leaves are not noisy!

In order to use Leaf Lemma in the context of composed functions, we generalise it to the direct sum setting where the inputs come from $\mathrm{dom}(g^n) := \mathrm{dom}(g)^n$. Let ℓ be a leaf over nm bits and write $\ell(x) = \prod_{i\in[n]} \ell_i(x^i)$ where $x^i \in \{0,1\}^m$ and each ℓ_i is over m bits. We define

$$\ell \text{ is } (\delta,\epsilon,\mu,b)\textit{-noisy} \quad \overset{\text{def}}{\iff} \quad \ell_i \text{ is } (\epsilon,\mu,b)\text{-noisy for at least } (1-\delta)n \text{ many } i\in[n].$$

Our generalised lemma says that we will typically reach a noisy leaf if we run a shallow decision tree on a random input from the product distribution $\mu^y := \mu^{y_1}\times\cdots\times\mu^{y_n}$ where $y\in\{0,1\}^n$.

▶ **Multileaf Lemma.** *For every partial g and $0<\epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0+\mu^1)$ over* $\mathrm{dom}(g)$ *and a hard side $b\in\{0,1\}$ such that for every deterministic tree t taking inputs from* $\mathrm{dom}(g^n)$ *and having* $\mathrm{depth}(t)/(n\overline{\mathrm{R}}_\epsilon(g)) \lll \delta$*,*

$$\forall y\in\{0,1\}^n: \qquad \mathbb{P}_{x\sim\mu^y}[\,\ell^t_x \text{ is } (\delta,\epsilon,\mu,b)\text{-noisy}\,] \;\geq\; 1-\delta.$$

Given Leaf Lemma the proof of the generalisation is not difficult: we can use linearity of expectation to see that the expected number of queries t makes to most copies of g is low, and hence we can apply Leaf Lemma for those copies. The details appear in Section 5.

3.2 Proof of Theorem 3

We conclude this overview section with a proof of Theorem 3 using Multileaf Lemma. We start with a lemma that shows how the noisy leaves in the domain of g^n can be "simulated" by low-degree conical juntas in the domain of $\textsc{GapOr}^n$. For simplicity, we state the lemma assuming a hard side $b=0$; an analogous lemma holds for $b=1$ by replacing $\textsc{GapOr}$ with $\neg\textsc{GapOr}$.

▶ **Simulation Lemma.** *Let ℓ be a $(\delta,\epsilon,\mu,0)$-noisy leaf over the variables of g^n. There exists a conical junta $p_\ell\colon (\{0,1\}^{\log n})^n \to \mathbb{R}_{\geq 0}$ of degree at most $n\cdot[\delta\log n + \log(1/\epsilon)]$ such that*

$$\forall x\in \mathrm{dom}(\textsc{GapOr}^n_{\log n}): \qquad p_\ell(x) \;=\; \ell(\mu^{\textsc{GapOr}^n_{\log n}(x)}).$$

Proof. We start by defining three conical juntas in the domain of $\textsc{GapOr}_m$ for $m := \log n$. Let $\mathcal{S}^m_k$ be the distribution over multisets obtained by picking k random elements from $[m]$ with replacement.

$$\begin{aligned} q_1(x) &:= \tfrac{2}{m}\textstyle\sum_{i\in[m]} x_i && \text{of degree } 1,\\ q_2(x) &:= \textstyle\prod_{i\in[m]} \bar{x}_i && \text{of degree } m=\log n,\\ q_3(x) &:= \mathbb{E}_{S\sim\mathcal{S}^m_k}\textstyle\prod_{i\in S}\bar{x}_i && \text{of degree } k:=\log(1/\epsilon). \end{aligned}$$

Note the following output values:

$$\begin{aligned} &\forall x\in(\textsc{GapOr}_m)^{-1}(0): && q_1(x)=0, && q_2(x)=1, && q_3(x)=1,\\ &\forall x\in(\textsc{GapOr}_m)^{-1}(1): && q_1(x)=1, && q_2(x)=0, && q_3(x)=2^{-k}=\epsilon. \end{aligned}$$

Let $y = (y^1, \dots, y^n)$ be the input variables of g^n. We write $\ell(y) = \prod_i \ell_i(y^i)$ so that $\ell(\mu^{\textsc{GapOr}^n_m(x)}) = \prod_i \ell_i(\mu^{\textsc{GapOr}_m(x^i)})$. We simulate each factor in this product separately. For $i \in [n]$ consider the function $p_i \colon \{0,1\}^m \to \mathbb{R}_{\geq 0}$ defined by

$$p_i(x) \;:=\; \ell_i(\mu^{\textsc{GapOr}_m(x)}).$$

First note that p_i can always be written as a conical combination of q_1 and q_2 in degree $\log n$. Moreover, if ℓ_i is $(\epsilon, \mu, 0)$-noisy, meaning $\ell_i(\mu^1) \geq \epsilon \cdot \ell_i(\mu^0)$, then we can do better and write p_i as a conical combination of q_1 and q_3 in degree $\log(1/\epsilon)$. We now define $p_\ell := \prod_i p_i$. The claimed bound on the degree of p_ℓ follows because at most δ fraction of the ℓ_i are non-noisy. ◀

We are now ready to prove Theorem 3 using Multileaf Lemma and Simulation Lemma.

Proof of Theorem 3. Suppose for contradiction that T is a randomised decision tree for $f \circ g^n$ having error 1/3 and depth $\gamma n \overline{\mathrm{R}}_{1/n}(g)$ where $\gamma = o(1)$ as $n \to \infty$. Our goal is to construct an $o(n \log n)$-degree $o(1)$-approximate conical junta for $f \circ \textsc{GapOr}^n_{\log n}$ (or $f \circ \neg\textsc{GapOr}^n_{\log n}$).

We make two simplifying assumptions wlog.

1. The randomised tree T has error $o(1)$. To ensure this, we may reduce T's error by running it $1/\sqrt{\gamma} = \omega(1)$ times. This will yield an $o(1)$-error tree of depth $\sqrt{\gamma} n \overline{\mathrm{R}}_{1/n}(g) = o(n\overline{\mathrm{R}}_{1/n}(g))$.
2. There is some $\epsilon := 1/n^{o(1)}$ such that T has depth $o(n\overline{\mathrm{R}}_\epsilon(g))$. To ensure this, we may apply Claim 7 to see that $\gamma n \overline{\mathrm{R}}_{1/n}(f) \leq \sqrt{\gamma} n \overline{\mathrm{R}}_\epsilon(f) \leq o(n\overline{\mathrm{R}}_\epsilon(f))$ where $\epsilon := 1/n^{4\sqrt{\gamma}}$.

We invoke Multileaf Lemma with the above $\epsilon \leq o(1)$ and $\delta := \max\{\gamma^c, \epsilon^c\} \leq o(1)$ for small enough constant $c > 0$. We get a hard distribution μ and a hard side b, say $b = 0$ (case $b = 1$ is similar, but using $\neg\textsc{GapOr}$), such that the following holds: For every t in the support of T if we run t on a random input $x \sim \mu^y$, where $y \in \{0,1\}^n$, then the leaf reached ℓ^t_x will be $(\delta, \epsilon, \mu, 0)$-noisy with probability $1 - o(1)$. This allows us to effectively ignore non-noisy leaves: denoting by $\mathcal{N}(t) \subseteq \mathcal{A}(t)$ the set of accepting leaves that are $(\delta, \epsilon, \mu, 0)$-noisy, we have

$$\begin{aligned} \forall y \in \{0,1\}^n: \quad \mathbb{E}_{x\sim\mu^y}[t(x)] \;&=\; \mathbb{E}_{x\sim\mu^y}\left[\textstyle\sum_{\ell\in\mathcal{A}(t)} \ell(x)\right] && \text{(Using (1))} \\ &\in\; \mathbb{E}_{x\sim\mu^y}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} \ell(x)\right] \pm o(1). && (2) \end{aligned}$$

We now define the approximating conical junta by

$$p(x) \;:=\; \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} p_\ell(x)\right],$$

where the p_ℓ are given by Simulation Lemma. Hence p has degree at most

$$n \cdot [\delta \log n + \log(1/\epsilon)] \;=\; n \cdot [o(1)\log n + \log n^{o(1)}] \;=\; o(n \log n).$$

We finish the proof of Theorem 3 by verifying that p indeed $o(1)$-approximates $f \circ \textsc{GapOr}^n_{\log n}$.

$$\begin{aligned} \forall x: \quad p(x) \;&=\; \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} p_\ell(x)\right] \\ &=\; \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} \ell(\mu^y)\right] && (y := \textsc{GapOr}^n_{\log n}(x)) \\ &=\; \mathbb{E}_{t\sim T}\left[\textstyle\sum_{\ell\in\mathcal{N}(t)} \mathbb{E}_{x'\sim\mu^y}[\ell(x')]\right] \\ &=\; \mathbb{E}_{t\sim T}\left[\mathbb{E}_{x'\sim\mu^y}[\textstyle\sum_{\ell\in\mathcal{N}(t)} \ell(x')]\right] \\ &\in\; \mathbb{E}_{t\sim T}\left[\mathbb{E}_{x'\sim\mu^y}[t(x')]\right] \pm o(1) && \text{(Using (2))} \end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{x'\sim\mu^y}\left[\mathbb{E}_{t\sim T}[t(x')]\right] \pm o(1) \\
&\in \mathbb{E}_{x'\sim\mu^y}\left[(f\circ g^n)(x')\right] \pm o(1) && (T \text{ has error } o(1)) \\
&= f(y) \pm o(1) \\
&= (f\circ \textsc{GapOr}^n_{\log n})(x) \pm o(1). && \blacktriangleleft
\end{aligned}
$$

4 Proof of Leaf Lemma

We prove Leaf Lemma in three subsections.

(§4.1) We start by recalling a distributional characterisation due to Blais and Brody [7] of expected query complexity $\overline{\mathrm{R}}_\epsilon$ using decision trees that can "abort".

(§4.2) We then formulate a Hard Side Lemma, which encapsulates the core challenge in finding the hard side of a given function g and from which Leaf Lemma is easy to derive.

(§4.3) Finally, we prove the Hard Side Lemma.

4.1 Distributional characterisation of $\overline{\mathrm{R}}_\epsilon$ due to Blais–Brody

A *(deterministic) abort-tree* t is a decision tree that outputs either a boolean value (0 or 1) or the *abort symbol* $\perp$. When an abort-tree is trying to compute a boolean function g, we do not consider the output $\perp$ as an "error"; the tree simply gives up on the computation. Indeed, we say that $t(x)$ *errs* iff $t(x) = 1 - g(x)$, that is, $t(x) \neq \perp$ and $t(x) \neq g(x)$. As before, a *randomised abort-tree* is a probability distribution over deterministic abort-trees. For $\gamma \in (0,1)$ and $\epsilon \in [0,1/2)$ we define $\mathrm{R}_{\gamma,\epsilon}(g)$ as the least (worst-case) depth of a randomised abort-tree T such that for all $x \in \mathrm{dom}(g)$:

$$\mathbb{P}_{t\sim T}[\,t(x) = \perp\,] \;\leq\; \gamma \quad\text{and}\quad \mathbb{P}_{t\sim T}[\,t(x)\,\text{errs}\,] \;\leq\; \epsilon.$$

We formulate a distributional version of $\mathrm{R}_{\gamma,\epsilon}(g)$ as follows. For a distribution μ over $\mathrm{dom}(g)$, we define $\mathrm{D}^{\mu}_{\gamma,\epsilon}(g)$ as the least depth of a deterministic abort-tree t such that

$$\mathbb{P}_{x\sim\mu}[\,t(x) = \perp\,] \;\leq\; \gamma \quad\text{and}\quad \mathbb{P}_{x\sim\mu}[\,t(x)\,\text{errs}\,] \;\leq\; \epsilon.$$

The following two lemmas from [7, §3.1] connect abort-trees and $\overline{\mathrm{R}}_\epsilon(g)$.

▶ **Lemma 8** (Abort vs. expected depth). *For every $\epsilon \in [0,1/2)$ and $\gamma \in (0,1)$,*

$$\gamma\cdot \mathrm{R}_{\gamma,\,\epsilon}(g) \;\leq\; \overline{\mathrm{R}}_\epsilon(g) \;\leq\; \tfrac{1}{1-\gamma}\cdot \mathrm{R}_{\gamma,\,(1-\gamma)\epsilon}(g).$$

▶ **Lemma 9** (Minimax). *For every $\epsilon \in [0,1/2)$, $\gamma \in (0,1)$, and $\alpha,\beta \in (0,1)$ with $\alpha+\beta \leq 1$,*

$$\max\nolimits_\mu \mathrm{D}^{\mu}_{\gamma/\alpha,\,\epsilon/\beta}(g) \;\leq\; \mathrm{R}_{\gamma,\,\epsilon}(g) \;\leq\; \max\nolimits_\mu \mathrm{D}^{\mu}_{\alpha\gamma,\,\beta\epsilon}(g).$$

4.2 Statement of Hard Side Lemma

When searching for the hard side of a partial function g under a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$, it is convenient to study a more symmetric notion of noisiness than the one-sided variant defined earlier. For a leaf $\ell \in \mathcal{L}(t)$ of an abort-tree t, we define the *relative error* $\mathrm{re}(\ell,\mu)$ so that if ℓ is an aborting leaf, then $\mathrm{re}(\ell,\mu) \coloneqq 0$; otherwise

$$\mathrm{re}(\ell,\mu) \;\coloneqq\; \frac{\min\{\ell(\mu^0),\, \ell(\mu^1)\}}{\ell(\mu^0)+\ell(\mu^1)} \;\in\; [0,1/2].$$

This definition captures the best achievable error of a leaf in an abort-tree. Namely, let us say that t is *μ-smart* if every non-abort leaf $\ell \in \mathcal{L}(t)$ outputs a boolean value $i \in \{0,1\}$ that maximises $\ell(\mu^i)$. Then for every leaf ℓ in a μ-smart t we have $\mathbb{P}_{x\sim\mu}[\,t(x)\,\text{errs} \mid \ell^t_x = \ell\,] = \mathrm{re}(\ell,\mu)$. An easy calculation gives the following claim, which we record for future use.

▷ Claim 10. For a μ-smart t we have $\mathbb{P}_{x\sim\mu}[\,t(x)\,\text{errs}\,] = \mathbb{E}_{x\sim\mu}[\mathrm{re}(\ell^t_x,\mu)]$.

Another easy calculation shows that relative error implies noisiness.

▷ Claim 11. If $\mathrm{re}(\ell,\mu) \geq \epsilon$, then ℓ is (ϵ,μ,b)-noisy for both $b \in \{0,1\}$.

We are now ready to formulate Hard Side Lemma, which isolates the technical challenge 3.1 (discussed in Subsection 3.1): Every partial function g admits a balanced distribution μ and a hard side b such that if we run a shallow abort-tree on the hard side μ^b of μ, then t must either abort with high probability or we reach a leaf of noticeable error (in expectation).

▶ **Hard Side Lemma.** *For every partial function g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over $\mathrm{dom}(g)$ and a hard side $b \in \{0,1\}$ such that for any deterministic abort-tree t with $\mathrm{depth}(t)/\overline{\mathrm{R}}_\epsilon(g) \lll \delta$ we have either*

$$\mathbb{P}_{x\sim\mu^b}[\,t(x) = \bot\,] \;>\; 1-\delta \qquad \textit{or} \qquad \mathbb{E}_{x\sim\mu^b}[\,\mathrm{re}(\ell^t_x,\mu)\,] \;>\; \epsilon. \tag{3}$$

We defer the proof until Subsection 4.3. We first use the lemma to prove Leaf Lemma, and here is where we address challenge 3.1: we exploit the high abort probability (namely, $1-\delta$) guaranteed by Hard Side Lemma to show that typical leaves are noisy.

▶ **Leaf Lemma** (restated). *For every partial g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over $\mathrm{dom}(g)$ and a hard side $b \in \{0,1\}$ such that for every deterministic tree t and $i \in \{0,1\}$:*

$$\frac{\mathbb{E}_{x\sim\mu^i}[q(t,x)]}{\overline{\mathrm{R}}_\epsilon(g)} \;\lll\; \delta \qquad \Longrightarrow \qquad \mathbb{P}_{x\sim\mu^i}[\,\ell^t_x \text{ is } (\epsilon,\mu,b)\text{-noisy}\,] \;\geq\; 1-\delta.$$

Proof. We observe first that regardless of μ, b, or even the expected depth of t, the lemma holds for the easy side $i = 1-b$. Indeed, if we let $\mathcal{B} \subseteq \mathcal{L}(t)$ denote the set of non-(ϵ,μ,b)-noisy leaves,

$$\mathbb{P}_{x\sim\mu^{1-b}}[\ell^t_x \in \mathcal{B}] \;=\; \textstyle\sum_{\ell\in\mathcal{B}} \ell(\mu^{1-b}) \;<\; \epsilon \textstyle\sum_{\ell\in\mathcal{B}} \ell(\mu^b) \;\leq\; \epsilon \textstyle\sum_{\ell\in\mathcal{L}(t)} \ell(\mu^b) \;=\; \epsilon \;\leq\; \delta.$$

Let us then focus on the interesting case $i = b$ where the careful choice of μ and b is essential. We invoke Hard Side Lemma with parameters ϵ and $\dot\delta := \delta^2$ (assuming suitably $0 < \epsilon \lll \dot\delta \ll 1$) to obtain μ and b such that for every abort-tree $\dot t$ with $\mathrm{depth}(\dot t) \leq \dot\delta^c\overline{\mathrm{R}}_\epsilon(g)$ the property (3) holds (with dotted parameters). Let $x \sim \mu^b$ henceforth and write $\mathrm{re}(\ell) := \mathrm{re}(\ell,\mu)$ for short. Suppose t satisfies $\mathbb{E}_x[q(t,x)] \leq \dot\delta^{c+2}\overline{\mathrm{R}}_\epsilon(g)$ (where we chose $2(c+2)$ as the exponent hidden by $\lll$). Recalling from Claim 11 that relative error implies noisiness, our goal is to show

$$\mathbb{P}_x[\,\mathrm{re}(\ell^t_x) \geq \epsilon\,] \;\geq\; 1-\delta. \tag{4}$$

We convert t into an abort-tree by letting t' be a modification of t that aborts whenever more than $\dot\delta^c\overline{\mathrm{R}}_\epsilon(g)$ queries are made. Using Markov's inequality and the low expected depth of t,

$$\mathbb{P}_x[t'(x) = \bot] \;=\; \mathbb{P}_x[q(t,x) > \dot\delta^c\overline{\mathrm{R}}_\epsilon(g)] \;\leq\; \mathbb{E}_x[q(t,x)]/\dot\delta^c\overline{\mathrm{R}}_\epsilon(g) \;\leq\; \dot\delta^2.$$

We also have $\mathbb{P}_x[\mathrm{re}(\ell_x^t) \geq \epsilon] \geq \mathbb{P}_x[\mathrm{re}(\ell_x^{t'}) \geq \epsilon]$ since we only made more executions abort. To prove (4), suppose for contradiction that $\mathbb{P}_x[\mathrm{re}(\ell_x^{t'}) \geq \epsilon] < 1-\delta$. Let $\dot{t}$ be a further modification of t' that aborts any leaf $\ell \in \mathcal{L}(t')$ with $\mathrm{re}(\ell) \geq \epsilon$. Note that

$$\mathbb{P}_x[\dot{t}(x) = \bot] \;\leq\; \mathbb{P}_x[t'(x) = \bot] + \mathbb{P}_x[\mathrm{re}(\ell_x^{t'}) \geq \epsilon] \;\leq\; \dot{\delta}^2 + 1 - \delta \;\leq\; 1 - \dot{\delta}.$$

Hence we get from (dotted) property (3) that $\mathbb{E}_x[\mathrm{re}(\ell_x^{\dot{t}})] > \epsilon$. But this contradicts the fact that $\mathrm{re}(\ell) < \epsilon$ for all $\ell \in \mathcal{L}(\dot{t})$ by construction. This verifies (4) and concludes the proof. ◀

4.3 Proof of Hard Side Lemma

Let ν be a distribution that witnesses $D := \max_{\nu'} \mathrm{D}^{\nu'}_{1-\delta,\,\epsilon^{1/3}}(g)$ so that every abort-tree t with $\mathrm{depth}(t) < D$ fails to satisfy at least one of the following:

$$\mathbb{P}_{x\sim\nu}[\,t(x) = \bot\,] \;\leq\; 1-\delta, \tag{5}$$

$$\mathbb{P}_{x\sim\nu}[\,t(x)\,\mathrm{errs}\,] \;\leq\; \epsilon^{1/3}. \tag{6}$$

As a minor technicality, we re-balance ν. We can write $\nu = \lambda\mu^0 + (1-\lambda)\mu^1$ where $\lambda \in (0,1)$ and μ^i is a distribution supported on $g^{-1}(i)$. We define $\mu := \frac{1}{2}(\mu^0 + \mu^1)$ as our balanced distribution.

Assume towards a contradiction that there does not exist a hard side for μ, that is, the claim of the lemma fails for both $b \in \{0,1\}$. This means there exists two abort-trees t_0 and t_1 of depth at most $\delta^3\overline{\mathrm{R}}_\epsilon(g)$ (where we chose 3 as the exponent hidden by $\lll$) such that for both $b \in \{0,1\}$:

$$\mathbb{P}_{x\sim\mu^b}[\,t_b(x) = \bot\,] \;\leq\; 1-\delta, \tag{7}$$

$$\mathbb{E}_{x\sim\mu^b}[\,\mathrm{re}(\ell_x^{t_b},\mu)\,] \;\leq\; \epsilon. \tag{8}$$

We will use t_0 and t_1 to construct a third tree t that computes g too well relative to ν contradicting our choice of ν. We may assume wlog that t_0 and t_1 are *μ-smart*, since the properties (7)–(8) do not depend on the boolean leaf-labels (only whether a leaf aborts or not). We now define t as follows: On input x we run both $t_0(x)$ and $t_1(x)$; if $t_0(x) \neq \bot$, we output $t_0(x)$; otherwise we output $t_1(x)$. We will show that t has $\mathrm{depth}(t) < D$ and satisfies (5)–(6), which will contradict our choice of ν.

Tree t is shallow. We have the following chain of inequalities

$$\begin{aligned}\mathrm{depth}(t) \;&\leq\; 2\delta^3\overline{\mathrm{R}}_\epsilon(g) \;\leq\; 32\delta^3\overline{\mathrm{R}}_{\epsilon^{1/4}}(g) \;<\; \delta^2\overline{\mathrm{R}}_{\epsilon^{1/4}}(g) \;\leq\; \mathrm{R}_{1-\delta^2,\,\delta^2\epsilon^{1/4}}(g)\\ &\leq\; \max\nolimits_{\nu'} \mathrm{D}^{\nu'}_{(1-\delta^2)^2,\,\delta^4\epsilon^{1/4}}(g) \;\leq\; \max\nolimits_{\nu'} \mathrm{D}^{\nu'}_{1-\delta,\,\epsilon^{1/3}}(g) \;=:\; D.\end{aligned}$$

The first inequality uses the definition of t. Second uses error reduction (Claim 7 with $k := 4$). Third uses $\delta \ll 1$. Fourth uses Lemma 8 (with $\gamma := 1-\delta^2$). Fifth uses the minimax lemma (Lemma 9 with $\alpha := 1-\delta^2$, $\beta := \delta^2$). The final inequality uses $\epsilon \lll \delta \ll 1$.

Tree t has bounded abort. We verify property (5) by

$$\begin{aligned}\mathbb{P}_{x\sim\nu}[t(x) = \bot] \;&=\; \mathbb{P}_{x\sim\nu}[t_0(x) = \bot \wedge t_1(x) = \bot]\\ &=\; \lambda\,\mathbb{P}_{x\sim\mu^0}[t_0(x) = \bot \wedge t_1(x) = \bot]\\ &\qquad + (1-\lambda)\,\mathbb{P}_{x\sim\mu^1}[t_0(x) = \bot \wedge t_1(x) = \bot]\\ &\leq\; \lambda\,\mathbb{P}_{x\sim\mu^0}[t_0(x) = \bot] + (1-\lambda)\,\mathbb{P}_{x\sim\mu^1}[t_1(x) = \bot]\\ &\leq\; 1-\delta. \qquad\qquad\qquad\qquad \text{(Using (7))}\end{aligned}$$

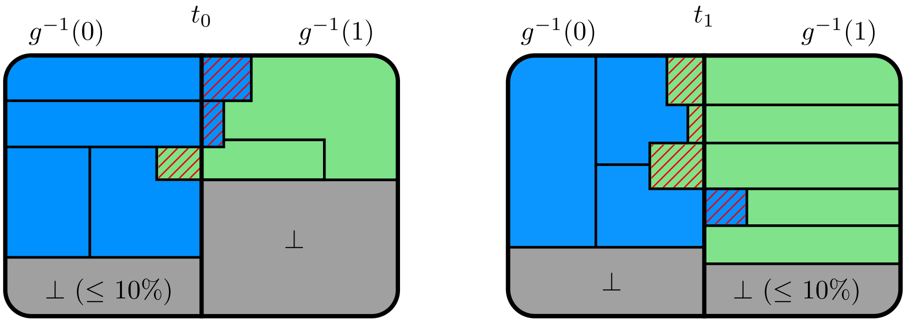


Figure 1 Two trees t_0 and t_1 in the proof of Hard Side Lemma. The leaves partition $\mathrm{dom}(g)$ into subcubes where grey leaves output $\perp$, green leaves output 1, and blue leaves output 0. Hatched regions are error. We are promised that, e.g., t_0 has bounded abort (10% in our figure) over μ^0, but not necessarily over μ^1.

Tree *t* errs rarely. We start with a claim that says that if the expected relative error is low over one side μ^b of μ, then a μ-smart tree errs rarely over the whole distribution μ.

▷ Claim 12. Let t' be μ-smart and $b \in \{0,1\}$. If $\mathbb{E}_{x\sim\mu^b}[\mathrm{re}(\ell_x^{t'},\mu)] \leq \epsilon$ then $\mathbb{P}_{x\sim\mu}[t'(x)\,\mathrm{errs}] \leq \epsilon^{1/2}$.

Proof. We prove the claim for $b = 0$ as the other case is analogous. Since t' and μ are fixed, we drop them from notation writing $\mathrm{re}(\ell) \coloneqq \mathrm{re}(\ell,\mu)$, $\ell_x \coloneqq \ell_x^{t'}$, $\mathcal{L} \coloneqq \mathcal{L}(t')$. We argue that relative error on one side of the distribution must spill over to the other side:

$$\mathbb{E}_{x\sim\mu^0}[\mathrm{re}(\ell_x)] = \textstyle\sum_{\ell\in\mathcal{L}} \ell(\mu^0)\,\mathrm{re}(\ell) \geq \sum_{\ell\in\mathcal{L}} \ell(\mu^1)\,\mathrm{re}(\ell)^2 = \mathbb{E}_{x\sim\mu^1}[\mathrm{re}(\ell_x)^2] \geq \mathbb{E}_{x\sim\mu^1}[\mathrm{re}(\ell_x)]^2.$$

Here the first inequality used $\ell(\mu^0) \geq \ell(\mu^1)\,\mathrm{re}(\ell)$ (from Claim 11) and the second inequality used Jensen's inequality. It follows that $\mathbb{E}_{x\sim\mu^1}[\mathrm{re}(\ell_x)] \leq \mathbb{E}_{x\sim\mu^0}[\mathrm{re}(\ell_x)]^{1/2} \leq \epsilon^{1/2}$ and therefore $\mathbb{E}_{x\sim\mu}[\mathrm{re}(\ell_x)] \leq \epsilon^{1/2}$. The claim then follows from Claim 10. ◁

We now verify property (6), which concludes the proof of Hard Side Lemma.

$$\begin{aligned}
\mathbb{P}_{x\sim\nu}[t(x)\,\mathrm{errs}] &\leq \mathbb{P}_{x\sim\nu}[t_0(x)\,\mathrm{errs} \vee t_1(x)\,\mathrm{errs}] \\
&\leq \textstyle\sum_{b\in\{0,1\}} \mathbb{P}_{x\sim\nu}[t_b(x)\,\mathrm{errs}] \\
&= \textstyle\sum_{b\in\{0,1\}} \lambda\,\mathbb{P}_{x\sim\mu^0}[t_b(x)\,\mathrm{errs}] + (1-\lambda)\,\mathbb{P}_{x\sim\mu^1}[t_b(x)\,\mathrm{errs}] \\
&\leq \textstyle\sum_{b\in\{0,1\}} 2\,\mathbb{P}_{x\sim\mu}[t_b(x)\,\mathrm{errs}] \\
&\leq \textstyle\sum_{b\in\{0,1\}} 2\cdot\epsilon^{1/2} && \text{(Claim 12 and (8))} \\
&= 4\epsilon^{1/2} \\
&\leq \epsilon^{1/3}. && (\epsilon \ll 1)
\end{aligned}$$

5 Proof of Multileaf Lemma

▶ **Multileaf Lemma** (restated). *For every partial g and $0 < \epsilon \lll \delta \ll 1$, there exists a distribution $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ over $\mathrm{dom}(g)$ and a hard side $b \in \{0,1\}$ such that for every deterministic tree t taking inputs from $\mathrm{dom}(g^n)$ and having $\mathrm{depth}(t)/(n\overline{\mathrm{R}}_\epsilon(g)) \lll \delta$,*

$$\forall y \in \{0,1\}^n: \qquad \mathbb{P}_{x\sim\mu^y}[\,\ell_x^t \text{ is } (\delta,\epsilon,\mu,b)\text{-noisy}\,] \;\geq\; 1-\delta.$$

Proof. Apply Leaf Lemma with parameters ϵ and $\dot{\delta} := \delta^3$ (assuming suitably $0 < \epsilon \lll \dot{\delta} \ll 1$) to obtain $\mu = \frac{1}{2}(\mu^0 + \mu^1)$ and $b \in \{0,1\}$ that satisfy the lemma for trees of depth at most $\dot{\delta}^c \overline{\mathrm{R}}_\epsilon(g)$. Fix $y \in \{0,1\}^n$ and a deterministic tree t over $\mathrm{dom}(g^n)$ with $\mathrm{depth}(t) \le \dot{\delta}^{c+4} n \overline{\mathrm{R}}_\epsilon(g)$ (where we chose $3(c+4)$ as the exponent hidden by $\lll$).

Here is the plan for our proof. An input $x \in \mathrm{dom}(g^n)$ can be seen as inducing several subtrees of t corresponding to distinct coordinates $i \in [n]$. Indeed, define $t^{x,i}$ as the tree over inputs from $\mathrm{dom}(g)$ that is obtained from t by substituting x as its input variables except retaining x^i as free variables. If we can show that $t^{x,i}$ has shallow depth in expectation over an input $z \sim \mu^{y_i}$ then we can hope to use and argue that the reached leaf $\ell_z \in \mathcal{L}(t^{x,i})$ (which is one of the n components of a leaf of t) is typically (ϵ, μ, b)-noisy.

Let us formalise this plan. Let $x \sim \mu^y$ henceforth. For $i \in [n]$ we define two events

$$\begin{array}{rccl} \textit{i-th tree is shallow:} & S_i(x) & \overset{\text{def}}{\iff} & \mathbb{E}_{z\sim\mu^{y_i}}[q(t^{x,i}, z)] \;\le\; \dot{\delta}^c \overline{\mathrm{R}}_\epsilon(g), \\ \textit{i-th leaf is noisy:} & N_i(x) & \overset{\text{def}}{\iff} & \ell_{x^i} \in \mathcal{L}(t^{x,i}) \text{ is } (\epsilon, \mu, b)\text{-noisy.} \end{array}$$

Note that Leaf Lemma states $\mathbb{P}_x[N_i \mid S_i] \ge 1 - \dot{\delta}$. Thinking of S_i and N_i as indicator variables, we define $S := \frac{1}{n}\sum_i S_i$ and $N := \frac{1}{n}\sum_i N_i$. With this notation, Multileaf Lemma becomes equivalent to

$$\mathbb{P}_x[N \ge 1 - \delta] \;\ge\; 1 - \delta. \tag{9}$$

To show this, we compute as follows (using Claim 13 that is proved below)

$$\begin{aligned} \mathbb{E}_x[N] &= \tfrac{1}{n}\textstyle\sum_i \mathbb{P}_x[N_i] \\ &\ge \tfrac{1}{n}\textstyle\sum_i (1-\dot{\delta})\,\mathbb{P}[S_i] && \text{(Leaf Lemma)} \\ &= (1-\dot{\delta})\,\mathbb{E}_x[S] \\ &\ge (1-\dot{\delta})(1-\dot{\delta}) && \text{(Claim 13)} \\ &\ge 1 - \delta^2. && (\dot{\delta} := \delta^3 \ll 1) \end{aligned}$$

Hence (9) follows by applying Markov's inequality to the nonnegative random variable $1 - N \ge 0$. This completes the proof apart from the following claim. ◀

▷ Claim 13. $\mathbb{E}_x[S] \ge 1 - \dot{\delta}$.

Proof. Let $q_i(t,x)$ denote the number of queries made by t to the i-th component of x. Define $x^{i\leftarrow z}$ as a copy of x but where z is inserted at the i-th component. Note that $q_i(t, x^{i\leftarrow z}) = q(t^{x,i}, z)$. Linearity of expectation gives

$$\textstyle\sum_{i\in[n]} \mathbb{E}_x[q_i(t,x)] \;\le\; \mathrm{depth}(t) \;\le\; \dot{\delta}^{c+4} n \overline{\mathrm{R}}_\epsilon(g). \tag{10}$$

Define $\mathcal{I} \subseteq [n]$ as the set of coordinates i satisfying

$$\mathbb{E}_x[q_i(t,x)] \;\le\; \dot{\delta}^{c+2} \overline{\mathrm{R}}_\epsilon(g). \tag{11}$$

We have that $|\mathcal{I}| \ge (1 - \dot{\delta}^2)n$ as otherwise more than $\dot{\delta}^2 n$ terms in the sum (10) are larger than $\dot{\delta}^{c+2}\overline{\mathrm{R}}_\epsilon(g)$ contradicting the upper bound on $\mathrm{depth}(t)$. Fix $i \in \mathcal{I}$. Sampling $x \sim \mu^y$ is equivalent to first taking $x \sim \mu^y$, then sampling independently $z \sim \mu^{y_i}$, and finally outputting $x^{i\leftarrow z}$. Hence

$$\mathbb{E}_x \mathbb{E}_{z\sim\mu^{y_i}}[q_i(t, x^{i\leftarrow z})] \;=\; \mathbb{E}_x[q_i(t,x)] \;\le\; \dot{\delta}^{c+2}\overline{\mathrm{R}}_\epsilon(g).$$

We get from Markov's inequality and the above that

$$\mathbb{P}_x[\neg S_i] \;=\; \mathbb{P}_x\left[\mathbb{E}_{z\sim\mu^{y_i}}[q_i(t,x^{i\leftarrow z})] > \dot{\delta}^c\overline{\mathrm{R}}_\epsilon(g)\right] \;\le\; \dot{\delta}^2. \tag{12}$$

In conclusion,

$$\mathbb{E}_x[S] \;\ge\; \tfrac{1}{n}\textstyle\sum_{i\in\mathcal{I}}\mathbb{P}_x[S_i] \;\ge\; \tfrac{1}{n}|\mathcal{I}|\cdot(1-\dot{\delta}^2) \;\ge\; (1-\dot{\delta}^2)^2 \;\ge\; 1-\dot{\delta}. \qquad \lhd$$

References

1 Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity. In *Proceedings of the 37th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 10:1–10:13. Schloss Dagstuhl, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.10.

2 Andrew Bassilakis, Andrew Drucker, Mika Göös, Lunjia Hu, Weiyun Ma, and Li-Yang Tan. The power of many samples in query complexity. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168, pages 9:1–9:18. Schloss Dagstuhl, 2020. doi:10.4230/LIPIcs.ICALP.2020.9.

3 Shalev Ben-David and Eric Blais. A new minimax theorem for randomized algorithms. In *Proceedings of the 61st Symposium on Foundations of Computer Science (FOCS)*, pages 403–411, 2020. doi:10.1109/FOCS46700.2020.00045.

4 Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions. In *Proceedings of the 61st Symposium on Foundations of Computer Science (FOCS)*, pages 240–246, 2020. doi:10.1109/FOCS46700.2020.00031.

5 Shalev Ben-David, Mika Göös, Robin Kothari, and Thomas Watson. When is amplification necessary for composition in randomized query complexity? In *Proceedings of the 22nd International Conference on Randomization and Computation (RANDOM)*, volume 176, pages 28:1–28:16. Schloss Dagstuhl, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.28.

6 Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. *Theory of Computing*, 14(1):1–27, 2018. doi:10.4086/toc.2018.v014a005.

7 Eric Blais and Joshua Brody. Optimal separation and strong direct sum for randomized query complexity. In *Proceedings of the 34th Computational Complexity Conference (CCC)*, pages 29:1–29:17. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.CCC.2019.29.

8 Joshua Brody, Jae Tak Kim, Peem Lerdputtipongporn, and Hariharan Srinivasulu. A strong XOR lemma for randomized query complexity. Technical report, arXiv, 2020. arXiv:2007.05580.

9 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

10 Chinmoy Dutta and Jaikumar Radhakrishnan. Lower bounds for noisy wireless networks using sampling algorithms. In *Proceedings of the 49th Symposium on Foundations of Computer Science (FOCS)*, pages 394–402. IEEE, 2008. doi:10.1109/FOCS.2008.72.

11 William Evans and Nicholas Pippenger. Average-case lower bounds for noisy boolean decision trees. *SIAM Journal on Computing*, 28(2):433–446, 1998. doi:10.1137/S0097539796310102.

12 Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994. doi:10.1137/S0097539791195877.

13 Dmitry Gavinsky, Troy Lee, Miklos Santha, and Swagato Sanyal. A composition theorem for randomized query complexity via max-conflict complexity. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 64:1–64:13. Schloss Dagstuhl, 2019. doi:10.4230/LIPIcs.ICALP.2019.64.

14 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016. doi:10.1137/15M103145X.

15 Navin Goyal and Michael Saks. Rounds vs. queries tradeoff in noisy computation. *Theory of Computing*, 6(1):113–134, 2010. doi:10.4086/toc.2010.v006a006.

16 Rahul Jain and Hartmut Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 25th Conference on Computational Complexity (CCC)*, pages 247–258. IEEE, 2010. doi:10.1109/CCC.2010.31.

17 Rahul Jain, Hartmut Klauck, and Miklos Santha. Optimal direct sum results for deterministic and randomized decision tree complexity. *Information Processing Letters*, 110(20):893–897, 2010. doi:10.1016/j.ipl.2010.07.020.

18 Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

19 Jedrzej Kaniewski, Troy Lee, and Ronald de Wolf. Query complexity in expectation. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 761–772. Springer, 2015. doi:10.1007/978-3-662-47672-7_62.

20 Claire Kenyon and Valerie King. On boolean decision trees with faulty nodes. *Random Structures and Algorithms*, 5(3):453–464, 1994. doi:10.1002/rsa.3240050306.

21 Troy Lee, Rajat Mittal, Ben Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd Symposium on Foundations of Computer Science (FOCS)*, pages 344–353. IEEE, 2011. doi:10.1109/FOCS.2011.75.

22 Ilan Newman. Computing in fault tolerant broadcast networks and noisy decision trees. *Random Structures and Algorithms*, 34(4):478–501, 2009. doi:10.1002/rsa.20240.

23 Ben Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Symposium on Discrete Algorithms (SODA)*, pages 560–569. SIAM, 2011.

24 Petr Savický. On determinism versus unambiquous nondeterminism for decision trees. Technical Report TR02-009, Electronic Colloquium on Computational Complexity (ECCC), 2002. URL: http://eccc.hpi-web.de/report/2002/009/.

25 Avishay Tal. Properties and applications of boolean function composition. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 441–454. ACM, 2013. doi:10.1145/2422436.2422485.

Hitting Sets and Reconstruction for Dense Orbits in $\mathbf{VP_e}$ and $\boldsymbol{\Sigma\Pi\Sigma}$ Circuits

Dori Medini ✉
StarkWare Industries Ltd., Netanya, Israel

Amir Shpilka ✉
Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract

In this paper we study polynomials in $\mathrm{VP_e}$ (polynomial-sized formulas) and in $\Sigma\Pi\Sigma$ (polynomial-size depth-3 circuits) whose orbits, under the action of the affine group $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ (the action of $(A, \boldsymbol{b}) \in \mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ on a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ is defined as $(A, \boldsymbol{b}) \circ f = f(A^T\boldsymbol{x} + \boldsymbol{b})$), are *dense* in their ambient class. We construct hitting sets and interpolating sets for these orbits as well as give reconstruction algorithms. Specifically, we obtain the following results:

1. For $\mathrm{C}_n(\ell_1(\boldsymbol{x}), \ldots, \ell_n(\boldsymbol{x})) \triangleq \mathrm{Trace}\left(\begin{pmatrix}\ell_1(\boldsymbol{x}) & 1\\ 1 & 0\end{pmatrix} \cdot \ldots \cdot \begin{pmatrix}\ell_n(\boldsymbol{x}) & 1\\ 1 & 0\end{pmatrix}\right)$, where the ℓ_is are linearly independent linear functions, we construct a polynomial-sized interpolating set, and give a polynomial-time reconstruction algorithm. By a result of Bringmann, Ikenmeyer and Zuiddam, the set of all such polynomials is dense in $\mathrm{VP_e}$ [14], thus our construction gives the first polynomial-size interpolating set for a dense subclass of $\mathrm{VP_e}$.
2. For polynomials of the form $\mathrm{ANF}_\Delta(\ell_1(\boldsymbol{x}), \ldots, \ell_{4^\Delta}(\boldsymbol{x}))$, where $\mathrm{ANF}_\Delta(\boldsymbol{x})$ is the canonical read-once formula in *alternating normal form*, of depth 2Δ, and the ℓ_is are linearly independent linear functions, we provide a quasipolynomial-size interpolating set. We also observe that the reconstruction algorithm of [35] works for *all* polynomials in this class. This class is also dense in $\mathrm{VP_e}$.
3. Similarly, we give a quasipolynomial-sized hitting set for read-once formulas (not necessarily in alternating normal form) composed with a set of linearly independent linear functions. This gives another dense class in $\mathrm{VP_e}$.
4. We give a quasipolynomial-sized hitting set for polynomials of the form $f(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x}))$, where f is an m-variate s-sparse polynomial. and the ℓ_is are linearly independent linear functions in $n \geq m$ variables. This class is dense in $\Sigma\Pi\Sigma$.
5. For polynomials of the form $\sum_{i=1}^{s}\prod_{j=1}^{d}\ell_{i,j}(\boldsymbol{x})$, where the $\ell_{i,j}$s are linearly independent linear functions, we construct a polynomial-sized interpolating set. We also observe that the reconstruction algorithm of [45] works for *every* polynomial in the class. This class is dense in $\Sigma\Pi\Sigma$.

As $\mathrm{VP} = \mathrm{VNC}^2$, our results for $\mathrm{VP_e}$ translate immediately to VP with a quasipolynomial blow up in parameters. If any of our hitting or interpolating sets could be made *robust* then this would immediately yield a hitting set for the superclass in which the relevant class is dense, and as a consequence also a lower bound for the superclass. Unfortunately, we also prove that the kind of constructions that we have found (which are defined in terms of k-independent polynomial maps) do not necessarily yield robust hitting sets.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Algebraic complexity, VP, VNP, algebraic circuits, algebraic formula

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.19

Related Version *Full Version*: `https://arxiv.org/pdf/2102.05632`

Funding The research leading to these results has received funding from the Israel Science Foundation (grant number 514/20) and from the Len Blavatnik and the Blavatnik Family foundation.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 19; pp. 19:1–19:27
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Proving lower bounds on the size of algebraic circuits (also called arithmetic circuits), is an outstanding open problem in algebraic complexity. In spite of much effort, only a handful of lower bounds are known (a detailed account of most known lower bounds can be found in the excellent survey of Saptharishi [61]). One common theme of most known lower bounds is that they are proved using *algebraic arguments*. That is, a proof of a lower bound for a class of circuits $\mathcal{C}$, usually has the following structure: one comes up with a set of (nonzero) polynomials $F_1, \ldots, F_m$, in $N = \binom{n+d}{d}$ many variables, such that the coefficient vector of every n-variate, degree-d polynomial that can be computed in $\mathcal{C}$, is a common zero of all the F_is (such F_is are called *separating polynomials*). Then, one exhibits a polynomial f whose coefficient vector is not a common zero, thus proving $f \notin \mathcal{C}$. As an example one can immediately see that the well known partial derivative technique, and its successor, shifted partial derivative technique, are algebraic. Grochow [29] demonstrated this for most of the known lower bound proofs. As the set of common zeros of a set of polynomials is closed,[1] this immediately implies that if we prove that $f \notin \mathcal{C}$ using an algebraic argument, then the same argument also implies that $f \notin \overline{\mathcal{C}}$, the closure of $\mathcal{C}$. Recall that, in characteristic zero, the closure of a class $\mathcal{C}$ is the set of all polynomials that are limit points of sequences of polynomials from $\mathcal{C}$, where convergence is coefficient-wise (see Definition 9 for a general definition over arbitrary characteristic). As most known techniques are algebraic, we see that for proving a lower bound for a class $\mathcal{C}$ one actually has to consider the larger, and less structured class, $\overline{\mathcal{C}}$.

Geometric Complexity Theory (GCT for short), which was initiated by Mulmuley and Sohoni [55, 56], approaches the lower bound question from a different angle. GCT also looks for an algebraic lower bound proof, but rather than exhibiting an algebraic argument, it aims to prove the existence of a separating polynomial. Specifically, GCT attempts to prove Valiant's hypothesis, that VP≠VNP, over $\mathbb{C}$, via *representation theory*. Valiant's hypothesis is, more or less, equivalent to showing that the permanent of a symbolic $n \times n$ matrix is not a *projection* of the symbolic $m \times m$ determinant for any $m = m(n)$ polynomial in n.[2] Recall that a projection of a polynomial is a restriction of the polynomial to an affine subspace of its inputs. Observe that a restriction of an n-variate polynomial $f(\boldsymbol{x})$ to a subspace of its inputs, is equivalent to considering the polynomial $f(A\boldsymbol{x} + \boldsymbol{b})$, where A is an $n \times n$ matrix and $\boldsymbol{b} \in \mathbb{C}^n$. As any matrix is a limit point of a sequence of invertible matrices, an algebraic proof that the permanent is not a projection of the $m \times m$ determinant, over $\mathbb{C}$, is equivalent to an algebraic proof showing that the permanent is not in the closure of the set of polynomials $\{\mathrm{Det}(AX + \boldsymbol{b}) \mid A \in \mathrm{GL}_m(\mathbb{C}),\ \boldsymbol{B} \in \mathbb{C}^{m^2}\}$, where $\mathrm{GL}_m(\mathbb{C})$ is the group of invertible $m \times m$ matrices (this is true for every field of characteristic $\neq 2$). The set $\{\mathrm{Det}(AX + \boldsymbol{b}) \mid A \in \mathrm{GL}_m(\mathbb{C}),\ \boldsymbol{B} \in \mathbb{C}^{m^2}\}$ is called the *orbit* of the determinant under the action of the affine group (we denote the affine group over $\mathbb{C}^m$ with $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$). GCT considers the linear space of polynomials that vanish on every coefficient vector in the orbit of the determinant, and similarly the linear space of polynomials that vanish on every coefficient vector in the orbit of the permanent. There is a natural action of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$ on those linear spaces, thus defining two representations of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$. GCT wishes

[1] It is closed in the Zariski topology. Over $\mathbb{R}$ or $\mathbb{C}$ this is the same as being closed in the Euclidean topology.

[2] A super-quasipolynomial lower bound would imply that VP≠VNP whereas a super-polynomial lower bound would imply that permanent does not have polynomial-size algebraic formulas or algebraic branching programs.

to find a separating polynomial by showing that some irreducible representation of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$ has strictly larger multiplicity when considering the representation corresponding to the determinant. This approach bypasses the barrier given in [28, 30] as it does not exhibit any efficiently computable separating polynomial but rather just proves the existence of one. However, the representation theory questions arising in this program are quite difficult, even when considering the analog questions for restricted classes. For an introduction to GCT see the lecture notes of Bläser and Ikenmeyer [13].

Another possible approach for proving lower bounds against a class of polynomials $\mathcal{C}$, is via the construction of a *hitting set* for $\mathcal{C}$. Recall that a hitting set $\mathcal{H}$ for a class $\mathcal{C}$ is a set of points such that for any nonzero polynomial f, that can be computed by a circuit from $\mathcal{C}$, there is $\boldsymbol{v} \in \mathcal{H}$ such that $f(\boldsymbol{v}) \neq 0$. In [37] Heintz and Schnorr observed that if we have such a hitting set $\mathcal{H}$ then any nonzero polynomial g that vanishes on $\mathcal{H}$ cannot be computed in $\mathcal{C}$. It is also not hard to see that this way of obtaining lower bounds also bypasses the natural proof barrier of [28, 30]. The problem is that in most cases we obtained a hitting set for a class only after proving a lower bound for it.

In [26] Forbes and Shpilka defined the notion of a *robust* hitting set for a circuit class $\mathcal{C}$. Over fields of characteristic zero, a hitting set $\mathcal{H}$ for a class $\mathcal{C}$ is c-robust if it also satisfies that for every $f \in \mathcal{C}$ there is $\boldsymbol{v} \in \mathcal{H}$ such that $|f(\boldsymbol{v})| \geq c \cdot \|f\|$, where $\|\cdot\|$ is some fixed norm on $\mathbb{C}[\boldsymbol{x}]$ (see Definition 13 for a definition over arbitrary fields). It is not hard to see that if $\mathcal{H}$ is a robust hitting set for a class $\mathcal{C}$ then it also hits the closure of $\mathcal{C}$.

In this work we focus on depth-3 algebraic circuits, known as $\Sigma\Pi\Sigma$, and on $\mathrm{VP_e}$, the class of algebraic formulas, two classes for which we lack strong lower bounds, and in particular we do not have hitting sets for them. For $\Sigma\Pi\Sigma$ circuits the best lower bound is the near cubic lower bound of Kayal, Saha and Tavenas [46], and for $\mathrm{VP_e}$ the best lower bound is the quadratic lower bound of Kalarkoti [39]. Recall that by the result of Valiant et al. [71], a super-quasipolynomial lower bound against $\mathrm{VP_e}$ implies a super-polynomial lower bound against VP. Similarly, a hitting set for $\mathrm{VP_e}$ implies a hitting set for VP. We also note that by a result of Gupta et al. [33], a strong enough lower bound or a hitting set for $\Sigma\Pi\Sigma$ imply both a lower bound for general circuits and a hitting set for them. This result also implies that a polynomial-time reconstruction algorithm for $\Sigma\Pi\Sigma$ circuits would give rise to a sub-exponential time *reconstruction algorithm* for general circuits. Recall that a reconstruction algorithm for a class $\mathcal{C}$ is an algorithm that, given black-box access to a circuit from $\mathcal{C}$, outputs a circuit in $\mathcal{C}$ that computes the same polynomial.

Instead of viewing robust hitting sets as a way to obtain hitting sets for the closure of circuit classes, we suggest to find subclasses of interesting classes, $\tilde{\mathcal{C}} \subset \mathcal{C}$, such that $\mathcal{C}$ is contained in the closure of $\tilde{\mathcal{C}}$, and aim to construct a robust hitting set for the subclass $\tilde{\mathcal{C}}$. This offers a new approach for constructing hitting sets for known classes and for obtaining lower bounds. Specifically, we consider subclasses of $\Sigma\Pi\Sigma$ and $\mathrm{VP_e}$ that are dense in their superclasses. Each of these subclasses is the orbit of some simple polynomial under the group of invertible affine transformations.

For $\mathrm{VP_e}$, we first consider a subclass that was defined by Bringmann, Ikenmeyer and Zuiddam [14]–the orbit of the so called *continuant* polynomial (see Definition 27). We give a polynomial-sized interpolating set[3] for this subclass as well as a polynomial-time

[3] Recall that an interpolating set for a class $\mathcal{C}$ of polynomials in n variables, over a field $\mathbb{F}$, is a set of points $\mathcal{H} \subset \mathbb{F}^n$ such that for every $f \in \mathcal{C}$, the list of values $f(\mathcal{H})$ uniquely determines f. See Definition 15.

deterministic reconstruction algorithm that uses as oracle a *root-finding algorithm.*[4] In particular, this implies a polynomial-time randomized reconstruction algorithm, and, in some cases, a polynomial-time deterministic algorithm.

In addition, we exhibit two other subclasses that are dense in $\mathrm{VP_e}$. The first class is defined as the orbit of read-once formulas (ROF for short, see Definition 5) and the second as the orbit of read-once formulas in *alternating normal form* (ROANF for short, see Definition 7). We obtain hitting sets for both classes and an interpolating set for the second. We also observe that the reconstruction algorithm of [35] works for the polynomials in the orbit of ROANFs. Although the results that we obtain for the subclass defined by the continuant polynomial are stronger, we think that every such dense subclass can shed more light on $\mathrm{VP_e}$ and may eventually be used in order to obtain new lower bounds.

For $\Sigma\Pi\Sigma$ we consider two subclasses. One is based on orbits of *sparse* polynomials (polynomials having polynomially many monomials) and the other on orbits of *diagonal* tensors (see Definition 40). We give a hitting set for the first, an interpolation set for the second, and we also observe that a slight modification of the randomized reconstruction algorithm of [43] applies for the second class.

In particular, our results give the first dense subclasses inside $\mathrm{VP_e}$ and $\Sigma\Pi\Sigma$ for which a polynomial-size interpolating set is known as well as a polynomial-time reconstruction algorithm. By [71] our result immediately translate to VP, giving a dense subclass of for which a quasipolynomial-sized interpolating set is known as well as a quasipolynomial-time reconstruction algorithm.

If we could transform the interpolating sets that we have found to *robust hitting sets* for the orbits, then this will immediately give hitting sets for the closure of the orbits, i.e. for $\Sigma\Pi\Sigma$ and $\mathrm{VP_e}$, which, by [37] gives a lower bound for the class. Thus, our work raises an intriguing problem:

▶ **Problem 1.** *Given an interpolating set for a class $\mathcal{C}$ construct a robust hitting set for $\mathcal{C}$.*

We stress that by our results, solving this problem would lead to hitting sets, and lower bounds, for $\mathrm{VP_e}$ and VP.

Another advantage for having small interpolating sets for dense subclasses is the following: One approach for searching for separating polynomials for a class, is by considering the map from circuits in the class to the coefficient vectors of the polynomials that they compute. That is, once we fix a computation graph, an assignment to the constants appearing in the circuit determines the output polynomial. Each coefficient is a polynomial in those constants, and as there are "few" constants (polynomially many for polynomially sized circuits), and there are exponentially many coefficients, there should be many polynomials vanishing on the closure of the image of this map. If we could get a good understanding of this map then perhaps we could use it to construct a polynomial that vanishes on all such coefficient vectors. This polynomial will vanish on all coefficient vectors of the superclass in which the subclass is dense. A different approach is to find a coefficient vector that is not in the closure of the image of this map (this is the approach of Raz in [57]). Now, assume that $\mathcal{H}$ is an interpolating set for a dense subclass $\tilde{\mathcal{C}} \subset \mathcal{C}$. We know that the map $f \to f\big|_{\mathcal{H}}$ is one-to-one on $\tilde{\mathcal{C}}$. Thus, the list of values $f\big|_{\mathcal{H}}$ can be viewed as an efficient encoding that is given in terms of values of the computed polynomial. This provides a different encoding of a circuit – instead of the constants in it, use the evaluations on $\mathcal{H}$. Thus, by studying the closure of

[4] A root-finding algorithm, over a field $\mathbb{F}$, when given black-box access to a univariate polynomial, outputs a root of that polynomial in $\mathbb{F}$, if such a root exists.

this map (i.e. the closure of the set of points on $\mathbb{F}^{|\mathcal{H}|}$ that can be obtained as evaluation vectors of polynomials in the subclass) we may be able to find a separating polynomial, or, as in Raz's approach, find an evaluation vector that is not obtained by any polynomial in the superclass. It is clear that one can also try this approach even if $\mathcal{H}$ is not an interpolating set, however, as interpolating sets "preserve information" of a dense set, we believe that such sets are better suited for this approach.

To conclude, focusing on dense subclasses and studying their properties could lead to better understanding of their superclasses and perhaps to breakthrough results in algebraic complexity.

To formally state our results we need some definitions that we give next.

1.1 Basic definitions

1.1.1 Notation

For $k \in \mathbb{N}$, we denote $[k] \triangleq \{1, 2, 3, \dots, k\}$ and $[k]_0 \triangleq \{0, 1, 2, \dots, k-1\}$. We use boldface lowercase letters to denote tuples of variables or vectors, as in $\boldsymbol{x} = (x_1, \dots, x_n)$, $\boldsymbol{a} = (a_1, \dots, a_m)$, when the dimension is clear from the context. For any two elements i, j coming from some set S (usually i and j will be numbers), $\delta_{i,j}$ equals 1 when $i = j$ and 0 otherwise.

The individual degree of a variable x_i in $f(\boldsymbol{x})$ is the degree of f as a polynomial in x_i. A polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ of $\deg(f) \leq 1$ is called a linear function, and if f is homogeneous then it is called a *linear form*. For a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ and an integer $k \in \mathbb{N}$ we denote by $f^{[k]}$ the degree-k homogeneous part of $f(\boldsymbol{x})$,i.e. the sum of all monomials of f of degree exactly k. In particular,

$$f(\boldsymbol{x}) = f^{[0]}(\boldsymbol{x}) + f^{[1]}(\boldsymbol{x}) + \ldots + f^{[\deg(f)]}(\boldsymbol{x}) \ .$$

Note that for a linear function f, $f^{[1]}$ is a linear form. We say that a polynomial f is homogeneous of degree k or that f is k-homogeneous if $f = f^{[k]}$. We say a set of linear functions $\{\ell_1(\boldsymbol{x}), \dots, \ell_n(\boldsymbol{x})\} \subset \mathbb{F}[\boldsymbol{x}]$ is *linearly independent* if the set $\left\{\ell_i^{[1]}\right\}$ is linearly independent.[5] Given a polynomial $f(\boldsymbol{x})$, a subset of variables $\boldsymbol{y} \subseteq \{x_1, \dots, x_n\}$ and an assignment to those variables $\boldsymbol{a} \in \mathbb{F}^{|\boldsymbol{y}|}$, we denote by $f\big|_{\boldsymbol{y}=\boldsymbol{a}} \in \mathbb{F}[\boldsymbol{x} \setminus \boldsymbol{y}]$ the polynomial resulting from assigning the values of $\boldsymbol{a}$ to the variables of $\boldsymbol{y}$ in $f(\boldsymbol{x})$. We sometimes abuse notation and write $\boldsymbol{y} \subseteq [n]$ to indicate the indices of the assigned variables instead of the variables themselves.

1.1.2 Circuit classes

▶ **Definition 2.** *An algebraic formula (also called arithmetic formula) over a field $\mathbb{F}$, is a rooted tree whose leaves are labeled with either variable or scalars from $\mathbb{F}$, and whose root and internal nodes (called gates) are labeled with either "$+$" (addition) or "$\times$" (multiplication). An algebraic formula computes a polynomial in the natural way. Each leaf computes the polynomial that labels it, and each gate computes either the sum or product of its children, depending on its label. The output of the formula is the polynomial computed at its root. The size of a formula is the number of wires in it. The depth of a formula is the length of the longest simple leaf-root path in it. The formula size of a polynomial f is defined as the smallest size of a formula that outputs f.*

[5] Note that by our definition, x and $x + 1$ are linearly dependent.

A sequence $m(n)$ of natural numbers is called polynomially bounded if there exists a univariate polynomial q such that $m(n) \leq q(n)$ for all n.

The complexity class $\mathrm{VP_e}$ is defined as the set of all families of polynomials $(f_n)_n$, with $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, whose formula size is polynomially bounded.

▶ **Definition 3.** *An arithmetic circuit Φ is a $\Sigma^{[s]}\Pi^{[d]}$ circuit if it is a layered graph of depth-2, has a top gate labeled $+$ with fan-in $\leq s$ and its second layer is comprised entirely of $\times$ gates with fan-in $\leq d$. In other words, $\Sigma^{[s]}\Pi^{[d]}$ compute polynomials of degree d with at most s monomials.*

▶ **Definition 4.** *An arithmetic circuit Φ in n variables is a $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit if it is a layered graph of depth-3, has a top gate labeled $+$ with fan-in $\leq s$, its second layer is comprised entirely of $\times$ gates with fan-in $\leq d$, and its bottom layer is comprised of linear functions in $x_1, \ldots, x_n$. In other words, $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit compute polynomials of the form*

$$f(\boldsymbol{x}) = \sum_{i=1}^{s} \prod_{j=1}^{d} (\alpha_{i,j,0} + \sum_{k=1}^{n} \alpha_{i,j,k} x_k) \ .$$

Given a family of circuits $\mathcal{C}$, we will sometime denote it as $\mathcal{C}(\mathbb{F})$ to stress that we allow coefficients to come from the field $\mathbb{F}$. Observe that the definitions of the classes above do not depend on the field and so we can define them over any field of our choice.

▶ **Definition 5.** *An* arithmetic read-once formula *(ROF for short) Φ over a field $\mathbb{F}$ in the variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a binary tree T whose leaves are labeled with input variables and a pairs of field elements $(\alpha, \beta) \in \mathbb{F}^2$, and whose internal nodes are labeled with the arithmetic operations $\{+, \times\}$ and a field element $\alpha \in \mathbb{F}$. Each input variable can label at most one leaf. The computation is performed in the following way: A leaf labeled with the variable x_i and with (α, β), computes the polynomial $\alpha x_i + \beta$. If a node v is labeled with the operation $* \in \{+, \times\}$ and with $\alpha \in \mathbb{F}$, and its children compute the polynomials Φ_{v_1} and Φ_{v_2}, then the polynomial computed at v is $\Phi_v = \Phi_{v_1} * \Phi_{v_2} + \alpha$. A polynomial $f(\boldsymbol{x})$ is called a* read-once polynomial *(ROP for short) if $f(\boldsymbol{x})$ can be computed by a ROF.*

▶ **Observation 6.** *Read-once polynomials are always multilinear polynomials.*

We next define formulas in alternating normal form, as was first defined in [35].

▶ **Definition 7** (Section 3.2 in [35]). *We say that an arithmetic formula Φ, over $\mathbb{F}$, is in* alternating normal form *(Φ is called an* ANF *for short) if:*

1. *The underlying tree of Φ is a complete rooted binary tree (the root node is called the output node). In particular, $\mathrm{size}(\Phi) = 2^{\mathrm{depth}(\Phi)+1} - 1$, where $\mathrm{size}(\Phi)$ is the number of nodes in the tree of Φ and $\mathrm{depth}(\Phi)$ is the maximum distance of a leaf node from the output node of Φ.*
2. *The internal nodes consist of alternating layers of $+$ and $\times$ gates. In particular, the label of an internal node at distance d from the closest leaf node is $+$ if d is even and $\times$ otherwise. So if the root node is a $+$ node, its children are all $\times$ nodes, its grandchildren are all $+$ etc.*
3. *The leaves of the tree are labeled with linear functions. That is, each leaf is labeled with $\ell(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i$, where each $a_i \in \mathbb{F}$ is a scalar.*

The product depth *Δ of Φ is the number of layers of product gates. The number of leaves of Φ is therefore always 4^{Δ} if the top gate is $+$, and $\frac{1}{2} \cdot 4^{\Delta}$ if the top gate is $\times$.*

The class $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ mentioned in Section 1.2.2 is defined in terms of the following canonical read-once ANF formula (ROANF for short):

▶ **Definition 8** (Notation from Fact 3.4 of [35]). *We denote the canonical ROANF polynomial, of product depth Δ on 4^{Δ} variables, as $ANF_{\Delta}(\boldsymbol{x})$. It is defined recursively as follows:*

$$ANF_0(\boldsymbol{x}) = x_1$$
$$ANF_{\Delta+1}(\boldsymbol{x}) = ANF_{\Delta}\left(\boldsymbol{x}^{(1)}\right) ANF_{\Delta}\left(\boldsymbol{x}^{(2)}\right) + ANF_{\Delta}\left(\boldsymbol{x}^{(3)}\right) ANF_{\Delta}\left(\boldsymbol{x}^{(4)}\right) ,$$

where $\boldsymbol{x}^{(i)}$ is the 4^{Δ}-tuple of variables $\{x_{(i-1)\cdot 4^{\Delta}+1}, \ldots, x_{i\cdot 4^{\Delta}}\}$.

For example, $\mathrm{ANF}_1(\boldsymbol{x}) = x_1x_2 + x_3x_4$.

Observe that any polynomial in $\mathrm{ANF}_{\Delta}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ is an ANF according to Definition 7, but not vice versa.

1.1.3 Approximate complexity

The following definition gives sense to the notion of approximation over arbitrary fields. In what follows we let ε be a new formal variable.[6] For a field $\mathbb{F}$ we denote with $\mathbb{F}[\varepsilon]$ the ring of polynomial expressions in ε over $\mathbb{F}$, and with $\mathbb{F}(\varepsilon)$ the fraction field of $\mathbb{F}[\varepsilon]$, i.e. the field of rational expressions in ε.

▶ **Definition 9.** *Let $\mathcal{C}(\mathbb{F})$ be a circuit class over a field $\mathbb{F}$. The closure of $\mathcal{C}$, denoted $\overline{\mathcal{C}(\mathbb{F})}$, is defined as follows: A family of functions $(f_n)_n$, where $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, is in $\overline{\mathcal{C}(\mathbb{F})}$ if there is a polynomially bounded function $m : \mathbb{N} \to \mathbb{N}$, and a family of functions $(g_{m(n)})_n \in \mathcal{C}(\mathbb{F}(\varepsilon))$, with $g_{m(n)} \in \mathbb{F}[\varepsilon][x_1, \ldots, x_{m(n)}]$, such that for all $n \in \mathbb{N}$,*

$$g_{m(n)}(x_1, \ldots, x_{m(n)}) = f_n(x_1, \ldots, x_n) + \varepsilon \cdot g_{n,0}(x_1, \ldots, x_{m(n)}) , \tag{1}$$

for some polynomial $g_{n,0} \in \mathbb{F}[\varepsilon][x_1, \ldots, x_{m(n)}]$. Whenever an equality as in (1) holds we say that

$$g_{m(n)} = f_n + O(\varepsilon) \quad \text{or} \quad f_n = g_{m(n)} + O(\varepsilon) .$$

In that case we think of $g_{m(n)}$ as an "approximation" of f_n, and we say that the family $(g_{m(n)})_n$ approximates the family $(f_n)_n$.

Alder [3] have shown that over $\mathbb{C}$ it holds that $(f_n) \in \overline{\mathcal{C}(\mathbb{C})}$, in the sense of Definition 9, if and only if it is in the closure of $\mathcal{C}(\mathbb{C})$ in the usual sense. That is, if for every n there exists a sequence of polynomials $g_{n,k} \in \mathcal{C}(\mathbb{C})$ such that $\lim_{k\to\infty} g_{n,k} = f_n$, where convergence is taken coefficient wise. This result holds over $\mathbb{R}$ as well, see [52, 17].

Finally, we note that every matrix is approximable (in the sense of Definition 9) by a non-singular matrix (which is equivalent to being a limit of a sequence of non-singular matrices, in characteristic zero).

▶ **Observation 10.** *For every $A \in \mathbb{F}^{n\times n}$ there exists a non-singular matrix $B \in \mathbb{F}(\varepsilon)^{n\times n}$ such that $A = B + O(\varepsilon)$.*

[6] Intuitively, one should think of ε as an infinitesimal quantity.

1.1.4 Hitting and interpolating sets

▶ **Definition 11.** *A set of points $\mathcal{H} \subseteq \mathbb{F}^n$ is called a* hitting set *for a circuit class $\mathcal{C}$ (we also say that $\mathcal{H}$* hits *$\mathcal{C}$) if for every circuit $\Phi \in \mathcal{C}$, computing a non-zero polynomial, there exists some $\boldsymbol{a} \in \mathcal{H}$ such that $\Phi(\boldsymbol{a}) \neq 0$.*

We next give the definition of a robust hitting set, a notion first defined in [26]. Here we extend the definition for arbitrary characteristic. We start by giving the definition of [26], over characteristic zero (and focus on $\mathbb{C}$) and then the more general definition.

▶ **Definition 12** (Following Definition 5.1 of [26])**.** *Let $\|\cdot\|$ be some norm on $\mathbb{C}[\boldsymbol{x}]$. A hitting set $\mathcal{H}$ for a circuit class $\mathcal{C} \subseteq \mathbb{C}[\boldsymbol{x}]$ is called* robust *if there exists some constant $c > 0$ such that, for every $0 \neq f \in \mathcal{C}$,*[7] *there exists some $\boldsymbol{a} \in \mathcal{H}$ such that $|f(\boldsymbol{a})| \geq c \cdot \|f\|$.*

For arbitrary characteristic we use the same approach as in Definition 9.

▶ **Definition 13.** *Let $\mathbb{F}$ be a field of arbitrary characteristic. A hitting set $\mathcal{H} \subset \mathbb{F}^n$ for a circuit class $\mathcal{C}(\mathbb{F})$ is called* robust *if for every circuit $\Phi \in \mathcal{C}(\mathbb{F}(\varepsilon))$ computing a polynomial $f(\boldsymbol{x}) = h(\boldsymbol{x}) + \varepsilon \cdot g(\boldsymbol{x})$, where $h(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ and $g(\boldsymbol{x}) \in \mathbb{F}[\varepsilon][\boldsymbol{x}]$, there exists some $\boldsymbol{a} \in \mathcal{H}$ such that $f(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$.*

It is not hard to prove using the result of [3] that for $\mathbb{F} = \mathbb{C}$, Definitions 12 and 13 are equivalent.

▶ **Observation 14.** *If $\mathcal{H}$ is a finite robust hitting set for $\mathcal{C}(\mathbb{F})$, then $\mathcal{H}$ hits $\overline{\mathcal{C}(\mathbb{F})}$ as well.*

Proof. Consider $0 \neq f \in \overline{\mathcal{C}(\mathbb{F})}$. By Definition 9 there is $g \in \mathcal{C}(\mathbb{F}(\varepsilon))$, such that $f = g + O(\varepsilon)$. Clearly $g \neq 0$. Let $\boldsymbol{a} \in \mathcal{H}$ be such that $g(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$. It follows that $f(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$. In particular, $f(\boldsymbol{a}) \neq 0$. ◀

We next define the notion of an interpolating set.

▶ **Definition 15.** *Let $\mathcal{C}$ be a class of n-variate polynomials. A set $\mathcal{H} \subseteq \mathbb{F}^n$ is called an* interpolating set *for $\mathcal{C}$ if, for every $f \in \mathcal{C}$, the evaluations of f on $\mathcal{H}$ uniquely determine f.*

▶ **Observation 16.** *If $\mathcal{H}$ is a hitting set for $\mathcal{C}(\mathbb{F}) + \mathcal{C}(\mathbb{F}) \triangleq \{\alpha f + \beta g : f, g \in \mathcal{C}, \alpha, \beta \in \mathbb{F}\}$, then $\mathcal{H}$ is an interpolating set for $\mathcal{C}$.*

A common method for designing hitting and interpolating sets is via hitting set generators.

▶ **Definition 17.** *A polynomial mapping $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ is called a* hitting set generator *(or simply a generator) for a circuit class $\mathcal{C}(\mathbb{F})$ if for any non-zero n-variate polynomial $f \in \mathcal{C}$, the k-variate polynomial $f \circ \mathcal{G}$ is non-zero.*

Similarly, we call $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ an interpolating set generator *for a circuit class $\mathcal{C}(\mathbb{F})$ if for any two different n-variate polynomials $f_1, f_2 \in \mathcal{C}$, the k-variate polynomial $(f_1 - f_2) \circ \mathcal{G}$ is non-zero.*

Generators immediately give rise to hitting sets.

▶ **Observation 18.** *Let $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ be a generator for $\mathcal{C}(\mathbb{F})$ such that the individual degree of each coordinate of $\mathcal{G}$ is at most r. Let $W \subset \mathbb{F}$ be any set of size $|W| = d \cdot r + 1$. Let $\mathcal{H} = \mathcal{G}\left(W^k\right)$. Then $\mathcal{H}$ hits every n-variate polynomial $f \in \mathcal{C}$ of degree at most d.*

Proof. As $\mathcal{G}$ is a generator, the k-variate polynomial $f \circ \mathcal{G}$ is nonzero. As its individual degrees are bounded by $d \cdot r$ it follows that at least one of the values in $(f \circ \mathcal{G})\left(W^k\right) = f(\mathcal{H})$ is not zero. ◀

[7] We abuse notation and write $f \in \mathcal{C}$ when f is the output of some circuit from $\mathcal{C}$.

1.1.5 k-independent maps

Our constructions rely on polynomial mappings $\mathcal{G}_k$, parameterized by some integer $k \le n$, with the property that the image of $f \circ \mathcal{G}_k$ contains all projections of f to k variables. We call such a map a k-independent map.

▶ **Definition 19.** *We call a polynomial mapping $\mathcal{G}(y_1, \dots, y_t, z_1) : \mathbb{F}^{t+1} \to \mathbb{F}^n$ a* 1-independent polynomial map *if for every index $i \in [n]$ there exists an assignment $\boldsymbol{a}_i \in \mathbb{F}^t$ to $y_1, \dots, y_t$ such that the ith coordinate of $\mathcal{G}(\boldsymbol{a}_i, z_1)$ is z_1, and the rest of the coordinates are 0. For $k > 1$, a polynomial mapping $\mathcal{G}(y_1, \dots, y_{tk}, z_1, \dots, z_k) : \mathbb{F}^{k(t+1)} \to \mathbb{F}^n$ is called a* k-independent polynomial map *(or a k-independent map) if $\mathcal{G}$ is a sum of k variable-disjoint 1-independent polynomial maps. We denote k-independent polynomial maps as $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ when k, t are implicit. The $\boldsymbol{y}$ variables are called* control variables.

A k-independent polynomial map $\mathcal{G}$ is called uniform *if all n coordinates of $\mathcal{G}$ are homogeneous polynomials of the same degree.*

We discuss k-independent maps in more detail in Section 2.

1.1.6 Subgroups of the linear and affine groups and their actions

Given a matrix $A \in \mathbb{F}^{n \times n}$ and a tuple of variables $\boldsymbol{x} = (x_1, \dots, x_n)$, we denote

$$A\boldsymbol{x} = \left(\sum_{i=1}^{n} A_{1,i} x_i, \sum_{i=1}^{n} A_{2,i} x_i, \dots, \sum_{i=1}^{n} A_{n,i} x_i \right) .$$

Let $n \ge m \in \mathbb{N}$. For an m-variate polynomial $f(x_1, \dots, x_m) \in \mathbb{F}[x_1, \dots, x_m]$, a matrix $A = (A_{i,j})_{i,j=1}^{n} \in \mathbb{F}^{n \times n}$ and a vector $\boldsymbol{b} = (b_1, \dots, b_n) \in \mathbb{F}^n$, we define the n-variate polynomial $f(A\boldsymbol{x} + \boldsymbol{b})$ to be

$$f(A\boldsymbol{x} + \boldsymbol{b}) \triangleq f\left(\sum_{i=1}^{n} A_{1,i} x_i + b_1, \sum_{i=1}^{n} A_{2,i} x_i + b_2, \dots, \sum_{i=1}^{n} A_{m,i} x_i + b_m \right) . \qquad (2)$$

Note that we ignored the last $n - m$ coordinates of $A\boldsymbol{x} + \boldsymbol{b}$.

We denote with $\mathrm{GL}_n(\mathbb{F})$ the group of invertible $n \times n$ matrices over $\mathbb{F}$, and with $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ the group of invertible affine transformation, i.e. all the maps $\boldsymbol{x} \to A\boldsymbol{x} + \boldsymbol{b}$, where $A \in \mathrm{GL}_n(\mathbb{F})$ and $\boldsymbol{b} \in \mathbb{F}^n$.

For an m-variate polynomial f over $\mathbb{F}$, and $n \ge m$ we denote with $f^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ the orbit of f under the natural action of $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$:[8]

$$f^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \triangleq \{ f(A\boldsymbol{x} + \boldsymbol{b}) \mid A \in \mathrm{GL}_n(\mathbb{F}),\ \boldsymbol{b} \in \mathbb{F}^n \} .$$

We similarly define $f^{\mathrm{GL}_n(\mathbb{F})}$. More generally, for a class of m-variate polynomials $\mathcal{C}(\mathbb{F})$, we denote the *orbit* of $\mathcal{C}$ under $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ by

$$\mathcal{C}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \triangleq \{ f(A\boldsymbol{x} + \boldsymbol{b}) \mid f \in \mathcal{C},\ A \in \mathrm{GL}_n(\mathbb{F}),\ \boldsymbol{b} \in \mathbb{F}^n \} .$$

We similarly define $\mathcal{C}^{\mathrm{GL}_n(\mathbb{F})}$. When we want to speak about orbits of families of polynomials from $\mathcal{C}(\mathbb{F})$, with arbitrary number of variables, we use the notation $\mathcal{C}^{\mathrm{GL}(\mathbb{F})}$ or $\mathcal{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

[8] To be precise, the action is $((A, \boldsymbol{b}) \circ f)(\boldsymbol{x}) = f(A^T \boldsymbol{x} + \boldsymbol{b})$. This is required in order to make the action a homomorphism, however, for the groups that we consider it does not change the orbit.

▶ **Observation 20.** *For any m variate polynomial $f(x_1,\dots,x_m)$ and $n \geq m$:*

- *For any $A \in GL_n(\mathbb{F})$ and $d \in \mathbb{N}$, $f^{[d]}(A\boldsymbol{x})$ is the d-homogeneous part of $f(A\boldsymbol{x})$.*
- *For any $A \in GL_n^{aff}(\mathbb{F})$, $f(\boldsymbol{x})$ is irreducible if and only if $f(A\boldsymbol{x})$ is irreducible.*
- *The set of matrices A for which $f(\boldsymbol{x}) = f(A\boldsymbol{x})$ forms a multiplicative subgroup of $GL_n(\mathbb{F})$ and a similar claim holds for $GL_n^{aff}(\mathbb{F})$.*

We next define some special groups that serve as group of symmetries of some of the models that we consider. We first define the group of symmetries of $\mathrm{ANF}_\Delta(\boldsymbol{x})$. We denote with I_k the $k \times k$ identity matrix.

▶ **Definition 21.** *For $m, \Delta \in \mathbb{N}$ such that $m = 2^\Delta$, the* tree-symmetry group *$TR_m(\mathbb{F})$ denotes the automorphisms of a rooted complete binary tree of depth Δ. It is defined recursively as follows.*

- *For $m = 1$, $TR_1(\mathbb{F})$ consists only of the identity matrix.*
- *For $m > 0$, $TR_m(\mathbb{F})$ is generated by matrices of the form*

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & I_{\frac{m}{2}} \\ I_{\frac{m}{2}} & 0 \end{pmatrix}$$

where $A, B \in TR_{\frac{m}{2}}(\mathbb{F})$.

▶ **Definition 22.** *For any $m = 4^\Delta$, the* tree-scale group *$TS_m(\mathbb{F})$ is the group generated by elements of $TR_m(\mathbb{F})$ and matrices of the form*

$$\begin{pmatrix} \alpha I_{\frac{m}{4}} & 0 & 0 & 0 \\ 0 & \alpha^{-1} I_{\frac{m}{4}} & 0 & 0 \\ 0 & 0 & \beta I_{\frac{m}{4}} & 0 \\ 0 & 0 & 0 & \beta^{-1} I_{\frac{m}{4}} \end{pmatrix}$$

where $0 \neq \alpha, \beta \in \mathbb{F}$.

The importance of the group $\mathrm{TS}_m(\mathbb{F})$ stems from the fact that it is the symmetry group of ANF_Δ. To intuitively see why this is the case, notice that in any representation of an ANF one may swap children of any node without changing the output polynomial. We call such symmetries "tree-symmetries" and they are captured by the group $\mathrm{TR}_n(\mathbb{F})$. A second source of ambiguity comes from the fact that we can rescale the formula. Recall that the output polynomial is of the form $f_1 \cdot f_2 + f_3 \cdot f_4$ (Definition 7). Clearly, the output does not change if we replace f_1 by, say, $2f_1$ and f_2 by $f_2/2$. Such rescaling symmetries are captured by the group $\mathrm{TS}_n(\mathbb{F})$. Finally, another source for ambiguity comes from the fact that the quadratic polynomials computed at the bottom two layers of the ANF may have different representations. For example,

$$4xy + 4wz = (x + y + w - z) \cdot (x + y - w + z) + (w + z + x - y) \cdot (w + z - x + y) .$$

As there is an infinite number of representations for each quadratic polynomial (over infinite fields), we can expect to characterize the symmetries in term of the quadratics computed at the bottom two layers of the ANF.

▶ **Fact 23** (Special case of Theorem 5.43(iii) of [35]). *Let $m, \Delta, n \in \mathbb{N}$ such that $m = 4^{\Delta-1} \leq n/4$. Let $f = ANF_\Delta(\ell_1,\dots,\ell_{4m}) \in ANF_\Delta^{GL_n^{aff}(\mathbb{F})}$. Let $Q = (q_1,\dots,q_m)$ be the list of quadratic polynomials that are computed at the bottom two layers of the formula $ANF_\Delta(\ell_1,\dots,\ell_{4m})$. In particular, $f = ANF_{\Delta-1}(q_1,\dots,q_m)$. If $Q' = (q'_1,\dots,q'_m)$ is any other m-tuple of quadratic polynomials for which $f = ANF_{\Delta-1}(q'_1,\dots,q'_m)$ then Q is $TS_m(\mathbb{F})$-equivalent to Q'.*

Next, we define the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$.

▶ **Definition 24.** *For any $n \in \mathbb{N}$ the* permutation-scale group, *denoted $PS_n(\mathbb{F})$, is the set of all matrices $A \in GL_n(\mathbb{F})$ which are row-permutations of non-singular diagonal matrices with determinant one.*

For example, $\begin{pmatrix} 0 & -2 & 0 \\ 0 & 0 & -1 \\ 1/2 & 0 & 0 \end{pmatrix} \in \mathrm{PS}_3(\mathbb{C})$.

▶ **Definition 25.** *Let $s, d, n \in \mathbb{N}$ such that $n = s \cdot d$. A matrix $A \in GL_n(\mathbb{F})$ is a member of the* tensor permutation-scale group, *denoted $TPS_{s,d}(\mathbb{F})$, if $A = (P \otimes I_d) \cdot B$, where P is an $s \times s$ permutation matrix and $B = \begin{pmatrix} B_1 & 0 & \dots & 0 \\ 0 & B_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & B_d \end{pmatrix}$ is a block diagonal matrix such that each block B_i of B satisfies $B_i \in PS_d(\mathbb{F})$.*

For example, for $s = d = 2$ the matrix $A = \begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 1/2 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$ is in $\mathrm{TPS}_{2,2}(\mathbb{C})$, as for $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 1/2 & 0 \end{pmatrix}$, we have $A = (P \otimes I_2) \cdot B$, and clearly each block of B is in $\mathrm{PS}_2(\mathbb{C})$.

Another way of defining the group is as follows: index rows and columns of A with pairs $(i, j) \in [s] \times [d]$. Then, $A \in \mathrm{TPS}_{s,d}(\mathbb{F})$ if and only if there exists a permutation $\pi : [s] \to [s]$, and for all $i \in [s]$ permutations $\theta_i : [d] \to [d]$ and constants $\alpha_{i,j}$ satisfying $\prod_{j=1}^{d} \alpha_{i,j} = 1$, such that $A_{(i,j),(i',j')} = \delta_{\pi(i),i'} \cdot \delta_{\theta_i(j),j'} \cdot \alpha_{i,j}$ for all i, j.

We next prove that $\mathrm{TPS}_{s,d}(\mathbb{F})$ is the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$. In other words, we show that $\mathrm{T}_{s,d}(\boldsymbol{x}) = \mathrm{T}_{s,d}(A\boldsymbol{x})$ if and only if $A \in \mathrm{TPS}_{s,d}(\mathbb{F})$. Intuitively, $\mathrm{T}_{s,d}$ admits no symmetries other than the trivial ones: permutations on the product gates, and internal permutation-scale of each product gate such that the product of the scale coefficients is 1. This is exactly captured by the group $\mathrm{TPS}_{s,d}(\mathbb{F})$, which is therefore contained in the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$.

▶ **Lemma 26.** *Let $s, d, n \in \mathbb{N}$, such that $d > 2$ and $n = s \cdot d$. If $A \in GL_n(\mathbb{F})$ satisfies $T_{s,d}(\boldsymbol{x}) = T_{s,d}(A\boldsymbol{x})$, then $A \in TPS_{s,d}(\mathbb{F})$.*

1.2 Our results

We first give our results for the class $\mathrm{VP_e}$ and then for the class of depth-3 circuits, for which it may be easier to obtain a robust hitting set, or prove super-polynomial lower bounds.

1.2.1 The continuant polynomial

Bringmann, Ikenmeyer and Zuiddam [14] defined the following polynomial (in Remark 3.14 of their paper), which they called the continuant polynomial:

▶ **Definition 27.** *The continuant polynomial on n variables, $C_n(x_1,\ldots,x_n)$, is defined as the* trace *of the following matrix product:*

$$C_n(x_1,\ldots,x_n) \triangleq Trace\left(\begin{pmatrix} x_1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_2 & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} x_n & 1 \\ 1 & 0 \end{pmatrix}\right) . \tag{3}$$

We denote with $C^{GL^{aff}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$ such that $f_n \in \mathbb{F}[x_1,\ldots,x_n]$ and for some $m \leq n$, $f_n \in C_m^{GL_n^{aff}(\mathbb{F})}$.

A result of Allender and Wang implies that the polynomial $x_1 \cdot y_1 + \cdots + x_8 \cdot y_8$ is not in $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ [4]. Thus, as a computational class it is very weak. However, Theorem 3.12 of [14] states that for every field $\mathbb{F}$ of characteristic different than 2, it holds that

$$\overline{\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}} = \overline{\mathrm{VP_e}} . \tag{4}$$

We give a polynomial-size interpolating set for the class $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ as well as a polynomial-time reconstruction algorithm for it. We first state a simple result that gives a hitting set for the class.

▶ **Theorem 28.** *Let $f(x_1,\ldots,x_n) \in C_m^{GL_n^{aff}(\mathbb{F})}$, for $m \leq n$, and arbitrary $\mathbb{F}$. Then, for any uniform 1-independent polynomial map $\mathcal{G}$ over $\mathbb{F}$, $f \circ \mathcal{G} \neq 0$.*

As immediate corollary we get a hitting set for the class.

▶ **Corollary 29.** *For every field $\mathbb{F}$, there is an explicit hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = O\left(n^6\right)$, that hits every $0 \neq f \in C_m^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \geq n^2$.*

▶ **Theorem 30.** *For every field $\mathbb{F}$, there is an explicit interpolating set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = O\left(n^{10}\right)$, for $\bigcup_{m=1}^{n} C_m^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \geq n^2$.*

▶ **Theorem 31.** *There is a deterministic algorithm that given $\mathbb{F}$, an integer n, oracle access to a root-finding algorithm over $\mathbb{F}$, and black-box access to a polynomial $f(x_1,\ldots,x_n) \in C_m^{GL_n^{aff}(\mathbb{F})}$ (for any $m \leq n$), runs in polynomial-time and outputs linear functions $(\ell_1(x_1,\ldots,x_n),\ldots,\ell_m(x_1,\ldots,x_n))$ such that*

$$f(x_1,\ldots,x_n) = C_m\left(\ell_1(\boldsymbol{x}),\ldots,\ell_m(\boldsymbol{x})\right) .$$

If $|\mathbb{F}| < n^3$ then the algorithm will make queries from a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$, such that $|\mathbb{K}| \geq n^3$, and it also requires oracle access to a root-finding algorithm over $\mathbb{K}$.

1.2.2 Orbits of read-once formulas

We denote with $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$, such that for every n there exists a ROF Φ, on $m \leq n$ variables, such that $f_n(x_1,\ldots,x_n) \in \Phi^{\mathrm{GL}_n(\mathbb{F})}$. Similarly, we denote with $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}[\mathbb{F}]}$ the class of families of polynomials $(f_n)_n$, such that for every n there exists Δ such that $4^\Delta \leq n$ and $f_n(x_1,\ldots,x_n) \in \mathrm{ANF}_\Delta^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$.

We first make the following simple observation.

▶ **Theorem 32.** *For every field $\mathbb{F}$, it holds that*

$$ANF^{GL^{aff}(\mathbb{F})} \subsetneq ROF^{GL(\mathbb{F})} \subsetneq VP_e(\mathbb{F}) . \tag{5}$$

However, when taking closures we get

$$\overline{ANF^{GL^{aff}(\mathbb{F})}} = \overline{ROF^{GL(\mathbb{F})}} = \overline{VP_e(\mathbb{F})} . \tag{6}$$

Our main results for ROFs and ROANFs are a construction of a hitting set for the orbit of ROFs, and an interpolating set for the orbit of ROANFs. Both constructions are obtained using independent polynomial maps (Definition 19).

▶ **Theorem 33.** *Let $0 \neq f \in ROF^{GL_n^{aff}(\mathbb{F})}$ where the underlying ROF depends on 2^t variables, for $2^t \leq n$. Then, for any $(t+1)$-independent polynomial map $\mathcal{G}$, over $\mathbb{F}$, $f \circ \mathcal{G} \neq 0$.*

▶ **Corollary 34.** *For every field $\mathbb{F}$, there is a hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = n^{O(\log n)}$, that hits every $0 \neq f \in ROF^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \geq n^2$.*

Since a hitting set for all polynomials of the form $g - h$ where $g, h \in \mathcal{C}$ is the same as an interpolating set for $\mathcal{C}$, the following theorem gives an interpolating set for the orbit of ROANFs.

▶ **Theorem 35.** *Let $f_1 = ANF_{\Delta_1}(A_1\boldsymbol{x} + \boldsymbol{b}_1), f_2 = ANF_{\Delta_2}(A_2\boldsymbol{x} + \boldsymbol{b}_2) \in ANF^{GL_n^{aff}(\mathbb{F})}$ and $f = f_1 - f_2$. Set $k \triangleq 2\max\{\Delta_1, \Delta_2\} + 7$ and let $\mathcal{G}$ be any uniform k-independent polynomial map, over $\mathbb{F}$. If $f \neq 0$ then $f \circ \mathcal{G} \neq 0$.*

▶ **Corollary 36.** *For any field $\mathbb{F}$, the class $ANF_{\Delta}^{GL_n^{aff}(\mathbb{F})}$, for $4^{\Delta} \leq n$, admits an interpolating set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = n^{O(\Delta)}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$, such that $|\mathbb{K}| \geq n^2$.*

Finally, we observe that the randomized algorithm of Gupta, Kayal And Qiao [35], for reconstructing *random algebraic formula* (for a natural definition of a random formula), yields a randomized reconstruction algorithm for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$. Naturally, the reconstruction is up to the symmetry group of ROANFs.

▶ **Theorem 37** (A special case of Theorem 1.1 of [35]). *Let T be a finite subset of $\mathbb{C}$. Let $n, \Delta \geq 1$ be integers such that $s \triangleq 4^{\Delta} \leq n$. Given black-box access to the output f of a circuit $\Phi \in ANF^{GL_n^{aff}(\mathbb{C})}$, with probability at least $1 - \frac{n^2 s^{O(1)}}{|T|}$ (on internal randomness), Algorithm 6.9 of [35] successfully computes a tuple of s linearly independent linear functions $L = (\ell_1, \ldots, \ell_s) \in (\mathbb{C}[\boldsymbol{x}])^s$ such that $f = ANF_{\Delta}(\ell_1, \ldots, \ell_s)$, and the ℓ_is are identical to the labels of the leaves of Φ up to $TS_n(\mathbb{C})$-equivalence (see Definition 22). Moreover, the running time of the algorithm is poly$(n, s, \log(|T|))$.*

▶ **Remark 38.** *Theorem 1.1 of [35] is stated only for characteristic zero fields. However, in Remark 6.10 they explain how to make the algorithm work over any characteristic, for a large enough field. Thus, Theorem 37 also holds over large enough fields in arbitrary characteristic.*

▶ **Remark 39.** *As a direct implication of Theorem 35, the reconstruction algorithm of Theorem 37 can be converted into a zero-error algorithm, with expected quasipolynomial running time: Given black-box access to some $f_1 \in ANF^{GL^{aff}(\mathbb{F})}$, we define f_2 to be the output of the algorithm of Theorem 37 on input f_1, and then verify $f_1 = f_2$ using Corollary 36.*

1.2.3 Dense subclasses of $\Sigma\Pi\Sigma$

We start by defining the canonical diagonal tensor of degree d and rank s, $\mathrm{T}_{s,d} \in \mathbb{F}[x_{1,1}, \ldots, x_{s,d}]$, and the resulting class of polynomials $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

▶ **Definition 40.** *Let $T_{s,d} \triangleq \sum_{i=1}^{s} \prod_{j=1}^{d} x_{i,j}$. I.e., it is a sum of s variable-disjoint monomials. For $n \geq s \cdot d$, we denote with $T_{s,d}^{GL_n^{aff}(\mathbb{F})}$ the orbit of $T_{s,d}$ over $\mathbb{F}$, under the action of the affine group. Finally, we denote with $\mathcal{T}^{GL^{aff}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$, such that for every n there exist s and d such that $n \geq s \cdot d$ and $f_n(x_1, \ldots, x_n) \in T_{s,d}^{GL_n^{aff}(\mathbb{F})}$.*

Clearly, $\mathrm{T}_{s,d}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \subset \Sigma^{[s]}\Pi^{[d]}\Sigma$. We next define the class consisting of orbits of sparse polynomials.

▶ **Definition 41.** *Let $\Sigma\Pi^{GL^{aff}(\mathbb{F})}$ denote the class of families of polynomials that are computed by orbits of depth-2 circuits, of polynomially bounded size, over $\mathbb{F}$. I.e., it is all families $(f_n)_n$, of polynomially bounded degree, such that for some polynomially bounded $m(n)$, there exist $\Sigma^{m(n)}\Pi^{\deg(f_n)}$ circuits Φ_m, in $k \leq n$, many variables, such that $f_n \in \Phi_m^{GL_n^{aff}(\mathbb{F})}$.*

As before we first give the basic observation connecting all three classes.

▶ **Theorem 42.** *For every field $\mathbb{F}$ it holds that*

$$\mathcal{T}^{GL^{aff}(\mathbb{F})} \subsetneq \Sigma\Pi^{GL^{aff}(\mathbb{F})} \subseteq \Sigma\Pi\Sigma(\mathbb{F}) ,$$

and for fields of size $|\mathbb{F}| \geq n+1$

$$\Sigma\Pi^{GL^{aff}(\mathbb{F})} \subsetneq \Sigma\Pi\Sigma(\mathbb{F}) .$$

In addition,

$$\overline{\mathcal{T}^{GL^{aff}(\mathbb{F})}} = \overline{\Sigma\Pi^{GL^{aff}(\mathbb{F})}} = \overline{\Sigma\Pi\Sigma(\mathbb{F})} . \tag{7}$$

Our main results for this section are a quasipolynomial-size hitting set for the class $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, and a polynomial-size interpolating set for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

▶ **Theorem 43.** *Let $0 \neq g \in \mathbb{F}[\boldsymbol{x}]$ have sparsity $\leq 2^t$. Let $(A, \boldsymbol{b}) \in GL_n^{aff}(\mathbb{F})$, and $f(\boldsymbol{x}) = g(A\boldsymbol{x} + \boldsymbol{b})$. Then, for any $(t+1)$-independent polynomial map $\mathcal{G}$, $f \circ \mathcal{G} \neq 0$.*

▶ **Corollary 44.** *For any integers s, d, n, there exists an explicit hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = (nd)^{O(\log s)}$, such that $\mathcal{H}$ hits every nonzero polynomial $f \in \left(\Sigma^{[s]}\Pi^{[d]}\right)^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| \leq n \cdot d$ then we let $\mathcal{H}$ be defined over an extension field $\mathbb{K}$ of $\mathbb{F}$ of size $|\mathbb{K}| > n \cdot d$.*

We next state our result concerning an interpolating set for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

▶ **Theorem 45.** *Let $n, s_1, s_2, d_1, d_2 \in \mathbb{N}$ be such that $n \geq s_1 \cdot d_1, s_2 \cdot d_2$. For $i \in \{1, 2\}$ let $f_i \in T_{s_i,d_i}^{GL_n(\mathbb{F})}$, and let $f = f_1 - f_2$. If $f \neq 0$, then any uniform 6-independent polynomial map $\mathcal{G}$ satisfies $f \circ \mathcal{G} \neq 0$.*

Finally we note that the randomized reconstruction algorithm of Kayal and Saha [45], which works for (as it is termed in their paper) "non-degenerate" homogeneous depth-3 circuits, works for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. This follows from the observation that $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ circuits are always non-degenerate.

▶ **Theorem 46** (special case of Theorem 1 of [45]). *Let $n, d, s \in \mathbb{N}$, $n \geq (3d)^2$ and $s \leq (\frac{n}{3d})^{\frac{d}{3}}$. Let $\mathbb{F}$ be a field of characteristic zero or greater than ds^2. There is a randomized poly$(n, d, s) =$ poly(n, s) time algorithm which takes as input black-box access to a polynomial f that is computable by a $T_{s,d}^{GL_n^{aff}(\mathbb{F})}$ circuit, and outputs a $T_{s,d}^{GL_n^{aff}(\mathbb{F})}$ circuit Φ computing f with high probability. Furthermore, Φ is unique up to $TPS_{s,d}(\mathbb{F})$-equivalence (see Definition 25).*

▶ **Remark 47.** *As in Remark 39, Theorem 45 enables us to convert the reconstruction algorithm of Theorem 46 to a zero-error algorithm, with expected polynomial running time. Given black-box access to some $f_1 \in \mathcal{T}^{GL^{aff}(\mathbb{F})}$, we define f_2 to be the output of the algorithm of Theorem 46 on input f_1, and then verify $f_1 \equiv f_2$ by applying Theorem 45 to $f = f_1 - f_2$.*

1.2.4 Robust hitting sets?

As we showed in Observation 14, if a hitting set $\mathcal{H}$ for a circuit class $\mathcal{C}$ is *robust*, then $\mathcal{H}$ hits $\overline{\mathcal{C}}$ as well. It is thus natural to ask whether our interpolating sets are already robust. Our next result shows that the property of being a t-independent map, which was sufficient for the constructions in Theorems 28, 30, 33, 35, 43, and 45 (for the appropriate values of t), by itself is not sufficient for obtaining robust hitting sets. We prove this by constructing an independent polynomial map which gives rise to a provably non-robust hitting set. Our construction is the same as the one given by Forbes et al. [27] (Construction 6.3 in the full version).

▶ **Theorem 48.** *Let $\mathbb{F}$ be of characteristic zero. For every t, there exists a uniform t-independent polynomial map $\mathcal{G}$ and a nonzero polynomial f such that $f \circ \mathcal{G} \equiv 0$, and f can be computed by a $\Sigma\Pi\Sigma$ formula of size $t^{O(\sqrt{t})}$. If $\mathbb{F}$ has a positive characteristic then f can be computed by a $\Sigma\Pi\Sigma$ formula of size t^t, or by a general formula of size $t^{O(\log t)}$. Furthermore, for a certain arrangement of the variables in a $\sqrt{n} \times \sqrt{n}$ matrix, f can be taken to be the determinant of any $(t+1) \times (t+1)$ minor.*

1.3 Polynomial Identity Testing

So far we discussed our work from the perspective of dense subclasses of classes for which no strong lower bounds are known. Here we put our work in the context of the polynomial identity testing problem.

Polynomial Identity Testing (PIT for short) is the problem of designing efficient deterministic algorithms for deciding whether a given arithmetic circuit computes the identically zero polynomial. PIT has many applications, e.g. deciding primality [1], finding a perfect matching in parallel [23, 69] etc., and strong connection to circuit lower bounds [38, 22, 19, 32]. See [67, 62, 63] for surveys on PIT and [50] for a survey of algebraic hardness-randomness tradeoffs.

PIT is considered both in the white-box model, in which we get access to the graph of computation of the circuit, and in the black-box model in which we only get query access to the polynomial computed by the circuit. Clearly, a deterministic PIT algorithm in the black-box model is equivalent to a hitting set for the circuit class. In this work we only focus on the black-box model.

The continuant polynomial and algebraic branching programs

The continuant polynomial is trivially computed by width-2 *Algebraic Branching Programs* (ABPs). Recall that an ABP of depth-d and width-w computes polynomials of the form $\mathrm{Trace}\left(M_1(\boldsymbol{x}) \cdot \ldots \cdot M_d(\boldsymbol{x})\right)$, where each M_i is a $w \times w$ matrix whose entries contain variables or field elements. Ben-Or and Cleve proved that every polynomial in $\mathrm{VP_e}$ can be computed by a width-3 ABP of polynomial-size [8].

Raz and Shpilka gave the first polynomial-time white-box PIT algorithm for read-once ABPs (ABPs in which every variable can appear in at most one matrix) [58]. Forbes, Saptharishi and Shpilka gave the first quasipolynomial-sized hitting set for read-once ABPs (ROABPs) [25]. This result was slightly improved in [31] for the case where the width of the ROABP is small. Anderson et al. gave a subexponential hitting set for read-k ABPs [5]. We note that none of these models is strong enough to contain the orbit $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. For ABPs that are not constant-read we do not have sub-exponential time PIT algorithms. Thus, the following is an interesting open problem (recall that by the result of Ben-Or and Cleve a PIT algorithm for width-3 ABPs works for $\mathrm{VP_e}$ as well).

▶ **Problem 49.** *Give a sub-exponential time PIT algorithm for ABPs of width-*2*.*

It is interesting to note that by a result of Saha, Saptharishi and Saxena [59], PIT for ABPs of width-2 would yield PIT for $\Sigma\Pi\Sigma$ circuits.

Although we do not have a PIT algorithm for general branching programs, in [44] Kayal et al. gave an average-case reconstruction algorithm for low width ABPs. Kayal, Nair and Saha obtained a significantly better algorithm in [43]. Their algorithm succeeds w.h.p, provided the ABP satisfies four non-degeneracy conditions (these conditions are defined in Section 4.3 of [43]). However, the ABP computing the continuant polynomial does not satisfy the non-degeneracy conditions that are required for their algorithm to work. Thus, Theorem 31 does not follow from [43].

To the best of our knowledge, $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ is the first natural[9] computational class that is dense in $\mathrm{VP_e}$ for which a polynomial (or even sub-exponential)-sized interpolating set (or a hitting set) is known.

Read-Once formulas

Hitting sets for read-once formulas were first constructed by Volkovich and Shpilka [66], who gave quasipolynomial-sized hitting set for the model, as well as a deterministic reconstruction algorithm of the same running time (earlier randomized reconstruction algorithms were known [16, 15]). Minahan and Volkovich obtained a polynomial-sized hitting set for the class, which led to a similar improvement in the running time of the reconstruction algorithm [54]. Anderson, van Melkebeek and Volkovich constructed a hitting set of size $n^{k^{O(k)}+O(k \log n)}$ for read-k formulas [6]. All these results work in a slightly stronger model in which we allow to label leaves with univariate polynomials, of polynomial degree, such that every variable appears in at most one polynomial, or with sparse polynomials on disjoint sets of variables.

The read-once models that we consider here, $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$, can be viewed as read-once formulas composed with a layer of addition gates with the restriction that the bottom layer of additions computes linearly independent linear functions. We note that these models do not fall into any of the previously studied models, as a variable can appear in all the linear functions.

As is the case with $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, our hitting sets for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ are the first sub-exponential-sized hitting sets for natural dense subclasses of $\mathrm{VP_e}$.

Small depth circuits

The class of $\Sigma\Pi$ circuits was considered in many works, see e.g. [9, 48] and polynomial-sized hitting sets were constructed. The class of $\Sigma\Pi\Sigma$ circuits also received a lot of attention but with lesser success. Dvir and Shpilka [21] and Karnin and Shpilka [40] gave the first quasipolynomial-time white-box and black-box PIT algorithms for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ circuits, respectively. Currently, the best result is by Saxena and Seshadhri who gave a hitting set of size $(nd)^{O(k)}$ for such circuits [64]. In [20] a subexponential-size hitting set for *multilinear* $\Sigma\Pi\Sigma$ circuits was given. In [2], Agrawal et al. gave a hitting set of size $n^{O(1)} \cdot (kd)^{O(r)}$ for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ circuits, where r is an upper bound on the *algebraic rank* of the multiplication gates in the circuit. Thus, known quasipolynomial-size hitting sets for subclasses of $\Sigma\Pi\Sigma$ circuits are known when the fan-in of the top gate is poly-logarithmic, or when the algebraic rank of

[9] It is hard to define what a natural class means, but, for example the set of all polynomials in $\mathrm{VP_e}$ with a nonzero free term has a trivial hitting set, but is not a "computational" subclass.

the set of multiplication gates is poly-logarithmic. In contrast, polynomials in $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ and $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, when viewed as $\Sigma\Pi\Sigma$ circuits, can have polynomially many multiplication gates and their algebraic rank can be n. On the other hand, the corresponding $\Sigma\Pi\Sigma$ circuits are such that the *different* linear functions that are computed at their bottom layer are linearly independent (when we view linear functions that are a constant multiple of each other as the same function). Thus, our Corollary 44 provides a hitting set for a new subclass of $\Sigma\Pi\Sigma$ circuits.

To the best of our knowledge, our results for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ give the first sub-exponential size hitting sets for natural subclasses that are dense in $\Sigma\Pi\Sigma$.

1.4 More related work

Approximations in algebraic complexity were first studied by Bini et al. in the context of algorithms for matrix multiplication [12]. For more on the history of border rank in the context of matrix multiplication see notes of chapter 15 in [18]. More recently, influenced by the GCT program, a lot of research was invested in trying to find polynomials characterizing tensors of small rank. See [51] for a discussion on this approach. More recently, Kumar proved that *every* polynomial over $\mathbb{C}$ can be approximated by a $\Sigma^{[2]}\Pi\Sigma$ circuit (of exponential degree) [49].

Very little is known about the closure of circuit classes. Forbes observed that the class of ROABPs is closed [24]. I.e. $\mathrm{ROABP} = \overline{\mathrm{ROABP}}$. We are not aware of other collapses or separation between general "natural" classes and their closures.

Beside the reconstruction algorithms mentioned earlier, reconstruction algorithms are known for $\Sigma\Pi$ circuits [9, 48]; for random depth three *powering* circuits [42]; for set-multilinear $\Sigma\Pi\Sigma$ and ROABPs [7, 47]; for $\Sigma\Pi\Sigma$ circuits with bounded top fan-in [65, 41, 68]; and for multilinear depth-4 circuits with a constant top fan-in [34, 11].

In general, we do not expect the reconstruction problem to be solvable efficiently, as the problem of finding the minimal circuit computing a given polynomial is a notoriously hard problem. A detailed discussion on the hardness of reconstruction can be found in [43].

Independently and concurrently with our work Saha and Thankkey gave PIT algorithms for orbits of different models of read-once oblivious algebraic branching programs (ROABPs) and for constant-depth, constant-occur formulas [60]. Their results concerning ROABPs were recently improved by Bhargava and Ghosh [10]. Interestingly, both [60, 10] use k-wise independent maps in their construction. We note that the only model that is studied in this paper and in [60, 10] is that of (orbits of) sparse polynomials. For orbits of sparse polynomials are hitting set is potentially much smaller than those constructed in [60, 10] as it does not depend on the individual degrees appearing in the sparse polynomial.

Simultaneously and independently, Saha and Thankey [60] studied PIT for orbits of related computational models. Specifically, they obtained quasi-polynomial sized hitting sets for: Low-individual-degree polynomials computable by commutative ROABP; Multilinear polynomials computable by constant-width ROABP; Polynomials computable by constant-depth, constant-occur formulas with low-individual-degree sparse polynomials at the leaves; and Polynomials computable by occur-once formulas with low-individual-degree sparse polynomials at the leaves. We refer the reader for their paper for definitions of these models. The results of [60] are mostly disjoint from ours, except for the model of sparse polynomials that is captured by commutative ABPs. In this case our result is superior to theirs as their hitting set has an exponential dependence in the individual degrees, while ours work for any polynomial degree sparse polynomial. It is interesting to note that the hitting set constructions of [60] are also based on k-independent maps.

1.5 Proof technique

Our proofs are based on the following simple yet important, and as far as we know novel, observations concerning k-independent polynomial maps. Specifically, our proofs are based on the following two claims:

1. If we have a hitting-set generator H for nonzero polynomials of the form $\frac{\partial f}{\partial x_1}$, for $f \in \mathcal{C}$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. This is proved in Lemma 61.
2. Similarly, we prove that if we have a hitting-set generator H for nonzero polynomials of the form $f\big|_{\ell=0}(A\boldsymbol{x} + \boldsymbol{b})$, for $f \in \mathcal{C}$, a linear function ℓ, and an invertible affine transformation $(A, \boldsymbol{b})$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. This follows from Lemma 62.

By applying these claims $k + r$ times we get that composition with a $(k + r)$-independent map allows to reduce the problem of hitting a class $\mathcal{C}$ to hitting polynomials of the form $\frac{\partial^k f}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_k}}\Big|_{\ell_1 = \ldots = \ell_r = 0}$. Thus, if we could prove that for a class $\mathcal{C}$, there is such a sequence of derivatives and restrictions that simplifies the polynomials in it to a degree that they can be easily hit by some map H, then we conclude that $H + \mathcal{G}_{k+r}$, for a $(k + r)$-independent map $\mathcal{G}_{k+r}$, is a hitting set generator for $\mathcal{C}$.

It seems that all that is left to do is prove that for each of the orbits that we consider in Section 1.2 that is such small k and r. However, a potential problem is that a partial derivative of the polynomial $g(\boldsymbol{x}) = f(A\boldsymbol{x} + \boldsymbol{b})$ gives $\frac{\partial g}{\partial x_1} = \sum_{i=1}^{n} \frac{\partial f}{\partial y_i} \cdot \frac{\partial \ell_i}{\partial x_1}$, where ℓ_i is the ith coordinate of $A\boldsymbol{x} + \boldsymbol{b}$. Thus, it is no longer a derivative composed with an affine transformation but rather a sum of such derivatives, which could lead to polynomials outside of our class. For example, it is not hard ot prove that if we compose the ROF $y_1 \cdot y_2 \cdot y_3$ with $(x_1, x_1 + x_2, x_1 + x_3)$ and then take a derivative according to x_1, then the resulting polynomial, $\frac{\partial(x_1 \cdot (x_1 + x_2) \cdot (x_1 + x_3))}{\partial x_1} = 3x_1^2 + 2x_1 \cdot (x_2 + x_3) + x_2 \cdot x_3$, is not in the orbit of any ROF. The solution to this problem is to take a *directional derivative* in a direction coming from a *dual basis.* For example if $\ell_i(\boldsymbol{v}_j) = \delta_{i,j}$ then $\frac{\partial g}{\partial \boldsymbol{v}_1} = \frac{\partial f}{\partial x_1}(A\boldsymbol{x} + \boldsymbol{b})$ (see Lemma 60). Now, comes another important observation: If H is a hitting-set generator for nonzero polynomials of the form $\frac{\partial f}{\partial \boldsymbol{v}}$, for $f \in \mathcal{C}$ and a direction $\boldsymbol{v}$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. The point is that if $\frac{\partial f}{\partial \boldsymbol{v}} \circ H \neq 0$ then for some i, $\frac{\partial f}{\partial x_i} \circ H \neq 0$ and the claim follows from the first claim above. Thus, composition with $(k + r)$-independent maps allows us to reduce the problem of hitting a class $\mathcal{C}$ to finding a generator for polynomials that are obtained as a restriction to a subspace of co-dimension r of a directional partial derivative of order k of polynomials in $\mathcal{C}$.

Let us demonstrate this idea for the case of orbits of sparse polynomials. I.e. to polynomials of the form $g(\boldsymbol{x}) = f(A\boldsymbol{x} + \boldsymbol{b})$, where the number of monomials in f is at most 2^t. It is not hard to see that there is a variable x_i such that if we consider $f\big|_{x_i=0}$ and $\frac{\partial f}{\partial x_i}$ then one of these polynomials has at most 2^{t-1} monomials.[10] Thus, after a a sequence of at most t partial derivatives and restrictions, we get to a polynomial with only one monomial that we can easily hit. Hence after at most t directional derivatives and restrictions to a subspace, we get that g is a product of linear forms, which we can easily hit. This proves that any $(t + 1)$-independent map hits such nonzero polynomials g.

[10] This is not exactly accurate – it only holds if f is not divisible by some variable x_i. However, the case where there is a monomial dividing f is also quite easy to handle as it is enough to hit the polynomial obtained after dividing by that monomial (since a composition with a 1-independent map keeps any nonzero linear function nonzero).

To obtain interpolating sets for our classes (and also a reconstruction algorithm for the orbit of the continuant polynomial), we prove that if two polynomials in the orbit, of any of the classes that we consider, are different, then there is a sequence of a few (directional) partial derivatives and restrictions that makes one of them zero while keeping the other nonzero. Using this and the ideas from above we construct our interpolating sets.

▶ **Remark 50.** *In this version of the paper we only give proofs of the main properties of k-wise independent maps (outlined above), as these are the main tool that we used in all our proofs. The full version can be found at [53].*

1.6 Discussion

As Theorem 48 shows, our hitting sets are not necessarily robust. It is thus an outstanding open problem to find a way to convert a hitting set to a robust one (recall Problem 1).

The following toy example demonstrates that converting a hitting set for a class $\mathcal{C}$ to a robust hitting set for $\mathcal{C}$, cannot be done in a black-box manner and one has to use information about $\mathcal{C}$ for that: let $\mathcal{C}(\mathbb{F})$ be the class of all polynomials with non-zero free term. A trivial hitting set for $\mathcal{C}$ would simply be the singleton set $\mathcal{H} = \{\mathbf{0}\}$. On the other hand, it is clear that $\overline{\mathcal{C}} = \mathbb{F}[\boldsymbol{x}]$, so making $\mathcal{H}$ robust would yield a hitting set for *all* polynomials. Note, however, that this is not a "computational class."

Another potential approach for obtaining robust hitting sets follows from the observation that the set of queries made by a non-adaptive deterministic black-box reconstruction algorithm, $\mathcal{A}$, for $\mathcal{C}$, which is *continuous* at 0 (i.e. at the identically zero polynomial) is a robust hitting set for $\mathcal{C}$. The reason is, that if $0 \neq f \in \overline{\mathcal{C}}$ and $\{f_k\}_{k=1}^{\infty} \subseteq \mathcal{C}$ converges to f, then for large enough k: $\|f_k\|_2 \geq \frac{1}{2}\|f\|_2 > 0$. As the f_k sequence converges and polynomial evaluation is continuous (and their evaluation vectors are bounded), the sequence $\boldsymbol{v}_k = f_k\big|_{\mathcal{H}} \subseteq \mathbb{C}^{|\mathcal{H}|}$ must also converge to some vector $\boldsymbol{v} = f\big|_{\mathcal{H}} \in \mathbb{C}^{|\mathcal{H}|}$. If $\boldsymbol{v} = \mathbf{0}$ then the continuity of $\mathcal{A}$ at $\mathbf{0}$ implies the coefficients of the polynomials $f_k(\boldsymbol{x})$ must also converge to zero, as $\mathcal{A}(\mathbf{0}) = 0$. This would contradict $\|f_k\|_2 \geq \frac{1}{2}\|f\|_2 > 0$ for large enough k, so $\boldsymbol{v} \neq \mathbf{0}$ and thus $\mathcal{H}$ hits $\overline{\mathcal{C}}$.

Thus, an interesting challenge is to derandomize the reconstruction algorithms given in Theorems 31, 37, and 46, hoping that the resulting algorithms are continuous at $\mathbf{0}$. We note however, that currently we do not even have efficient deterministic root-finding algorithms over $\mathbb{C}$. It is also known that in general, finding the minimal circuit for a polynomial can be very difficult. E.g., in [36, 70] it was shown that the question of computing, or even approximating, tensor rank, for degree 3 tensors, is NP hard, over any field.

▶ Remark 51. In Theorem 45, we have seen that any uniform $O(\log(sn))$-independent polynomial map $\mathcal{G}$ is an interpolating set generator for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$; i.e, $\mathcal{G}$ induces an interpolating set $\mathcal{H}$ for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$. On the other hand, in Theorem 48, we constructed such a map $\mathcal{G}$, with the additional property that $\mathcal{G}$ is *not* a hitting set generator for $\Sigma\Pi\Sigma$ circuits. In particular, this implies that the induced (non-efficient) reconstruction map $\mathcal{A}$ (that takes $f(\mathcal{H})$ and returns a circuit computing f) is not continuous at $\mathbf{0}$.

We conclude this section with a somewhat vague question.

▶ **Problem 52.** *Find a "computational" class of polynomials $\mathcal{C}$ with a known hitting set $\mathcal{H}$, such that $\overline{\mathcal{C}} \neq \mathcal{C}$, and convert $\mathcal{H}$ to a robust hitting set.*

We note that the closure of $\Sigma\bigwedge\Sigma$ circuits (i.e. circuits computing polynomials of the form $\sum_i \ell_i(\boldsymbol{x})^d$, for linear functions ℓ_i) is contained in the class of commutative read-once algebraic branching programs (see [25]). Thus, the hitting set for the latter class gives a robust hitting set for the former [25]. However, we seek an example in which there is an "interesting" conversion of a hitting set to a robust one.

2 k-independent polynomial maps and their properties

▶ **Observation 53.** *It holds that*

1. *If $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ is a $(k+1)$-independent polynomial map, then there exists a subset of variables S and an assignment $\boldsymbol{\alpha} \in \mathbb{F}^{|S|}$ such that $\mathcal{G}\big|_{S=\boldsymbol{\alpha}}$ is a k-independent polynomial map.*
2. *For any $k \geq 1$, the n coordinates of any k-independent polynomial map are $\mathbb{F}$-linearly independent.*
3. *Let $\ell_1(\boldsymbol{x})$ and $\ell_2(\boldsymbol{x})$ be linearly independent linear functions in $\mathbb{F}[\boldsymbol{x}]$. Let $\mathcal{G}(\mathbf{y}, z_1, z_2)$ be any 2-independent polynomial map. Consider $\ell_1 \circ \mathcal{G}$ and $\ell_2 \circ \mathcal{G}$ as polynomials in z_1, z_2 over $\mathbb{F}(\mathbf{y})$. Then, $(\ell_1 \circ \mathcal{G})^{[1]}$ and $(\ell_2 \circ \mathcal{G})^{[1]}$ are linearly independent, as linear forms in z_1, z_2 over $\mathbb{F}(\mathbf{y})$.*

We next give the construction of [66] of a k-independent polynomial map (denoted G_k in [66]).

▶ **Definition 54.** *Fix n and a set of n distinct field elements $\mathcal{A} = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}$.[11] For every $i \in [n]$ let $L_i(w) : \mathbb{F} \to \mathbb{F}$ be the ith Lagrange Interpolation polynomial for the set $\mathcal{A}$. That is, each $L_i(w)$ is polynomial of degree $n-1$ that satisfies $L_i(\alpha_j) = \delta_{i,j}$. We define $\mathcal{G}_1^{SV}(y_1, z_1) : \mathbb{F}^2 \to \mathbb{F}^n$ as:*

$$\mathcal{G}_1^{SV}(y_1, z_1) \triangleq (L_1(y_1) \cdot z_1, L_2(y_1) \cdot z_1, \ldots, L_n(y_1) \cdot z_1),$$

and for any $k \geq 1$, we define $\mathcal{G}_k^{SV} : \mathbb{F}^{2k} \to \mathbb{F}^n$ as:

$$\mathcal{G}_k^{SV}(\boldsymbol{y}, \boldsymbol{z}) \triangleq \mathcal{G}_1^{SV}(y_1, z_1) + \mathcal{G}_1^{SV}(y_2, z_2) + \ldots + \mathcal{G}_1^{SV}(y_k, z_k)$$

$$= \left(\sum_{j=1}^{k} L_1(y_j) \cdot z_j, \sum_{j=1}^{k} L_2(y_j) \cdot z_j, \ldots, \sum_{j=1}^{k} L_n(y_j) \cdot z_j \right).$$

▶ **Observation 55.** *$\mathcal{G}_k^{SV}$ is a k-independent polynomial map, in which each variable has degree at most $n-1$.*

The generator $\mathcal{G}_k^{\mathrm{SV}}$ can be converted to a uniform k-independent polynomial map by adding another k control variables $y_{k+1}, \ldots, y_{2k}$, and swapping out the $L_i(y_j)$s for their homogenizations $y_{j+k}^{n-1} L_i\left(\frac{y_j}{y_{j+k}}\right)$:

▶ **Definition 56.** *With the notation used in Definition 54, define the* uniform SV-generator *with k independence $\mathcal{G}_k^{SV\text{-}hom} : \mathbb{F}^{3k} \to \mathbb{F}^n$ as:*

$$\begin{aligned}
&\mathcal{G}_k^{SV\text{-}hom}(y_1, \ldots, y_{2k}, z_1, \ldots, z_k) \\
&\quad \triangleq y_{1+k}^{n-1} \cdot \mathcal{G}_1^{SV}\left(\frac{y_1}{y_{1+k}}, z_1\right) + y_{2+k}^{n-1} \cdot \mathcal{G}_1^{SV}\left(\frac{y_2}{y_{2+k}}, z_2\right) + \ldots + y_{2k}^{n-1} \cdot \mathcal{G}_1^{SV}\left(\frac{y_k}{y_{2k}}, z_k\right) \\
&\quad = \left(\sum_{j=1}^{k} y_{j+k}^{n-1} L_1\left(\frac{y_j}{y_{j+k}}\right) \cdot z_j, \sum_{j=1}^{k} y_{j+k}^{n-1} L_2\left(\frac{y_j}{y_{j+k}}\right) \cdot z_j, \ldots, \sum_{j=1}^{k} y_{j+k}^{n-1} L_n\left(\frac{y_j}{y_{j+k}}\right) \cdot z_j \right).
\end{aligned}$$

▶ **Observation 57.** *$\mathcal{G}_k^{SV\text{-}hom}$ is a uniform k-independent polynomial map, with individual degrees at most $n-1$.*

[11] If $|\mathbb{F}| < n$ then we take these elements from an appropriate extension field of $\mathbb{F}$.

We next show how we can use k-independent polynomial maps in order to, roughly, simulate a kth order directional derivative or, project a polynomial to a subspace of co-dimension k. We first need to define the notion of a directional derivative.

▶ **Definition 58.** *For an n-variate polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ and $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}^n$, the* derivative of $f(\boldsymbol{x})$ in the direction $\boldsymbol{v}$ *is defined as:*

$$\frac{\partial f}{\partial \boldsymbol{v}} = \sum_{i=1}^{n} v_i \cdot \frac{\partial f}{\partial x_i}.$$

If $\mathbb{F}$ has positive characteristic then by $\frac{\partial F}{\partial x_i}$ we refer to the formal derivative (which in the case of fields of characteristic zero is equal to the analytical definition). Observe that we still have that

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}, \quad \frac{\partial (fg)}{\partial x} = \frac{\partial f}{\partial x} \cdot g + \frac{\partial g}{\partial x} \cdot f$$

and

$$\frac{\partial f\left(g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\right)}{\partial x_k} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i}\left(g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\right) \cdot \frac{\partial g_i}{\partial x_k},$$

where in the last expression f is an m variate polynomial, and $g_1, \ldots, g_m$ are n variate polynomials.

We shall often take derivatives according to a *dual set* to a set of linearly independent linear functions:

▶ **Definition 59.** *A dual set for m linearly independent linear functions (recall that we say that linear functions are linearly independent if and only if their degree-1 homogeneous parts are linearly independent) in $n \geq m$ variables, $\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})$ is a set of m vectors $\{\boldsymbol{v}_i\} \subset \mathbb{F}^n$ such that $\ell_i^{[1]}(\boldsymbol{v}_j) = \delta_{i,j}$.*

▶ **Lemma 60.** *Let $\ell_1, \ldots, \ell_m \in \mathbb{F}[x_1, \ldots, x_n]$, for $n \geq m$, be linearly independent linear functions. Let $\{\boldsymbol{v}_i\} \subset \mathbb{F}^n$ be a dual set. Let $g \in \mathbb{F}[y_1, \ldots, y_m]$ be a polynomial. Then, for $f(\boldsymbol{x}) = g\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right)$ it holds that*

$$\frac{\partial f}{\partial \boldsymbol{v}_i}(\boldsymbol{x}) = \frac{\partial g}{\partial y_i}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right).$$

Proof.

$$\begin{aligned}\frac{\partial f}{\partial \boldsymbol{v}_i}(\boldsymbol{x}) &= \sum_{j} v_{i,j} \cdot \frac{\partial f}{\partial x_j}(\boldsymbol{x}) = \sum_{j,k} v_{i,j} \cdot \frac{\partial \ell_k}{\partial x_j} \cdot \frac{\partial g}{\partial y_k}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) \\ &= \sum_{k} \ell_k^{[1]}(\boldsymbol{v}_i) \cdot \frac{\partial g}{\partial y_k}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) = \frac{\partial g}{\partial y_i}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right).\end{aligned}$$ ◀

▶ **Lemma 61.** *Let $f \in \mathbb{F}[\boldsymbol{x}]$ where $\boldsymbol{x} = (x_1, \ldots, x_n)$. Let $H(\boldsymbol{w}) : \mathbb{F}^t \to \mathbb{F}^n$ be a polynomial map in variables $\boldsymbol{w}$, and let $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ be a k-independent polynomial map such that $\operatorname{var}(H) \cap \operatorname{var}(\mathcal{G}) = \emptyset$. Then, for any $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{F}^n$:*

$$\frac{\partial^k f}{\partial \boldsymbol{v}_1 \partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k} \circ H \neq 0 \quad \Rightarrow \quad f \circ (\mathcal{G} + H) \neq 0\,.$$

Proof. By definition of k-independent polynomial maps, $\mathcal{G} = \mathcal{G}_1(\boldsymbol{y_1}, z_1) + \ldots + \mathcal{G}_k(\boldsymbol{y_k}, z_k)$ for some variable-disjoint 1-independent polynomial maps $\mathcal{G}_1, \ldots, \mathcal{G}_k$. It is therefore enough to prove the lemma for $k = 1$, as we can replace f with $\frac{\partial^{k-1} f}{\partial \boldsymbol{v_2} \cdots \partial \boldsymbol{v_k}}$, H with $H + \mathcal{G}_2 + \ldots + \mathcal{G}_k$ and $\mathcal{G}$ with $\mathcal{G}_1$; by iterative application of the result for $k = 1$, we will get the general result for an arbitrary $k \in \mathbb{N}$.

Denote $H = (H_1, H_2, \ldots, H_n)$. By Definition 58, the condition $\frac{\partial f}{\partial \boldsymbol{v}} \circ H \neq 0$ implies that there exists some $i \in [n]$ such that $\frac{\partial f}{\partial x_i} \circ H \neq 0$. Assume, WLOG, $\frac{\partial f}{\partial x_1} \circ H \neq 0$. As $\mathcal{G}$ is a 1-independent polynomial map, there exists some $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y_1}|}$ such that $f \circ (\mathcal{G} + H)\big|_{\boldsymbol{y_1} = \boldsymbol{\alpha}} = f(z_1 + H_1, H_2, \ldots, H_n)$; denote $g \triangleq f \circ (\mathcal{G} + H)\big|_{\boldsymbol{y_1} = \boldsymbol{\alpha}}$. As no coordinate of H depends on z_1:

$$\frac{\partial g}{\partial z_1} = \frac{\partial\,(z_1 + H_1)}{\partial z_1} \cdot \frac{\partial f}{\partial x_1}(z_1 + H_1, H_2, \ldots, H_n) = 1 \cdot \left(\frac{\partial f}{\partial x_1}\right)(z_1 + H_1, H_2, \ldots, H_n)$$

and therefore:

$$\left.\frac{\partial g}{\partial z_1}\right|_{z_1 = 0} = 1 \cdot \left(\frac{\partial f}{\partial x_1}\right)(0 + H_1, H_2, \ldots, H_n) = \left(\frac{\partial f}{\partial x_1}\right) \circ H \neq 0 \,.$$

As g is a projection of $f \circ (\mathcal{G} + H)$, it follows that $f \circ (\mathcal{G} + H) \neq 0$. ◀

The next lemma shows how to use k-independent maps in order to project a polynomial to a subset of its coordinates.

▶ **Lemma 62.** *Let $m \leq n \in \mathbb{N}$ and $g(\boldsymbol{w}) \in \mathbb{F}[w_1, \ldots, w_m]$. Let $f(\boldsymbol{x}) = g(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x}))$ for linearly independent linear functions $\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})$. Let $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ be a k-independent polynomial map. For a set $S \subseteq [n]$ of size k denote by $\tilde{g}(x_i : i \in [m] \setminus S) = g\big|_{S=0}$ the projection of g to the variables outside of S. Then, there exist linearly independent linear functions $\{\tilde{\ell}_i(\boldsymbol{x}) : i \in [m] \setminus S\}$, additional linear functions $\boldsymbol{L}(\boldsymbol{x}) = (L_1(\boldsymbol{x}), \ldots, L_k(\boldsymbol{x}))$ and an assignment $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y}|}$ such that:*

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{L}(\boldsymbol{x}))) = \tilde{g}(\tilde{\ell}_i(\boldsymbol{x}) : i \in [m] \setminus S) \,.$$

Proof. It is enough to prove the lemma for the case $k = 1$, as we may then define $\tilde{f}(\boldsymbol{x}) \triangleq f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L_1(\boldsymbol{x}))) = \tilde{g}(\tilde{\ell}_1(\boldsymbol{x}), \ldots, \tilde{\ell}_{m-1}(\boldsymbol{x}))$ and apply the result iteratively. Thus, assume $k = 1$, and WLOG assume $S = \{x_1\}$ (thus, $\tilde{g}(w_2, \ldots, w_m) = g(0, w_2, \ldots, w_m)$).

Let x_i be some variable with a non-zero coefficient in $\ell_1(\boldsymbol{x})$. Such a variable exists as the ℓ_js are linearly independent. For $j \in [m]$, denote $\beta_j = \frac{\partial \ell_j}{\partial x_i}$, i.e. β_j is the coefficient of x_i in ℓ_j. By our choice of i, $\beta_1 \neq 0$. Choose some $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y}|}$ such that $\mathcal{G}(\boldsymbol{\alpha}, z_1)$ has z_1 in the ith coordinate, and 0 in all other coordinates. Define $L(\boldsymbol{x}) \triangleq -\frac{\ell_1(\boldsymbol{x})}{\beta_1}$, so we get:

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x})) = f\left(x_1, x_2, \ldots, x_{i-1}, x_i - \frac{\ell_1(\boldsymbol{x})}{\beta_1}, x_{i+1}, \ldots, x_n\right).$$

Observe that for every i,

$$\ell_i\left(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x}))\right) = \ell_i\left(x_1, x_2, \ldots, x_{i-1}, x_i - \frac{\ell_1(\boldsymbol{x})}{\beta_1}, x_{i+1}, \ldots, x_n\right) = \ell_i(\boldsymbol{x}) - \frac{\beta_i}{\beta_1} \cdot \ell_1(\boldsymbol{x}) \,.$$

In particular, $\ell_1\left(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x}))\right) = 0$. For $i = 2, \ldots, m$, define:

$$\tilde{\ell}_i(\boldsymbol{x}) \triangleq \ell_i(\boldsymbol{x}) - \frac{\beta_i}{\beta_1} \cdot \ell_1(\boldsymbol{x}) \,.$$

As $\ell_1, \ldots, \ell_m$ are linearly independent, it follows that $\tilde{\ell}_2, \ldots, \tilde{\ell}_m$ are also linearly independent. We get that

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x}))) = g(0, \tilde{\ell}_2(\boldsymbol{x}), \ldots, \tilde{\ell}_m(\boldsymbol{x})) = \tilde{g}(\tilde{\ell}_2(\boldsymbol{x}), \ldots, \tilde{\ell}_m(\boldsymbol{x})) \,.$$ ◀

2.1 Proof of Theorem 48

We next prove that there are k-independent maps that are provably not robust. The proof is by giving a different construction of such maps that, for an appropriate arrangement of the n variables in a matrix, is guaranteed to output matrices of rank at most k. Thus, a determinant of any $(k+1) \times (k+1)$ minor, a polynomial that has small formulas for small values of k, vanishes on the output of any such map.

The fact that such a construction exists was already noticed in [27] (Construction 6.3 of the full version of the paper). For completeness we repeat the construction here.

Proof. *(of Theorem 48)* Fix the number of variables n and assume WLOG n is a perfect square, i.e., $n = m^2$. We index the variables as $x_{i,j}$ for $i,j \in [m]$. We let $f = \mathrm{Det}_{t+1}$. By [33], over fields of characteristic zero, f has a $t^{O(\sqrt{t})} = O(n)$ sized $\Sigma\Pi\Sigma$ formula, which is polynomial in n for $t = O\left((\log n/\log\log n)^2\right)$. Over fields of positive characteristic the formula size is quasipolynomial in t, and the $\Sigma\Pi\Sigma$ complexity is at most $t!$, which is polynomial in n for $t = O(\log n/\log\log n)$.

Denote by $\boldsymbol{M}$ the $(t+1) \times (t+1)$ symbolic matrix of variables $\boldsymbol{M}_{i,j} = x_{i,j}$. We first construct a uniform 1-independent polynomial map $\mathcal{G}_1$ such that $\boldsymbol{M} \circ \mathcal{G}_1$ is of rank 1, and define $\mathcal{G}$ to be a sum of t variable-disjoint copies of $\mathcal{G}_1$. As $rank(\boldsymbol{M} \circ \mathcal{G}_1) = 1$, we have $rank(\boldsymbol{M} \circ \mathcal{G}) \leq t$ so $\mathrm{Det}_{t+1}(\boldsymbol{M} \circ \mathcal{G}) = 0$, as required. We now focus on $\mathcal{G}_1$.

Fix n distinct field elements $\{\alpha_{i,j}\}_{i,j=1}^{m} \subseteq \mathbb{F}$ and let w, y, z be new variables. Define two vectors of polynomials of degree $n-1$, $R = (R_1, \ldots, R_m), C = (C_1, \ldots, C_m) \in \mathbb{F}[y]^m$, such that for every $k \in [m]$ R_k and C_k satisfy

$$R_k(\alpha_{i,j}) = \delta_{i,k} \quad \text{and} \quad C_k(\alpha_{i,j}) = \delta_{j,k}.$$

Define $\mathcal{G}_1(w,y,z)$ as the $m \times m$ matrix $z \cdot (w^{2n-2} R(\frac{y}{w}) \cdot C(\frac{y}{w})^T)$ (the (i,j) entry of $\mathcal{G}_1$ is $z \cdot w^{2n-2} \cdot R_i(\frac{y}{w}) \cdot C_j(\frac{y}{w})$). As every coordinate of $\mathcal{G}_1$ is a homogeneous polynomial of degree $2n-1$, $\mathcal{G}_1$ is a uniform polynomial map. For any $i,j \in [m]$ we have that

$$\mathcal{G}_1(1, \alpha_{i,j}, z) = z \cdot (R_{i'}(\alpha_{i,j}) \cdot C_{j'}(\alpha_{i,j}))_{i',j' \in [m]} = z \cdot (\delta_{i,i'}\delta_{j,j'})_{i',j' \in [m]} .$$

The above matrix has z in entry (i,j) and 0 everywhere else, so $\mathcal{G}_1$ is a uniform 1-independent polynomial map. The resulting matrix $\boldsymbol{M} \circ \mathcal{G}_1$ is of rank 1 since it is a product of vectors $R \cdot C^T$, so the variable-disjoint sum $\mathcal{G} = \sum_1^t \mathcal{G}_1(w_i, y_i, z_i)$ is a uniform t-independent polynomial map satisfying $f \circ \mathcal{G} = 0$. ◀

References

1 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Ann. of Math*, 2:781–793, 2002.

2 Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM J. Comput.*, 45(4):1533–1562, 2016. doi:10.1137/130910725.

3 A. Alder. *Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht*. PhD thesis, Universität Zürich, Philosophische Fakultät II, 1984.

4 Eric Allender and Fengming Wang. On the power of algebraic branching programs of width two. *Computational Complexity*, 25(1):217–253, 2016. doi:10.1007/s00037-015-0114-7.

5 Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *ACM Trans. Comput. Theory*, 10(1):3:1–3:30, 2018. doi:10.1145/3170709.

6 Matthew Anderson, Dieter van Melkebeek, and Ilya Volkovich. Deterministic polynomial identity tests for multilinear bounded-read formulae. *Computational Complexity*, 24(4):695–776, 2015. doi:10.1007/s00037-015-0097-4.

7 Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000. doi:10.1145/337244.337257.

8 Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992. doi:10.1137/0221006.

9 Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynominal interpolation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 301–309. ACM, 1988. doi:10.1145/62212.62241.

10 Vishwas Bhargava and Sumanta Ghosh. Improved hitting set for orbit of roabps. *Electron. Colloquium Comput. Complex.*, 28:62, 2021. URL: https://eccc.weizmann.ac.il/report/2021/062/.

11 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction of depth-4 multilinear circuits. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2144–2160. SIAM, 2020. doi:10.1137/1.9781611975994.132.

12 Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Information Processing Letters*, 8(5):234–235, 1979. doi:10.1016/0020-0190(79)90113-3.

13 Markus Bläser and Christian Ikenmeyer. Introduction to geometric complexity theory. https://pcwww.liv.ac.uk/~iken/teaching_sb/summer17/introtogct/gct.pdf, 2019.

14 Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5):32:1–32:29, 2018. doi:10.1145/3209663.

15 Daoud Bshouty and Nader H. Bshouty. On interpolating arithmetic read-once formulas with exponentiation. *J. Comput. Syst. Sci.*, 56(1):112–124, 1998. doi:10.1006/jcss.1997.1550.

16 Nader H. Bshouty, Thomas R. Hancock, and Lisa Hellerstein. Learning arithmetic read-once formulas. *SIAM J. Comput.*, 24(4):706–735, 1995. doi:10.1137/S009753979223664X.

17 Peter Bürgisser. The complexity of factors of multivariate polynomials. *Found. Comput. Math.*, 4(4):369–396, 2004. doi:10.1007/s10208-002-0059-5.

18 Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.

19 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Hardness vs randomness for bounded depth arithmetic circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 13:1–13:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CCC.2018.13.

20 Rafael Mendes de Oliveira, Amir Shpilka, and Ben lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. *Computational Complexity*, 25(2):455–505, 2016. doi:10.1007/s00037-016-0131-1.

21 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007.

22 Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-Randomness Tradeoffs for Bounded Depth Arithmetic Circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009. doi:10.1137/080735850.

23 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Commun. ACM*, 62(3):109–115, 2019. doi:10.1145/3306208.

24 Michael A. Forbes. Some concrete questions on the border complexity of polynomials. https://www.youtube.com/watch?v=1HMogQIHT6Q, 2016.

25 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 867–875. ACM, 2014. doi:10.1145/2591796.2591816.

26 Michael A. Forbes and Amir Shpilka. A PSPACE construction of a hitting set for the closure of small algebraic circuits. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1180–1192. ACM, 2018. doi:10.1145/3188745.3188792.

27 Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 32:1–32:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. Full version at http://arxiv.org/abs/1606.05050. doi:10.4230/LIPIcs.CCC.2016.32.

28 Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14(1):1–45, 2018. doi:10.4086/toc.2018.v014a018.

29 Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015. doi:10.1007/s00037-015-0103-x.

30 Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017. arXiv:1701.01717.

31 Zeyu Guo and Rohit Gurjar. Improved explicit hitting-sets for roabps. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.4.

32 Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness: Treading the borders. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 147–157. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00018.

33 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. doi:10.1137/140957123.

34 Ankit Gupta, Neeraj Kayal, and Satya Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 625–642, 2012.

35 Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random arithmetic formulas can be reconstructed efficiently. *Computational Complexity*, 23(2):207–303, 2014. doi:10.1007/s00037-014-0085-0.

36 Johan Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990. doi:10.1016/0196-6774(90)90014-6.

37 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute (extended abstract). In Raymond E. Miller, Seymour Ginsburg, Walter A. Burkhard, and Richard J. Lipton, editors, *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 262–272. ACM, 1980. doi:10.1145/800141.804674.

38 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. doi:10.1007/s00037-004-0182-6.

39 Kyriakos Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985. doi:10.1137/0214050.

40 Zohar S Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 280–291. IEEE, 2008.

41 Zohar Shay Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 274–285. IEEE Computer Society, 2009. doi:10.1109/CCC.2009.18.

42 Neeraj Kayal. Affine projections of polynomials. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 643–662, 2012.

43 Neeraj Kayal, Vineet Nair, and Chandan Saha. Average-case linear matrix factorization and reconstruction of low width algebraic branching programs. *Computational Complexity*, 28(4):749–828, 2019. doi:10.1007/s00037-019-00189-0.

44 Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of full rank algebraic branching programs. *ACM Transactions on Computation Theory (TOCT)*, 11(1):1–56, 2018.

45 Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 413–424. ACM, 2019. Full version at https://eccc.weizmann.ac.il/report/2018/191. doi:10.1145/3313276.3316360.

46 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An almost cubic lower bound for depth three arithmetic circuits. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.33.

47 Adam R. Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of Computing*, 2(10):185–206, 2006. doi:10.4086/toc.2006.v002a010.

48 Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223, 2001.

49 Mrinal Kumar. On the power of border of depth-3 arithmetic circuits. *ACM Trans. Comput. Theory*, 12(1):5:1–5:8, 2020. doi:10.1145/3371506.

50 Mrinal Kumar and Ramprasad Saptharishi. Hardness-Randomness tradeoffs for algebraic computation. *Bull. EATCS*, 129, 2019. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/591/599.

51 Joseph M. Landsberg. *Geometry and Complexity Theory.* Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2017. doi:10.1017/9781108183192.

52 Thomas Lehmkuhl and Thomas Lickteig. On the order of approximation in approximative triadic decompositions of tensors. *Theor. Comput. Sci.*, 66(1):1–14, 1989. doi:10.1016/0304-3975(89)90141-2.

53 Dori Medini and Amir Shpilka. Hitting sets and reconstruction for dense orbits in vp$_e$ and $\sigma\pi\sigma$ circuits. *Electron. Colloquium Comput. Complex.*, 28:14, 2021. URL: https://eccc.weizmann.ac.il/report/2021/014.

54 Daniel Minahan and Ilya Volkovich. Complete derandomization of identity testing and reconstruction of read-once formulas. *ACM Transactions on Computation Theory (TOCT)*, 10(3):1–11, 2018.

55 Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory I: an approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526, 2001. doi:10.1137/S009753970038715X.

56 Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory II: towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008. doi:10.1137/080718115.

57 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010. doi:10.4086/toc.2010.v006a007.

58 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. doi:10.1007/s00037-005-0188-8.

59 Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. The power of depth 2 circuits over algebras. In Ravi Kannan and K. Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, volume 4 of *LIPIcs*, pages 371–382. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2009. doi:10.4230/LIPIcs.FSTTCS.2009.2333.

60 Chandan Saha and Bhargav Thankey. Hitting sets for orbits of circuit classes and polynomial families. *Electron. Colloquium Comput. Complex.*, 28:15, 2021. URL: https://eccc.weizmann.ac.il/report/2021/015.

61 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. *Github survey*, 2015. Available at https://github.com/dasarpmar/lowerbounds-survey.

62 Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009.

63 Nitin Saxena. *Progress on Polynomial Identity Testing-II*, volume 26 of *Progress in Computer Science and Applied Logic*, pages 131–146. Birkhäuser Basel, 2014. arXiv:1401.0976.

64 Nitin Saxena and C. Seshadhri. Blackbox Identity Testing for Bounded Top-Fanin Depth-3 Circuits: The Field Doesn't Matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012. doi:10.1137/10848232.

65 Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM Journal on Computing*, 38(6):2130–2161, 2009.

66 Amir Shpilka and Ilya Volkovich. Read-once polynomial identity testing. *Computational Complexity*, 24(3):477–532, 2015.

67 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388, 2010. doi:10.1561/0400000039.

68 Gaurav Sinha. Reconstruction of real depth-3 circuits with top fan-in 2. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 31:1–31:53. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.31.

69 Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.70.

70 Joseph Swernofsky. Tensor rank is hard to approximate. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 26:1–26:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.26.

71 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. doi:10.1137/0212043.

Variety Evasive Subspace Families

Zeyu Guo
Department of Computer Science, University of Haifa, Israel

Abstract

We introduce the problem of constructing explicit *variety evasive subspace families*. Given a family $\mathcal{F}$ of subvarieties of a projective or affine space, a collection $\mathcal{H}$ of projective or affine k-subspaces is $(\mathcal{F}, \epsilon)$-*evasive* if for every $\mathcal{V} \in \mathcal{F}$, all but at most ϵ-fraction of $W \in \mathcal{H}$ intersect every irreducible component of $\mathcal{V}$ with (at most) the expected dimension. The problem of constructing such an explicit subspace family generalizes both deterministic black-box polynomial identity testing (PIT) and the problem of constructing explicit (weak) lossless rank condensers.

Using Chow forms, we construct explicit k-subspace families of polynomial size that are evasive for all varieties of bounded degree in a projective or affine n-space. As one application, we obtain a complete derandomization of Noether's normalization lemma for varieties of bounded degree in a projective or affine n-space. In another application, we obtain a simple polynomial-time black-box PIT algorithm for depth-4 arithmetic circuits with bounded top fan-in and bottom fan-in that are not in the Sylvester–Gallai configuration, improving and simplifying a result of Gupta (ECCC TR 14-130).

As a complement of our explicit construction, we prove a lower bound for the size of k-subspace families that are evasive for degree-d varieties in a projective n-space. When $n - k = n^{\Omega(1)}$, the lower bound is superpolynomial unless d is bounded. The proof uses a dimension-counting argument on Chow varieties that parametrize projective subvarieties.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases algebraic complexity, dimension reduction, Noether normalization, polynomial identity testing, pseudorandomness, varieties

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.20

Related Version *Full Version*: https://arxiv.org/abs/2105.02908

Acknowledgements We thank Nitin Saxena, Noga Ron-Zewi, Amit Sinhababu, and Suryajith Chillara for helpful discussions. We also thank the anonymous reviewers of CCC'21 for their careful reading of our manuscript and their insightful suggestions that helped improve this paper.

1 Introduction

Polynomial identity testing (PIT) is a fundamental problem in the areas of derandomization and algebraic complexity theory. The problem asks whether a multivariate polynomial, computed by an arithmetic circuit, formula, or other algebraic computational models, is identically zero. For example, the polynomial $(X+Y)(X-Y)-X^2-Y^2$ is identically zero while $(X+Y)^2-X^2$ is not.

It is easy to solve PIT in randomized polynomial time, as we may simply evaluate the input polynomial at a random point and check if the evaluation is zero. On the other hand, finding a deterministic polynomial-time PIT algorithm for general arithmetic circuits is a long-standing open problem. Such algorithms are known for some special cases, and we refer the readers to the surveys [67, 68, 73] for details.

Black-box PIT algorithms are a special kind of PIT algorithm. A (deterministic) black-box PIT algorithm tests if a polynomial in a family $\mathcal{F}$ is zero by constructing a *hitting set* for $\mathcal{F}$, which is a finite collection $\mathcal{H}$ of evaluation points with the following property: for any nonzero $Q \in \mathcal{F}$, there exists $p \in \mathcal{H}$ such that the evaluation of Q at p is nonzero. After constructing such a hitting set $\mathcal{H}$, the algorithm simply checks if the evaluation of the given polynomial at every point in $\mathcal{H}$ is zero. The problem of designing a deterministic black-box PIT algorithm is thus equivalent to constructing a hitting set. To make the algorithm efficient, such a hitting set should be small and efficiently computable.

From a geometric perspective, an n-variate nonzero polynomial Q over an algebraically-closed field $\mathbb{F}$ defines a *hypersurface* $\mathcal{V}(Q) := \{\alpha \in \mathbb{F}^n : Q(\alpha) = 0\}$ of $\mathbb{F}^n$. A hitting set $\mathcal{H}$ for $\mathcal{F}$ has the property that for every nonzero $Q \in \mathcal{F}$, there exists a point $p \in \mathcal{H}$ that is disjoint from the hypersurface $\mathcal{V}(Q)$, or we say p *evades* $\mathcal{V}(Q)$. It is natural to consider the generalization of this property to higher dimensions/codimensions. Namely, we want to construct a finite collection $\mathcal{H}$ of *affine k-subspaces* (i.e. affine subspaces of dimension k) such that for every *variety* $\mathcal{V} \subseteq \mathbb{F}^n$ (i.e., solution set of a set of polynomial equations) from a certain family, some (or most) $W \in \mathcal{H}$ *evade* $\mathcal{V}$, in the sense that the dimension of the intersection $\mathcal{V} \cap W$ is bounded by the expected dimension achieved by W in general position. A similar property can be defined for projective k-spaces, to be defined below. We call such a collection $\mathcal{H}$ of projective or affine k-subspaces a *variety evasive subspace family.* The formal definition is given below.

1.1 Variety Evasive Subspace Families

Let $\mathbb{F}$ be an algebraically closed field. An *affine n-space* $\mathbb{A}^n$, as a set, is simply defined to be the vector space $\mathbb{F}^n$. We also need the notion of a *projective n-space*, denoted by $\mathbb{P}^n$, which is (intuitively) the set of lines passing through the origin $\mathbf{0}$ of $\mathbb{A}^{n+1}$. Formally, it is defined to be the quotient set $(\mathbb{A}^{n+1} \setminus \{\mathbf{0}\})/\sim$, where $\sim$ is the equivalence relation defined by scaling, i.e., $u \sim v$ if $u = cv$ for some nonzero scalar $c \in \mathbb{F}$.

An *(affine) subvariety* $\mathcal{V} \subseteq \mathbb{A}^n$ is the set of common zeros of a set of n-variate polynomials over $\mathbb{F}$. Similarly, a *(projective) subvariety* $\mathcal{V} \subseteq \mathbb{P}^n$ is the set of common zeros of a set of *homogeneous* $(n+1)$-variate polynomials over $\mathbb{F}$, where we represent each element of $\mathbb{P}^n$ as an $(n+1)$-tuple in $\mathbb{A}^{n+1}$. In this paper, a *variety* refers to a subvariety of a projective or affine space, and is said to be *irreducible* if it cannot be written as a union of finitely many proper subvarieties.[1]

The *dimension* of a variety $\mathcal{V}$, denoted by $\dim(\mathcal{V})$, is intuitively the "degree of freedom" of picking a point in the variety. See Subsection 2.3 for its formal definition. For a linear subspace $\mathcal{V} \subseteq \mathbb{A}^n$, the linear-algebraic dimension of $\mathcal{V}$ is the same as its dimension as a variety.

For two irreducible subvarieties $\mathcal{V}_1$ and $\mathcal{V}_2$ of $\mathbb{P}^n$ or $\mathbb{A}^n$ in *general position*, we expect the dimension of $\mathcal{V}_1 \cap \mathcal{V}_2$ to be $\dim(\mathcal{V}_1) + \dim(\mathcal{V}_2) - n$ (unless $\dim(\mathcal{V}_1) + \dim(\mathcal{V}_2) < n$, in which case we expect $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$). The following definition captures the condition that $\dim(\mathcal{V}_1 \cap \mathcal{V}_2)$ is bounded by the expected dimension.

▶ **Definition 1** (Evading). *Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be irreducible subvarieties of $\mathbb{P}^n$ or $\mathbb{A}^n$. We say $\mathcal{V}_1$* evades *$\mathcal{V}_2$ if*

$$\dim(\mathcal{V}_1 \cap \mathcal{V}_2) \leq \dim(\mathcal{V}_1) + \dim(\mathcal{V}_2) - n,$$

where the dimension of an empty set is assumed to be $-\infty$. In particular, if $\dim(\mathcal{V}_1) + \dim(\mathcal{V}_2) < n$, then $\mathcal{V}_1$ evades $\mathcal{V}_2$ iff $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$.

More generally, suppose $\mathcal{V}_1$ is irreducible but $\mathcal{V}_2$ is possibly reducible. We say $\mathcal{V}_1$ evades *$\mathcal{V}_2$ if it evades every irreducible component of $\mathcal{V}_2$.*

[1] Varieties in this paper are not necessarily irreducible and are often called *algebraic sets* in literature.

Next, we define *subspace families* and *variety evasive subspace families.*

▶ **Definition 2** (Subspace family)**.** *For $0 \le k \le n$, a finite collection*[2] *of k-subspaces of $\mathbb{P}^n$ is called a* (projective) k-subspace family *on $\mathbb{P}^n$. Similarly, a finite collection of affine k-subspaces of $\mathbb{A}^n$ is called an* affine k-subspace family *on $\mathbb{A}^n$.*

▶ **Definition 3** (Variety evasive subspace family)**.** *Let $\mathcal{F}$ be a family of subvarieties of $\mathbb{P}^n$ (resp. $\mathbb{A}^n$). Let $\mathcal{H}$ be a k-subspace family on $\mathbb{P}^n$ (resp. affine k-subspace family on $\mathbb{A}^n$) where $0 \le k \le n$. Then:*

- *We say $\mathcal{H}$ is* $\mathcal{F}$-evasive *if for every $\mathcal{V} \in \mathcal{F}$, there exists $W \in \mathcal{H}$ that evades $\mathcal{V}$.*
- *We say $\mathcal{H}$ is* $(\mathcal{F}, \epsilon)$-evasive *if for every $\mathcal{V} \in \mathcal{F}$, a random element $W \in \mathcal{H}$ evades $\mathcal{V}$ with probability at least $1 - \epsilon$.*

Connection with hitting sets. Definition 3 naturally generalizes the notions of hitting sets in the context of PIT. For example, a collection of points in $\mathbb{P}^n$ is a hitting set for a family $\mathcal{F}$ of homogeneous polynomials in $\mathbb{F}[X_1, \dots, X_{n+1}]$ iff it is an $\mathcal{F}'$-evasive 0-subspace family, where $\mathcal{F}' = \{\mathcal{V}(P) : P \in \mathcal{F}\}$ is the family of hypersurfaces defined by the polynomials in $\mathcal{F}$. In other words, hitting sets may be viewed as 0-subspace families that are evasive for varieties of codimension one.

Connection with lossless rank condensers. Other than the case of codimension one, we may also consider the special case of degree one, and this leads to another important family of pseudorandom objects, called *(weak) lossless rank condensers* [33, 29, 28, 27]. These objects were used by Gabizon and Raz [33] to construct affine extractors. They also play a crucial role in polynomial identity testing [51, 69, 29, 28].

A lossless rank condenser is defined as follows: Let $r \le t \le n$ be positive integers. A finite collection $\mathcal{H}$ of matrices $E \in \mathbb{F}^{t \times n}$ is called an (r, L)*-lossless rank condenser* if for every matrix $M \in \mathbb{F}^{n \times r}$ of rank r, the number of $E \in \mathcal{H}$ satisfying $\operatorname{rank}(EM) < r$ is at most L.

The connection between lossless rank condensers and variety evasive subspace families can be seen as follows: Let us assume every matrix $E \in \mathcal{H}$ has full rank t. Such a matrix E corresponds a linear t-subspace W of $\mathbb{F}^n$. On the other hand, a matrix $M \subseteq \mathbb{F}^{n \times r}$ of rank r corresponds to a linear $(n - r)$-subspaces of $\mathbb{F}^n$ via $M \mapsto \ker(M)$, where $\ker(M) = \{u \in \mathbb{F}^n : uM = 0\}$ denotes the left kernel of M. It is easy to see that the condition $\operatorname{rank}(EM) = r$ is equivalent to $\dim(W \cap \ker(M)) = t - r$. Passing from $\mathbb{F}^n$ to $\mathbb{P}^{n-1}$ by taking the quotient modulo scalars, this condition is also equivalent to the condition that the two projective subspaces $\mathbb{P}W$ and $\mathbb{P}(\ker(M))$ evade each other.

Every projective $(n-r-1)$-subspace of $\mathbb{P}^{n-1}$ can be realized as $\mathbb{P}(\ker(M))$ for some rank-r matrix M. Therefore, $\mathcal{H}$ is an (r, L)-lossless rank condenser iff it is an $(\mathcal{F}, \epsilon)$-evasive $(t - 1)$-subspace family on $\mathbb{P}^{n-1}$, where $\epsilon = L/|\mathcal{H}|$ and $\mathcal{F}$ is the family of all $(n - r - 1)$-subspaces of $\mathbb{P}^{n-1}$.

Rank condensers are central objects in the theory of "linear-algebraic pseudorandomness" coined by Guruswami and Forbes [27]. Our study of variety evasive subspace families may be seen as one step of extending the theory to a nonlinear setting.

Explicit lossless rank condensers were used to construct explicit (deterministic) *affine extractors* [33] and more generally, *extractors for varieties* [18]. Similar ideas were used to construct explicit *deterministic extractors* (and *rank extractors*) for *polynomial sources* [19],

[2] In this paper, a *collection* is a multiset, i.e., its elements are allowed to appear more than once.

which also generalize affine extractors. It is an interesting question to us whether explicit variety-evasive subspace families and the related derandomized Noether's normalization lemma (see below) can be similarly useful in this area.

1.2 Our Results

We have seen that variety evasive subspace families generalize some important and well-studied pseudorandom objects. This leads to the following natural question: For which interesting families $\mathcal{F}$ of subvarieties can we construct explicit $\mathcal{F}$-evasive or $(\mathcal{F}, \epsilon)$-evasive subspace families?

In this paper, we focus on the families of subvarieties of *bounded degree*. First, we recall the definition of the *degree* of a variety.

▶ **Definition 4** (degree). *The degree of an irreducible variety $\mathcal{V}$ in $\mathbb{P}^n$ (resp. $\mathbb{A}^n$) is the number of intersections of $\mathcal{V}$ with a general projective (resp. affine) subspace of codimension* $\dim(\mathcal{V})$. *Following [46], we define the degree of a (possibly reducible) variety to be the sum of the degrees of its irreducible components.*

For convenience, we introduce the following definition.

▶ **Definition 5.** *We say a projective (resp. affine) k-subspace family $\mathcal{H}$ on $\mathbb{P}^n$ (resp. $\mathbb{A}^n$) is* (n, d)-evasive *if it is $\mathcal{F}$-evasive, where $\mathcal{F}$ is chosen to be the family of all subvarieties of $\mathbb{P}^n$ (resp. $\mathbb{A}^n$) of degree at most d. Similarly, we say $\mathcal{H}$ is* (n, d, ϵ)-evasive *if it is $(\mathcal{F}, \epsilon)$-evasive.*

▶ Remark. In Definition 5, we do not make any assumption about the dimension of the varieties in $\mathcal{F}$ or their irreducible components. We will see in Subsection 3.1 that in fact, it suffices to consider the subfamily of equidimensional varieties or even irreducible varieties of dimension $n - k - 1$ when constructing variety evasive k-subspace families.

For $n, d \in \mathbb{N}^+$ and $k \in \{0, 1, \dots, n\}$, define $N(k, d, n)$ by

$$N(k,d,n) := \min\left\{\binom{(k+1)(n+1+d)}{(k+1)d}, \binom{(n-k)(n+1+d)}{(n-k)d}, \binom{(d-1)(n+1+d)}{(d-1)d}\right\}.$$

Our main theorem then states as follows.

▶ **Theorem 6** (Main Theorem). *For $n, d \in \mathbb{N}^+$, $k \in \{0, 1, \dots, n\}$, and $\epsilon \in (0, 1)$, there exists an (n, d, ϵ)-evasive k-subspace family (resp. affine k-subspace family) $\mathcal{H}$ on $\mathbb{P}^n$ (resp. $\mathbb{A}^n$) of size* $\mathrm{poly}(N(k, d, n), n, 1/\epsilon)$, *which is* $\mathrm{poly}(n^{\min\{k+1,n-k,d\}d}, 1/\epsilon)$ *when $d = o(n)$. Moreover, the total time complexity of computing the linear equations defining the projective or affine subspaces in $\mathcal{H}$ is polynomial in $|\mathcal{H}|$ (and $\log p$, if the characteristic of the base field $\mathbb{F}$ is $p > 0$). In particular, $\mathcal{H}$ can be constructed in polynomial time when d is bounded.*

▶ Remark (Boundedness of coefficients). For simplicity, the base field $\mathbb{F}$ in this paper is assumed to be an algebraically closed field. Nevertheless, we choose the coefficients of the linear equations defining the subspaces in $\mathcal{H}$ so that they live in either $\mathbb{Q}$ (if $\mathrm{char}(\mathbb{F}) = 0$) or a finite extension of $\mathbb{F}_p$ (if $\mathrm{char}(\mathbb{F}) = p > 0$). Moreover, when $\mathrm{char}(\mathbb{F}) = 0$, the bit-length of the numerators and denominators of these coefficients are bounded by $|\mathcal{H}|^{O(1)}$. And when $\mathrm{char}(\mathbb{F}) = p > 0$, the finite field that contains these coefficients has size $\max\{|\mathcal{H}|^{O(1)}, p\}$. This can be readily checked from our construction. Similar properties hold for all constructions presented in this paper.

Lower bound. As a complement of the above result, we establish the following lower bound for projective k-subspace families. It implies that when $n-k=n^{\Omega(1)}$, the assumption of d being bounded is necessary for a projective (n,d)-evasive k-subspace family to have polynomial size.

▶ **Theorem 7.** *Let $n,d\in\mathbb{N}^+$ and $k\in\{0,1,\dots,n-1\}$. Let $\mathcal{F}$ be the family of equidimensional projective subvarieties of $\mathbb{P}^n$ of dimension $n-k-1$ and degree at most d. Suppose $\mathcal{H}$ is an $\mathcal{F}$-evasive k-subspace family on $\mathbb{P}^n$. Then*

$$|\mathcal{H}|\ge\begin{cases}(n-k)(k+1)+1 & \text{if } d=1,\\ \max\left\{d(n-k)(k+1)+1,\binom{d+n-k}{d}+(n-k+1)k\right\} & \text{if } d>1.\end{cases}$$

In particular, $|\mathcal{H}|$ is superpolynomial in n when $n-k=\Omega(n)$ and $d=\omega(1)$.

When $d=1$, the lower bound $|\mathcal{H}|\ge(n-k)(k+1)+1$ in Theorem 7 is achieved by known explicit lossless rank condensers [29, 28, 26] (see Subsection 2.2). For general d, the lower bound in Theorem 7 is also tight and matched by non-explicit constructions. See Section 4 for a discussion.

Next, we list two applications of our Main Theorem (Theorem 6): derandomizing Noether's normalization lemma for varieties of bounded degree, and polynomial identity testing for a special family of depth-4 arithmetic circuits.

1.2.1 Derandomizing Noether's Normalization Lemma

Noether's normalization lemma, introduced by Noether [64], is an important result in commutative algebra and algebraic geometry with many applications. For example, it is used in the development of dimension theory and can be used to prove Grothendieck's generic freeness lemma [23]. It also has applications in computational algebraic geometry, e.g., computing the dimension of a projective variety [36, 35].

The usual geometric formulation of Noether's normalization lemma states that for any affine variety $\mathcal{V}\subseteq\mathbb{A}^n$ of dimension r, there exists a surjective *finite morphism* $\pi:\mathcal{V}\to\mathbb{A}^r$. (See Subsection 2.3 for the definition of finite morphisms.) Moreover, π may be chosen to be the restriction of a linear map $\mathbb{A}^n\to\mathbb{A}^r$.[3] There is also a related projective or graded version of the lemma, which states that for any projective variety $\mathcal{V}$ of dimension r, there exists a surjective finite morphism $\pi:\mathcal{V}\to\mathbb{P}^r$. A special form of this lemma goes back to Hilbert [48].

In these versions of Noether's normalization lemma, it can be shown that with high probability, a random linear map yields a valid finite morphism π, where "random" means the coefficients of the linear map are chosen randomly from a sufficiently large finite set $S\subseteq\mathbb{F}$. It is thus a natural question to derandomize the lemma.

Mulmuley [62] studied a form of Noether's normalization lemma and proved that derandomizing it is equivalent to a strengthened form of the black-box derandomization of PIT. There, the ambient projective space has exponential dimension and the problem is

[3] For simplicity, we assume the base field is algebraically closed and hence infinite. But the lemma and our derandomization are valid as long as the field is large enough, depending on the variety $\mathcal{V}$. Nagata [63] proved a version of the normalization lemma that is deterministic and does not require the base field to be sufficiently large, but the morphism he used is highly nonlinear. Due to the inductive nature of Nagata's argument, it only yields a multiply exponential degree bound for the polynomials that define the morphism. Bruce and Erman [9] proved an effective Noether normalization result over finite fields, which states that with high probability, a random tuple of degree-d polynomials over a finite field induces a valid finite morphism for large enough d satisfying a certain effective bound. We leave it as an open problem to derandomize their version of the normalization lemma.

constructing a finite morphism $\pi : \mathcal{V} \to \mathbb{P}^k$ with a *succinct* specification in deterministic polynomial time, where $k = \mathrm{poly}(\dim(\mathcal{V}))$ and $\mathcal{V}$ is an *explicit variety* [62]. This problem was later shown to be in PSPACE [31, 38]. The special case for the ring of matrix invariants under simultaneous conjugation was solved in quasipolynomial time by Forbes and Shpilka [30].

We consider Noether's normalization lemma in its original context and completely derandomize it for projective/affine varieties of bounded degree. The following two theorems summarize our results.

▶ **Theorem 8.** *Let $n, d \in \mathbb{N}^+$, $r \in \{0, 1, \ldots, n\}$, and $\epsilon \in (0, 1)$. There exists an explicit collection $\mathcal{L}$ of linear maps $\mathbb{A}^{n+1} \to \mathbb{A}^{r+1}$ of size $\mathrm{poly}(N(k, d, n), n, 1/\epsilon)$ such that for every subvariety $\mathcal{V} \subseteq \mathbb{P}^n$ of dimension r and degree at most d, all but at most ϵ-fraction of $\pi \in \mathcal{L}$ induce a surjective finite morphism from $\mathcal{V}$ to $\mathbb{P}^r$. Moreover, $\mathcal{L}$ can be computed in time polynomial in $|\mathcal{L}|$ (and $\log p$, if $\mathrm{char}(\mathbb{F}) = p > 0$).*

▶ **Theorem 9.** *Let $n, d \in \mathbb{N}^+$ and $r \in \{0, 1, \ldots, n\}$, and $\epsilon \in (0, 1)$. There exists an explicit collection $\mathcal{L}$ of linear maps $\mathbb{A}^n \to \mathbb{A}^r$ of size $\mathrm{poly}(N(k, d, n), n, 1/\epsilon)$ such that for every subvariety $\mathcal{V} \subseteq \mathbb{A}^n$ of dimension r and degree at most d, all but at most ϵ-fraction of $\pi \in \mathcal{L}$ restrict to a surjective finite morphism from $\mathcal{V}$ to $\mathbb{A}^r$. Moreover, $\mathcal{L}$ can be computed in time polynomial in $|\mathcal{L}|$ (and $\log p$, if $\mathrm{char}(\mathbb{F}) = p > 0$).*

Theorem 8 is proved by derandomizing a standard proof of Noether's normalization lemma that has a geometric flavor [71]. Namely, we consider a projection $\pi : \mathbb{P}^n \setminus W \to \mathbb{P}^r$ sending $\mathbf{x}$ to $(\ell_1(\mathbf{x}), \cdots, \ell_{r+1}(\mathbf{x}))$, where $\ell_1, \ldots, \ell_r$ are linear forms and W is the $(n - r - 1)$-subspace where these linear forms simultaneously vanish. It is known that π restricts to a finite morphism $\mathcal{V} \to \mathbb{P}^r$ iff $W \cap \mathcal{V} = \emptyset$. So the problem reduces to choosing a family of $(n - r - 1)$-subspaces of $\mathbb{P}^n$ such that most of them are disjoint from $\mathcal{V}$. This is exactly the property satisfied by our explicit variety evasive subspace families.

Theorem 9 is proved similarly. Here $\mathbb{A}^n$ is viewed as an open subset of $\mathbb{P}^n$ whose complement is the "hyperplane at infinity" H_∞. Then we first construct a projection $\pi : \mathbb{P}^n \setminus W \to \mathbb{P}^r$ such that W is a subspace of H_∞ and is disjoint from the *projective closure* of $\mathcal{V}$. Then restrict π to $\mathbb{A}^n$. By carefully choosing π, we can make sure that the restriction is a linear map $\mathbb{A}^n \to \mathbb{A}^r$ and is a surjective finite morphism.

Dimension-preserving morphisms vs. finite morphisms. Our construction of finite linear morphisms preserve the dimension of a variety of low degree while reducing the dimension of the ambient space. This generalizes the property of lossless rank condensers. However, for the dimension-preserving property, better constructions are known. For example, it follows implicitly from the proof in [18] that most of the linear maps $\mathbb{A}^n \to \mathbb{A}^t$ from a lossless rank condenser $\mathcal{H} \subseteq \mathbb{F}^{t \times n}$ already preserve the dimension of a variety $\mathcal{V} \subseteq \mathbb{A}^n$.[4] This was used by Dvir [18] in his explicit constructions of *extractors for varieties*, which generalize *affine extractors* [33].

On the other hand, the morphisms we construct are *finite morphisms*, which are strictly stronger than morphisms that are dimension-preserving. In particular, a finite morphism π always maps a closed set onto a closed set in the Zariski topology. Moreover, the preimage $\pi^{-1}(p)$ of *every* point p in the image of π is a finite set. Neither of these two properties is necessarily satisfied by morphisms that are only dimension-preserving.

[4] The intuition here is that $\mathcal{V}$ can be locally approximated at a nonsingular point $p \in \mathcal{V}$ by its *tangent space* at p. So any linear map that preserves the dimension of this tangent space also preserves the dimension of $\mathcal{V}$.

These properties of finite morphisms may be useful in extractor theory or other areas. For example, in Theorem 9, the cardinality of $\pi^{-1}(p)$ is bounded by the degree of $\mathcal{V}$ for every $p \in \pi(\mathcal{V})$, which translates into a lower bound for the min-entropy of the output of π when the input random source is distributed over the variety $\mathcal{V}$.

1.2.2 Depth-4 Polynomial Identity Testing

Depth-4 arithmetic circuits, also known as $\Sigma\Pi\Sigma\Pi$ circuits, play a very important role in polynomial identity testing. In a surprising result, Agrawal and Vinay [3] proved that a complete derandomization of black-box PIT for depth-4 circuits implies an $n^{O(\log n)}$-time derandomization of PIT for general circuits of $\mathrm{poly}(n)$ degree.

Dvir and Shpilka [22] initialized the approach of applying *Sylvester–Gallai* type theorems in geometry to PIT for depth-3 ($\Sigma\Pi\Sigma$) circuits. Extending this approach, Gupta [39] formulated a conjecture of Sylvester–Gallai type and proved that his conjecture implies a complete derandomization of black-box PIT for depth-4 circuits with bounded top fan-in and bottom fan-in (also called $\Sigma\Pi\Sigma\Pi(k, r)$ circuits, where $k, r = O(1)$). In a recent breakthrough (built on [72, 65]), Peleg and Shpilka [66] proved that this conjecture holds for $k = 3$ and $r = 2$, and used it to give a polynomial-time black-box PIT algorithm for $\Sigma\Pi\Sigma\Pi(3, 2)$ circuits.

In [39], Gupta divided $\Sigma\Pi\Sigma\Pi(k, r)$ into two families: those in a certain Sylvester–Gallai configuration and those that are not. His conjecture states that the circuits in the first family always have bounded *transcendence degree*, depending only on k and r. If the conjecture is true, then the results in [6, 2] imply a complete derandomization of the black-box PIT for this family. For the second family of circuits, which we call *non-SG* circuits, he proved that the black-box PIT can also be derandomized completely.

▶ **Theorem 10** ([39]). *There exists a deterministic black-box PIT algorithm with time complexity $(dnk)^{\mathrm{poly}(r^{k^2}+k)}$ for non-SG $\Sigma\Pi\Sigma\Pi(k, r)$ circuits of degree at most d in $X_1, \ldots, X_n$ over . In particular, the algorithm runs in polynomial time when k and r are bounded.*

Gupta's proof of Theorem 10 is quite complex and used tools from computational algebraic geometry, including an effective version of Bertini irreducibility theorem [47] and radical membership testing (which in turn depends on *effective Nullstellensatz* [53, 17]).

We observe that what is needed here is simply an explicit construction of subspaces intersecting certain varieties with (at most) the expected dimension. Plugging in our explicit construction of variety evasive subspace families, we obtain an improved black-box PIT algorithm with a simple proof.

▶ **Theorem 11.** *There exists a deterministic black-box PIT algorithm with time complexity polynomial in $d \cdot \binom{k(n+1+r^k)}{kr^k} \cdot \binom{k-1+d}{k-1} \le \mathrm{poly}(d^k, n^{r^k}, r^{k^2 r^k})$ (and $\log p$, if $\mathrm{char}(\mathbb{F}) = p > 0$) for non-SG $\Sigma\Pi\Sigma\Pi(k, r)$ circuits of degree at most d in $X_1, \ldots, X_n$ over an algebraically closed field $\mathbb{F}$.*

In particular, Theorem 11 improves the exponent of n in the time complexity from $\mathrm{poly}(r^{k^2} + k)$ to $O(r^k)$, and the exponent of d from $\mathrm{poly}(r^{k^2} + k)$ to $O(k)$. Moreover, our proof is more direct and conceptually simpler than the proof in [39].

▶ Remark. In [61], Mukhopadhyay gave a deterministic polynomial-time black-box PIT algorithm for $\Sigma\Pi\Sigma\Pi(k, r)$ circuits satisfying a variant of the non-SG assumption. (Its time complexity is similar to the time complexity in Theorem 10.) It appears to us that his assumption in fact implies the non-SG assumption. The main tool used there is the *multivariate resultant*, which may be related to our approach based on Chow forms (see Subsection 1.3). Indeed, it is known that a multivariate resultant is the Chow form of a Veronese variety [34, Chapter 3, Example 2.4].

1.3 Proof Overview

We present an overview of our proof of Theorem 6 and that of Theorem 7.

Overview of the proof of Theorem 6

In the proof of Theorem 6, we focus on constructing a k-subspace family on $\mathbb{P}^n$. The case of $\mathbb{A}^n$ can be easily derived from it by viewing $\mathbb{A}^n$ as an open subset of $\mathbb{P}^n$ and restricting to this subset.

Consider a variety $\mathcal{V} \subseteq \mathbb{P}^n$ of degree at most d. We want to construct a k-subspace family $\mathcal{H}$ on $\mathbb{P}^n$, independent of $\mathcal{V}$, such that all but at most ϵ-fraction of $W \in \mathcal{H}$ evade $\mathcal{V}$. Our key ideas can be summarized as follows.

Reducing to the equidimensional/irreducible case of dimension $n-k-1$. As a first step, we reduce the problem to the special case that $\mathcal{V}$ is an *equidimensional* (or even irreducible) variety of $\mathbb{P}^n$ of dimension $n-k-1$, which means every irreducible component of $\mathcal{V}$ has dimension exactly $n-k-1$. This step is explained in Subsection 3.1.

Hitting the Chow form of $\mathcal{V}$. Denote by $\mathbb{G}(k,n)$ the Grassmannian consisting of of all k-subspaces of $\mathbb{P}^n$. As $\operatorname{codim}(\mathcal{V}) = n-(n-k-1) > k$, a *general* k-subspace $W \in \mathbb{G}(k,n)$ is disjoint from $\mathcal{V}$, but we want to find such W explicitly.

One remarkable fact in algebraic geometry is that there is a single polynomial $\widetilde{R}_{\mathcal{V}}$ on the Grassmannian $\mathbb{G}(k,n)$ that defines precisely the subset of k-subspaces that intersect $\mathcal{V}$. This polynomial $\widetilde{R}_{\mathcal{V}}$ is called the *Chow form* of $\mathcal{V}$ (in *Stiefel coordinates*). Chow forms are also known as *Cayley forms* or *Cayley–van der Waerden–Chow forms* in literature. They were introduced by Cayley [11] to represent curves in $\mathbb{P}^3$ and later generalized by Chow and van der Waerden [13]. See [15] for an introduction to Chow forms and [34] for an exposition in the context of elimination theory.

To be more specific, for a k-subspace $W \in \mathbb{G}(k,n)$, we choose a $(k+1)\times(n+1)$ matrix A that represents W. The Chow form $\widetilde{R}_{\mathcal{V}}$ is a polynomial of degree $(k+1)\deg(\mathcal{V})$ in $(k+1)(n+1)$ variables with the following property: $\widetilde{R}_{\mathcal{V}}$ vanishes at the matrix A (viewed as a list of $(k+1)(n+1)$ coordinates) if and only if $\mathcal{V}\cap W \neq \emptyset$. Thus, $\widetilde{R}_{\mathcal{V}}$ defines precisely the subset of "bad" k-subspaces that we want to avoid.

Therefore, the problem becomes finding a collection of $(k+1)\times(n+1)$ matrices of full rank that "hit" the polynomial $\widetilde{R}_{\mathcal{V}}$ of degree $(k+1)\deg(\mathcal{V}) \le (k+1)d$. Using black-box PIT for low degree polynomials (see Subsection 2.1), we are able to construct an (n,d,ϵ)-evasive k-subspace family of size polynomial in $\binom{(k+1)(n+1+d)}{(k+1)d}$ and $1/\epsilon$, which is $\operatorname{poly}(n,1/\epsilon)$ when k and d are both bounded. A similar "dual" construction yields a k-subspace family of size polynomial in $\binom{(n-k)(n+1+d)}{(n-k)d}$ and $1/\epsilon$, which is $\operatorname{poly}(n,1/\epsilon)$ when both $n-k$ and d are bounded. For applications where d is small and either k or $n-k$ is small (e.g., Theorem 11), these constructions are good enough. However, when k and $n-k$ are both linear in n, the resulting k-subspace families have exponential size in n, even if d is bounded.

A two-step construction. To obtain a good construction for *arbitrary* dimension k, we use a standard fact from algebraic geometry, which states that the codimension of an irreducible subvariety $\mathcal{V} \subseteq \mathbb{P}^n$ in $\operatorname{span}(\mathcal{V})$ is at most $\deg(\mathcal{V})-1$, where $\operatorname{span}(\mathcal{V})$ denotes the smallest projective subspace containing $\mathcal{V}$ (see Lemma 32). Therefore, for irreducible $\mathcal{V}$ of degree at most d, there exists a projective subspace Λ of dimension (at most) $\dim(\mathcal{V})+d-1$ that contains $\mathcal{V}$.

Our idea is to use a two-step construction. Namely, we first construct subspaces of dimension $n - \dim \Lambda - 1$ that evade Λ, and then extend these subspaces to k-subspaces that evade $\mathcal{V}$. The first step is just the problem of constructing lossless rank condensers, which has an optimal solution [29, 28] (see Subsection 2.2). The second step is equivalent to extending a $((k+1)-(d-1)) \times (n+1)$ matrix B to a $(k+1) \times (n+1)$ matrix $\binom{A}{B}$ such that the polynomial $\widetilde{R}_{\mathcal{V}}$ does not vanish at $\binom{A}{B}$. The polynomial $\widetilde{R}_{\mathcal{V}}(\binom{\cdot}{B})$ has degree $(d-1)\deg(\mathcal{V}) \leq (d-1)d$, as there are only $d-1$ rows of free variables. Using black-box PIT for low degree polynomials, we obtain a construction of size polynomial in $\binom{(d-1)(n+1+d)}{(d-1)d}$ and $1/\epsilon$, which is $\mathrm{poly}(n, 1/\epsilon)$ for any bounded d.

Overview of the proof of Theorem 7

Our lower bound (Theorem 7) follows from a dimension counting argument. Let $C(r,d,n)$ be the set of all varieties $\mathcal{V} \subseteq \mathbb{P}^n$ of dimension $r := n-k-1$ and degree d, which is the space of varieties that we want to evade.

Roughly speaking, the idea is to show that (1) $C(r,d,n)$ itself can be realized as a subvariety of some projective space $\mathbb{P}^N$, and (2) for every k-subspace W, the subset of $\mathcal{V} \in C(r,d,n)$ that W fails to evade is the intersection of $C(r,d,n)$ with some hyperplane H_W of $\mathbb{P}^N$.

To see how (1) and (2) above lead to a lower bound, suppose $\mathcal{H}$ is a $C(r,d,n)$-evasive k-subspace family, i.e., for any $\mathcal{V} \in C(r,d,n)$, there exists $W \in \mathcal{H}$ that is disjoint from $\mathcal{V}$. Then the intersection $C(r,d,n) \cap \bigcap_{W \in \mathcal{H}} H_W$ must be empty. On the other hand, taking the intersection with each hyperplane H_W reduces the dimension of a projective variety by at most one. So we have a lower bound $|\mathcal{H}| \geq \dim(C(r,d,n)) + 1$.

How do we realize $C(r,d,n)$ as a subvariety of $\mathbb{P}^N$? It turns out that this is a classical problem in the study of moduli spaces and a solution was given by Cayley [11] and Chow–van der Waerden [13] using the *Chow embedding*: The Chow embedding $C(r,d,n) \to \mathbb{P}^N$ simply sends a variety $\mathcal{V}$ to its Chow form $\widetilde{R}_{\mathcal{V}}$, where $\widetilde{R}_{\mathcal{V}}$ is viewed as a point in the projective space $\mathbb{P}^N$ whose homogeneous coordinates are given by the coefficients of $\widetilde{R}_{\mathcal{V}}$.[5]

A technical issue here is that the image of $C(r,d,n)$ under the Chow embedding is generally not closed in the Zariski topology. To fix this issue, the definition of $C(r,d,n)$ needs to be modified so that it contains not only subvarieties of $\mathbb{P}^n$, but also *(effective) algebraic cycles* on $\mathbb{P}^n$, which are a generalization of subvarieties. A theorem of Chow and van der Waerden [13] then states that the Chow embedding does embed $C(r,d,n)$ in a projective subspace $\mathbb{P}^N$ as a subvariety, known as a *Chow variety.*

Finally, we also need a lower bound for the dimension of the Chow variety $C(r,d,n)$. In fact, the exact value of $\dim(C(r,d,n))$ was determined by Azcue [5] and independently by Lehmann [59]. Plugging in the value of $\dim(C(r,d,n))$ proves Theorem 7.

1.4 Other Related Work

In [20], Dvir, and Kollár, and Lovett constructed explicit *variety evasive sets*, which are large subsets of $\mathbb{F}_q^n$ over a finite field $\mathbb{F}_q$ that have small intersection with affine varieties of fixed dimension and bounded degree. It generalizes an earlier construction of *subspace evasive sets* of Dvir and Lovett [21]. The definition of evasiveness there is different from ours, but they are related, since a key step in the proofs of [21, 20] is proving the intersection of two

[5] The actual Chow embedding we use has a slightly different form, which is essentially equivalent to the one described here.

varieties has dimension zero. We also note that a subspace/variety evasive set is a single set, defined in a highly nonlinear way, whereas we define a variety evasive subspace family to be a collection of projective or affine subspaces. Finally, the results in [21, 20] hold only for affine subspaces/subvarieties, whereas we give our construction first in the projective setting and then derive the affine counterpart from it.

Guruswami and Xing in [43] introduced a related notion called *subspace designs*. A subspace design is a collection $\mathcal{H}$ of large subspaces of $\mathbb{F}^n$ such that for any small subspace $V \subseteq \mathbb{F}^n$, the number of $W \in \mathcal{H}$ satisfying $\dim(W \cap V) > 0$ is small (or even the sum $\sum_{W \in \mathcal{H}} \dim(W \cap V)$ is small). An equivalence between subspace designs and lossless rank condensers was proved in [27]. Explicit subspace designs were constructed by Guruswami and Kopparty [40] and also by Guruswami, Xing, and Yuan [44]. They have applications to constructing explicit list-decodable codes with small list size [43, 42, 55, 37] and explicit dimension expanders [27, 41]. Subspace designs were also used to prove lower bounds in communication complexity [12].

Jeronimo, Krick, Sabia, and Sombra [49] gave a randomized algorithm, in the Blum-Shub-Smale model over fields of characteristic zero, that computes the Chow forms of varieties defined by input polynomials. The (expected) time complexity of their algorithm is polynomial in the sizes of the arithmetic circuits encoding the input polynomials and the *geometric degree* of the polynomial system. See also the survey by Krick [56].

Chow varieties of effective zero-cycles and their higher secant varieties are related to lower bounds for depth-3 arithmetic circuits. They have received a considerable amount of attention in Geometric Complexity Theory [57, 58].

Organization of the paper. Preliminaries and notations are given in Section 2. We prove the Main Theorem (Theorem 6) in Section 3. The lower bound (Theorem 7) is proved in Section 4. The applications to the derandomization of Noether's normalization lemma and PIT for depth-4 circuits are explained in Section 5. Finally, we list some open problems and future directions in Section 6.

2 Preliminaries and Notations

Define $\mathbb{N} := \{0, 1, 2 \dots\}$ and $\mathbb{N}^+ := \{1, 2, \dots\}$. Let $[n] := \{1, 2, \dots, n\}$ for $n \in \mathbb{N}$. For a set S and $k \in \mathbb{N}$, denote by $\binom{S}{k}$ the set of all subsets of S of cardinality k.

Denote by $\mathbb{F}$ an algebraically closed field throughout this paper. We use notations like $\mathbb{F}[X_{i,j} : i \in [n], j \in [m]]$ to denote the polynomial ring over $\mathbb{F}$ in a finite set of variables (in this case, in the set of variables $\{X_{i,j} : i \in [n], j \in [m]\}$). The vector space of $n \times m$ matrices over $\mathbb{F}$ is denoted by $\mathbb{F}^{n \times m}$.

For an $n \times m$ matrix A and subsets $S \subseteq [n]$, $T \subseteq [m]$, denote by $A_{S,T}$ the submatrix of A whose rows and columns are selected by S and T respectively, where the orderings of rows and columns are preserved.

2.1 Black-Box PIT for Low Degree Polynomials

For convenience, we strengthen the definition of hitting sets as follows.

▶ **Definition 12** (ϵ-hitting set). *Let $\mathcal{F}$ be a family of polynomials in $\mathbb{F}[X_1, \dots, X_n]$ and $\epsilon \in (0, 1)$. We say a finite collections of points $\mathcal{H} \subseteq \mathbb{F}^n$ is an* ϵ-hitting set *for $\mathcal{F}$ if for any nonzero $Q \in \mathcal{F}$, the evaluation $Q(\alpha)$ is nonzero for all but at most ϵ-fraction of $\alpha \in \mathcal{H}$.*

We need an explicit construction of ϵ-hitting sets for low degree polynomials. This problem has been well studied [16, 74, 70, 52, 8, 60, 14, 10, 7]. For completeness, we present a construction based on sparse polynomial identity testing.

Recall that a polynomial is *s-sparse* if it has at most s monomials. We need the following lemma from [1].

▶ **Lemma 13** ([1, Lemma 4, restated]). *For $n, s, d \in \mathbb{N}^+$ and $\epsilon_0 \in (0, 1)$, there exist maps $w_1, w_2, \ldots, w_N : [n] \to [N \log N]$, where $N = \text{poly}(n, s, \log d, \epsilon_0^{-1})$, such that for any nonzero s-sparse polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$ of individual degree at most d, all but at most ϵ_0-fraction of w_i among $w_1, w_2, \ldots, w_N$ satisfies $f(Y^{w_i(1)}, \ldots, Y^{w_i(n)}) \neq 0$. Moreover, the time complexity of computing $w_1, w_2, \ldots, w_N$ is polynomial in N.*

Given $n, d \in \mathbb{N}^+$ and $\epsilon \in (0, 1)$, we construct an ϵ-hitting set for n-variate polynomials of degree at most d as follows:

1. Let $s = \binom{n+d}{d}$, $\epsilon_0 = \epsilon/2$, and $M = \lceil \epsilon_0^{-1} dN \log N \rceil$, where N is as in Lemma 13.
2. Let $w_1, \ldots, w_N$ be as in Lemma 13, which can be computed in time $\text{poly}(N)$.
3. If $\text{char}(\mathbb{F}) = 0$, let $S = [M] \subseteq \mathbb{Z} \subseteq \mathbb{F}$. If $\text{char}(\mathbb{F}) = p > 0$, choose a finite extension $\mathbb{F}_q$ of $\mathbb{F}_p$ such that $M \leq q = \text{poly}(M, p)$, and choose S to be a subset of $\mathbb{F}_q \subseteq \mathbb{F}$ of cardinality M.
4. Finally, construct the following collection of points in $\mathbb{F}^n$ of size MN

$$T = \{(\alpha^{w_i(1)}, \ldots, \alpha^{w_i(n)}) : \alpha \in S, i \in [N]\} \subseteq \mathbb{F}^n.$$

▶ **Lemma 14.** *For any nonzero polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$ of degree at most d, we have $f(u) \neq 0$ for all but at most ϵ-fraction of $u \in T$. The collection T has cardinality $\text{poly}\left(\binom{n+d}{d}, 1/\epsilon\right)$ and can be computed in time $\text{poly}(|T|)$.*

Proof. Let $f \in \mathbb{F}[X_1, \ldots, X_n]$ be a nonzero polynomial of degree at most d. Note that f is trivially s-sparse, where $s = \binom{n+d}{d}$. So by Lemma 13, for all but at most ϵ_0-fraction of $i \in [N]$, we have $\widetilde{f}_i := f(Y^{w_i(1)}, \ldots, Y^{w_i(n)}) \neq 0$. Consider $i \in [N]$ such that $\widetilde{f}_i \neq 0$. Note that $\widetilde{f}_i$ is a univariate polynomial of degree at most $dN \log N$. So it has at most $dN \log N \leq \epsilon_0 M$ zeros. Therefore, by the choice of M, we have $f(\alpha^{w_i(1)}, \ldots, \alpha^{w_i(n)}) = \widetilde{f}_i(\alpha) \neq 0$ for all but at most ϵ_0-fraction of $\alpha \in S$. It follows that $f(u) \neq 0$ holds for all but at most ϵ-fraction of $u \in T$, as claimed. The rest of the lemma follows easily from the construction. ◀

Note that the seed length required to choose a random element in T is $\log |T| = O(\log \binom{n+d}{d} + \log(1/\epsilon))$, which is optimal up to a constant factor. We have made no effort to optimize the constant hidden in $O(\cdot)$. Interested readers may find the state-of-the-art result in [7], which achieves the optimal constant, at least for $d = o(n)$.

2.2 Explicit Lossless Rank Condensers

We need the following lemma in the context of *lossless rank condensers*. The construction in the lemma was given by Forbes and Shpilka [29] and the lemma itself follows implicitly from the analysis of Forbes, Saptharishi, and Shpilka in [28]. It was also stated explicitly in [26, Theorem 5.4.3].

▶ **Lemma 15** ([28, 26]). *Let $n \in \mathbb{N}^+$ and $r \in [n]$. Let $\omega \in \mathbb{F}^\times$ such that the multiplicative order of ω is at least n. Define the $r \times n$ matrix $W = (w_{i,j})_{i \in [r], j \in [n]}$ over $\mathbb{F}[X]$ by*

$$w_{i,j} = (\omega^{i-1} X)^{j-1}.$$

Then for every $n \times r$ matrix M over $\mathbb{F}$ of rank r, the polynomial $\det(WM) \in \mathbb{F}[X]$ is nonzero and has degree at most $r(n - r)$ after dividing out powers of X.

▶ **Corollary 16.** *Let n, r, W be as in Lemma 15 and $\epsilon \in (0,1)$. Let $S \subseteq \mathbb{F}^\times$ be a finite set of cardinality at least $r(n-r)/\epsilon$. For every $n \times r$ matrix M over $\mathbb{F}$ of rank r, we have* $\operatorname{rank}(W(\alpha)M) = r$ *for all but at most ϵ-fraction of $\alpha \in S$, where $W(\alpha)$ denotes the matrix $(w_{i,j}(\alpha))_{i\in[r],j\in[n]}$ over $\mathbb{F}$.*

Corollary 16 states that the collection $\{W(\alpha) : \alpha \in S\}$ of matrices is a (weak) $(r, \epsilon|S|)$-*lossless rank condenser*, as defined in [27]. Note that for each $\alpha \in S$, we have $\operatorname{rank}(W(\alpha)) = r$ and hence $W(\alpha)$ correspond to an $(r-1)$-subspace $U_{W(\alpha)}$ of $\mathbb{P}^{n-1}$. As explained in the introduction, the collection $\mathcal{H} = \{U_{W(\alpha)} : \alpha \in S\}$ is an $(\mathcal{F}, \epsilon)$-evasive $(r-1)$-subspace family on $\mathbb{P}^{n-1}$, where $\mathcal{F}$ is the family of $(n-r-1)$-subspaces of $\mathbb{P}^{n-1}$. Choosing S of size $r(n-r)+1$ and $\epsilon = 1 - \frac{1}{r(n-r)+1}$ shows that the lower bound in Theorem 7 is achieved when $d = 1$.

2.3 Preliminaries on Algebraic Geometry

We list basic preliminaries and notations on algebraic geometry used in this paper. One can also refer to a standard text, e.g., [71, 45].

Affine and projective spaces. For $n \in \mathbb{N}$, write $\mathbb{A}^n$ for the *affine n-space* over $\mathbb{F}$. It is defined to be the set $\mathbb{F}^n$ equipped with the *Zariski topology*, defined as follows: A subset $S \subseteq \mathbb{A}^n$ is *(Zariski-)closed* if it is the set of common zeros of a set of polynomials in $\mathbb{F}[X_1, \dots, X_n]$. The complement of a closed set is an *open* set. The origin of an affine space is denoted by $\mathbf{0}$.

Write $\mathbb{P}^n$ for the *(projective) n-space* over $\mathbb{F}$, defined to be the quotient set $(\mathbb{A}^{n+1} \setminus \{\mathbf{0}\})/\sim$, where $\sim$ is the equivalence relation defined by scaling, i.e., $u \sim v$ if $u = cv$ for some $c \in \mathbb{F}^\times$. The set $\mathbb{P}^n$ is again equipped with the *Zariski topology*, where a subset is closed if it is the set of common zeros of a set of *homogeneous* polynomials in $\mathbb{F}[X_1, \dots, X_{n+1}]$. We use $(n+1)$-tuples $(x_1, \dots, x_{n+1})$ to represent points in $\mathbb{P}^n$, called *homogeneous coordinates.*

For a vector space V over $\mathbb{F}$ of dimension $n+1$, where $n \in \mathbb{N}$, define the projective space $\mathbb{P}V = (V \setminus \{\mathbf{0}\})/\sim$, where $\sim$ is again the equivalence relation defined by scaling. By fixing a coordinate system of V and identifying it with $\mathbb{A}^{n+1}$, we may identify $\mathbb{P}V$ with $\mathbb{P}^n$.

Varieties. *Varieties* in this paper refer to either projective or affine varieties. A *projective (resp. affine) variety* is simply a closed subset of a projective (resp. affine) subspace. If $\mathcal{V}_1$ and $\mathcal{V}_2$ are closed subsets of a projective or affine space and $\mathcal{V}_1 \subseteq \mathcal{V}_2$, we say $\mathcal{V}_1$ is a *subvariety* of $\mathcal{V}_2$.

A variety is *reducible* if it is the union of finitely many proper subvarieties, and otherwise *irreducible.* Affine and projective spaces are irreducible. A variety $\mathcal{V}$ can be uniquely written as the union of finitely many irreducible varieties, which are called the *irreducible components* of $\mathcal{V}$.

A projective or affine variety is called a *hypersurface* (resp. *hyperplane*) if it is definable by a single polynomial (resp. single linear polynomial).

Hilbert's Nullstellensatz. An ideal I of a commutative ring R is *radical* if $a^m \in I$ implies $a \in I$ for every $a \in R$ and $m \in \mathbb{N}^+$. For an ideal I of $\mathbb{F}[X_1, \dots, X_n]$, denote by $\mathcal{V}(I)$ the subvariety of $\mathbb{A}^n$ defined by the polynomial in I. Define $\mathcal{V}(f_1, \dots, f_k) = \mathcal{V}(\langle f_1, \dots, f_k \rangle)$ for $f_1, \dots, f_k \in \mathbb{F}[X_1, \dots, X_n]$. For a subvariety $\mathcal{V}$ of $\mathbb{A}^n$, denote by $I(\mathcal{V})$ the ideal of $\mathbb{F}[X_1, \dots, X_n]$ consisting of all the polynomials vanishing on $\mathcal{V}$. *Hilbert's Nullstellensatz* states that the map $\mathcal{V} \mapsto I(\mathcal{V})$ is an inclusion-reversing one-to-one correspondence between the subvarieties of $\mathbb{A}^n$ and the radical ideals of $\mathbb{F}[X_1, \dots, X_n]$, with the inverse map $I \mapsto \mathcal{V}(I)$.

For a subvariety $\mathcal{V}$ of $\mathbb{A}^n$, define $\mathbb{F}[\mathcal{V}] := \mathbb{F}[X_1, \ldots, X_n]/I(\mathcal{V})$, called the *coordinate ring* of $\mathcal{V}$.

Projective Nullstellensatz. Consider the polynomial ring $R = \mathbb{F}[X_1, \ldots, X_{n+1}]$. It can be written as a direct sum $R = \bigoplus_{d=0}^{\infty} R_d$ where each R_d denotes the space of degree-d homogeneous polynomials, called the *homogeneous part of degree d* of R or simply the *degree-d part* of R. For an ideal I of R and $d \in \mathbb{N}$, let $I_d := I \cap R_d$, called the *degree-d part* of I. We say I is a *homogeneous ideal* if $I = \bigoplus_{d=0}^{\infty} I_d$. For a homogeneous ideal I of R, we have $R/I = \bigoplus_{d=0}^{\infty}(R/I)_d$ where $(R/I)_d := R_d/I_d$.

For a homogeneous ideal I of R, denote by $\mathcal{V}(I)$ the subvariety of $\mathbb{P}^n$ defined by the homogeneous polynomials in I. Define $\mathcal{V}(f_1, \ldots, f_k) = \mathcal{V}(\langle f_1, \ldots, f_k \rangle)$ for homogeneous polynomials $f_1, \ldots, f_k \in R$. For a subvariety $\mathcal{V}$ of $\mathbb{P}^n$, denote by $I(\mathcal{V})$ the ideal generated by the homogeneous polynomials vanishing on $\mathcal{V}$, which is a homogeneous ideal. The *projective Nullstellensatz* states that the map $\mathcal{V} \mapsto I(\mathcal{V})$ is an inclusion-reversing one-to-one correspondence between the nonempty subvarieties of $\mathbb{P}^n$ and the radical homogeneous ideals of R properly contained in $\langle X_1, \ldots, X_{n+1} \rangle$, with the inverse map $I \mapsto \mathcal{V}(I)$.

For a subvariety $\mathcal{V} \subseteq \mathbb{P}^n$ and the corresponding homogeneous ideal $I = I(\mathcal{V})$, we say R/I is the *homogeneous coordinate ring* of $\mathcal{V}$.

Morphisms. Let $\mathcal{V}_1 \subseteq \mathbb{A}^n$ and $\mathcal{V}_2 \subseteq \mathbb{A}^m$ be affine varieties. A *morphism* from $\mathcal{V}_1$ to $\mathcal{V}_2$ is a map $f : \mathcal{V}_1 \to \mathcal{V}_2$ that is a restriction of a polynomial map $\mathbb{A}^n \to \mathbb{A}^m$. Such a morphism f is associated with a ring homomorphism $f^\sharp : \mathbb{F}[\mathcal{V}_2] \to \mathbb{F}[\mathcal{V}_1]$, making $\mathbb{F}[\mathcal{V}_1]$ an algebra over $\mathbb{F}[\mathcal{V}_2]$. We say f is *finite* if $\mathbb{F}[\mathcal{V}_1]$ is finitely generated as an $\mathbb{F}[\mathcal{V}_2]$-module.

Let $f : \mathcal{V}_1 \to \mathcal{V}_2$ be a map between projective varieties $\mathcal{V}_1$ and $\mathcal{V}_2$. We say f is a morphism from $\mathcal{V}_1$ to $\mathcal{V}_2$ if there exists a collection of open subsets $\{U_i\}_{i\in I}$ of $\mathcal{V}_2$ such that $\mathcal{V}_2 = \bigcup_{i\in I} U_i$ (i.e., $\{U_i\}_{i\in I}$ is an open cover of $\mathcal{V}_2$) and for each $i \in I$, the restriction $f|_{f^{-1}(U_i)} : f^{-1}(U_i) \to U_i$ is a morphism between affine varieties. Furthermore, if each $f|_{f^{-1}(U_i)}$ is finite, then we say f is finite. Finiteness does not depend on the choice of the affine open cover. Namely, if $f : \mathcal{V}_1 \to \mathcal{V}_2$ is a finite morphism between projective varieties $\mathcal{V}_1$ and $\mathcal{V}_2$, and U is an open subset of $\mathcal{V}_2$ such that $f|_{f^{-1}(U)} : f^{-1}(U) \to U$ is a morphism between affine varieties, then $f|_{f^{-1}(U)}$ is also finite.

The image of a morphism $f : \mathcal{V}_1 \to \mathcal{V}_2$ is denoted by $\mathrm{Im}(f)$ or $f(\mathcal{V}_1)$. The image of a closed set under a finite morphism is still closed.

Dimension. The *dimension* of an irreducible variety $\mathcal{V}$, denoted by $\dim(\mathcal{V})$, is the largest integer m such that there exists a chain of irreducible varieties $\emptyset \subsetneq \mathcal{V}_0 \subsetneq \mathcal{V}_1 \subsetneq \cdots \subsetneq \mathcal{V}_m = \mathcal{V}$. More generally, the dimension of a nonempty variety is the maximal dimension of its irreducible components. We define the dimension of an empty set to be $-\infty$. A variety is *equidimensional* if its irreducible components have the same dimension.

If $\pi : \mathcal{V} \to \mathcal{V}'$ is a finite morphism, then $\dim(\mathcal{V}) = \dim(\pi(\mathcal{V}))$.

Degree. The *degree* of an irreducible subvariety $\mathcal{V}$ of $\mathbb{P}^n$ (resp. $\mathbb{A}^n$), denoted by $\deg(\mathcal{V})$, is the number of intersections of $\mathcal{V}$ with a projective (resp. affine) subspace of codimension $\dim(\mathcal{V})$ in general position. More generally, we define the degree of a subvariety of $\mathbb{P}^n$ or $\mathbb{A}^n$ to be the sum of the degrees of its irreducible components.

Projective closure. The affine n-space $\mathbb{A}^n$ may be regarded as an open subset of $\mathbb{P}^n$ via the map $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, 1)$. The complement $H_\infty := \mathbb{P}^n \setminus \mathbb{A}^n$ is a hyperplane of $\mathbb{P}^n$ defined by $X_{n+1} = 0$, called the *hyperplane at infinity*. For an affine subvariety $\mathcal{V}$ of $\mathbb{A}^n \subseteq \mathbb{P}^n$, the smallest projective subvariety of $\mathbb{P}^n$ containing $\mathcal{V}$ is the *projective closure* of $\mathcal{V}$, which we denote by $\mathcal{V}_{\text{cl}}$. It is known that $\mathcal{V}_{\text{cl}} \cap \mathbb{A}^n = \mathcal{V}$, $\dim(\mathcal{V}_{\text{cl}}) = \dim(\mathcal{V})$, and $\deg(\mathcal{V}_{\text{cl}}) = \deg(\mathcal{V})$.

Joins of disjoint projective varieties. For two distinct points $p, q \in \mathbb{P}^n$, denote by $\overline{pq}$ the unique projective line passing through them. For two *disjoint* projective subvarieties $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathbb{P}^n$, define the *join* $J(\mathcal{V}_1, \mathcal{V}_2)$ of $\mathcal{V}_1$ and $\mathcal{V}_2$ as

$$J(\mathcal{V}_1, \mathcal{V}_2) := \bigcup_{p \in \mathcal{V}_1, q \in \mathcal{V}_2} \overline{pq}.$$

▶ **Lemma 17** ([45, Examples 6.17, 11.36, and 18.17])**.** *$J(\mathcal{V}_1, \mathcal{V}_2)$ is a subvariety of $\mathbb{P}^n$ of dimension $\dim(\mathcal{V}_1) + \dim(\mathcal{V}_2) + 1$ and degree at most $\deg(\mathcal{V}_1) \cdot \deg(\mathcal{V}_2)$.*

We also need the following lemmas.

▶ **Lemma 18.** *Let $\mathcal{V}$ be a nonempty subvariety of $\mathbb{P}^n$ of dimension $r < n$. Let $W \subseteq \mathbb{P}^n$ be a k-subspace disjoint from $\mathcal{V}$. Let k' be an integer satisfying $k \le k' \le n - r - 1$. Then there exists a k'-subspace $W' \subseteq \mathbb{P}^n$ such that $W \subseteq W'$ and W' is disjoint from $\mathcal{V}$. In particular, choosing W to be a point not in $\mathcal{V}$ shows that there exists an $(n - r - 1)$-subspace disjoint from $\mathcal{V}$.*

Proof. We prove the lemma for the special case $k' = k + 1 \le n - r - 1$ and the general case follows from iteration. By Lemma 17, $J(\mathcal{V}, W)$ has dimension $r + k + 1 \le n - 1$. Pick a point $p \in \mathbb{P}^n \setminus J(\mathcal{V}, W)$ and let $W' = J(p, W)$. Then W' is a $(k+1)$-subspace and $W \subseteq W'$. To prove W' is disjoint from $\mathcal{V}$, assume to the contrary that there exists a point $q \in W' \cap \mathcal{V}$. By definition, $q \in \overline{pq'}$ for some $q' \in W$. As W is disjoint from $\mathcal{V}$, we have $q' \ne q$. Then $p \in \overline{pq'} = \overline{qq'} \in J(\mathcal{V}, W)$, contradicting the choice of p. ◀

▶ **Lemma 19** ([45, Exercise 11.6 and Corollary 18.5])**.** *Let $\mathcal{V}$ be a nonempty equidimensional subvariety of $\mathbb{P}^n$ and H a hypersurface of $\mathbb{P}^n$ not containing an irreducible component of $\mathcal{V}$. Then $\mathcal{V} \cap H$ is an equidimensional subvariety of dimension $\dim(\mathcal{V}) - 1$ and degree at most $\deg(\mathcal{V}) \cdot \deg(H)$ (or an empty set if $\dim(\mathcal{V}) = 0$).*

▶ **Lemma 20** ([71, Section I.6.2, Theorem 6])**.** *Suppose $\mathcal{V}_1$ and $\mathcal{V}_2$ are subvarieties of $\mathbb{P}^n$ and $\dim(\mathcal{V}_1) + \dim(\mathcal{V}_1) \ge n$. Then $\mathcal{V}_1 \cap \mathcal{V}_2 \ne \emptyset$ and $\dim(\mathcal{V}_1 \cap \mathcal{V}_2) \ge \dim(\mathcal{V}_1) + \dim(\mathcal{V}_1) - n$.*

3 Proof of the Main Theorem

We prove the Main Theorem (Theorem 6) in this section. In Subsection 3.1, we show that it suffices to consider equidimensional or irreducible subvarieties of dimension $n - k - 1$. Subsection 3.2 contains an introduction to Chow forms. Finally, in Subsection 3.3, we present the explicit constructions and complete the proof of Theorem 6.

3.1 Reducing to the Case of Equidimensional or Irreducible Varieties

The following lemma states that to construct k-subspace families that are evasive for subvarieties of $\mathbb{P}^n$, it suffices to consider equidimensional subvarieties of dimension $n - k - 1$ (i.e., codimension $k + 1$).

▶ **Lemma 21.** *Let $n, d \in \mathbb{N}^+$ and $k \in \{0, 1, \dots, n-1\}$. Let $\mathcal{F}$ be the family of all equidimensional subvarieties of $\mathbb{P}^n$ of dimension $n-k-1$ and degree at most d. Then an $(\mathcal{F}, \epsilon)$-evasive k-subspace family is also (n, d, ϵ)-evasive.*

The proof of Lemma 21 is based on the following claim.

▷ Claim 22. Let $\mathcal{V}$ be an irreducible subvariety of $\mathbb{P}^n$. There exists a subvariety $\widetilde{\mathcal{V}} \subseteq \mathbb{P}^n$ of dimension $n-k-1$ and degree at most $\deg(\mathcal{V})$ such that any k-subspace of $\mathbb{P}^n$ that evades $\widetilde{\mathcal{V}}$ also evades $\mathcal{V}$.

Proof. If $\dim(\mathcal{V}) = n-k-1$, then just let $\widetilde{\mathcal{V}} = \mathcal{V}$.

Now assume $\dim(\mathcal{V}) < n-k-1$. Let $t = (n-k-1) - \dim(\mathcal{V}) - 1$ and let $\widetilde{\mathcal{V}}$ be the join of $\mathcal{V}$ and a t-subspace disjoint from $\mathcal{V}$ (which exists by Lemma 18). Then $\widetilde{\mathcal{V}}$ is a projective subvariety of dimension $n-k-1$ and degree at most $\deg(\mathcal{V})$ by Lemma 17. Suppose W is a k-subspace that evades $\widetilde{\mathcal{V}}$. Then W is disjoint from $\widetilde{\mathcal{V}} \supseteq \mathcal{V}$. So W also evades $\mathcal{V}$.

Finally, assume $\dim(\mathcal{V}) > n-k-1$. Let $t = \dim(\mathcal{V}) - (n-k-1)$. By Lemma 19, there exist t hyperplanes $H_1, \dots, H_t$ of $\mathbb{P}^n$ such that $\mathcal{V} \cap \bigcap_{i=1}^t H_i$ is equidimensional of dimension $n-k-1$ and degree at most $\deg(\mathcal{V})$. Let $\widetilde{\mathcal{V}} = \mathcal{V} \cap \bigcap_{i=1}^t H_i$. Suppose W is a k-subspace that evades $\widetilde{\mathcal{V}}$. Then $W \cap \widetilde{\mathcal{V}} = (W \cap \mathcal{V}) \cap \bigcap_{i=1}^t H_i = \emptyset$. Again by Lemma 19, we have $\dim(W \cap \mathcal{V}) \leq t-1 = \dim(\mathcal{V}) + \dim(W) - n$. So W also evades $\mathcal{V}$. ◁

Proof of Lemma 21. Consider a projective subvariety $\mathcal{V} \subseteq \mathbb{P}^n$ of degree at most d. Let $\mathcal{V}_1, \dots, \mathcal{V}_s$ be the irreducible components of $\mathcal{V}$. For each $i \in [s]$, use Claim 22 to choose a projective subvariety $\widetilde{\mathcal{V}}_i \subseteq \mathbb{P}^n$ of dimension $n-k-1$ and degree at most $\deg(\mathcal{V}_i)$ such that any k-subspace that evades $\widetilde{\mathcal{V}}_i$ also evades $\mathcal{V}_i$. Let $\widetilde{\mathcal{V}} = \bigcup_{i=1}^s \widetilde{\mathcal{V}}_i$. Then $\widetilde{\mathcal{V}} \in \mathcal{F}$. By construction, any k-subspace that evades $\widetilde{\mathcal{V}}$ also evades $\mathcal{V}$. It follows that an $(\mathcal{F}, \epsilon)$-evasive k-subspace family is also (n, d, ϵ)-evasive. ◀

We further reduce to the case of irreducible varieties at the cost of blowing up the parameter ϵ by a factor of d. This is useful as we need irreducibility later in Lemma 32.

▶ **Lemma 23.** *Let $n, d \in \mathbb{N}^+$ and $k \in \{0, 1, \dots, n-1\}$. Let $\mathcal{F}'$ be the family of all irreducible subvarieties of $\mathbb{P}^n$ of dimension $n-k-1$ and degree at most d. Then an $(\mathcal{F}', \epsilon)$-evasive k-subspace family is also an $(n, d, d\epsilon)$-evasive k-subspace family.*

Proof. Let $\mathcal{F}$ be as in Lemma 21. Each $\mathcal{V} \in \mathcal{F}$ has at most d irreducible components, which are all in $\mathcal{F}'$ since their degrees are bounded by d. By definition and the union bound, if a k-subspace family $\mathcal{H}$ is $(\mathcal{F}', \epsilon)$-evasive, then it is also $(\mathcal{F}, d\epsilon)$-evasive. Combining this with Lemma 21 proves the lemma. ◀

3.2 Chow Forms

By Lemma 21 and Lemma 23, we only need to evade equidimensional or irreducible projective subvarieties of codimension $k+1$. The "bad" k-subspaces that intersect such a variety $\mathcal{V}$ form a hypersurface of the Grassmannian defined by a single form called the *Chow form* of $\mathcal{V}$. We now explain the basic theory of Chow forms.

Grassmannians. Let $n \in \mathbb{N}$ and $k \in \{0, 1, \dots, n-1\}$. The *Grassmannian* $\mathrm{G}(k+1, n+1)$ is the set of all $(k+1)$-dimensional linear subspaces of $\mathbb{A}^{n+1}$. By taking the quotient modulo scalars, it may also be identified with the set of all k-subspaces of $\mathbb{P}^n$, which we denote by $\mathbb{G}(k, n)$.

The Plücker embedding and Plücker coordinates. Consider a linear subspace $W \in \mathrm{G}(k+1, n+1)$. The simplest way of representing W is using a $(k+1) \times (n+1)$ matrix A over $\mathbb{F}$ such that W equals the row space of A. We call such a matrix A a *generating matrix* of W. For convenience, we also say A is a generating matrix of $\mathbb{P}W \in \mathbb{G}(k, n)$.

The entries of A are called the *(primal) Stiefel coordinates* of W. However, note that A is not uniquely determined by W since for any $(k+1) \times (k+1)$ invertible matrix M over $\mathbb{F}$, the matrix MA is also a generating matrix of W.

Another way of representing W is using the vector $(\det A_{[k+1],S})_{S \in \binom{[n+1]}{k+1}}$ of maximal minors of a generating matrix A of W. For a $(k+1) \times (k+1)$ invertible matrix M over $\mathbb{F}$, replacing A by MA corresponds to multiplying all the maximal minors $\det A_{[k+1],S}$ by $\det M \in \mathbb{F}^\times$. To remove ambiguity, we could view $(\det A_{[k+1],S})_{S \in \binom{[n+1]}{k+1}}$ as a point in the projective space $\mathbb{P}^{\binom{n+1}{k+1}-1}$, which is then uniquely determined by W. This leads to the definition of the *Plücker embedding.*

► **Definition 24** (Plücker embedding)**.** *Define* $\phi : \mathrm{G}(k+1, n+1) \to \mathbb{P}^{\binom{n+1}{k+1}-1}$ *by*

$$\phi(W) = (\det A_{[k+1],S})_{S \in \binom{[n+1]}{k+1}}$$

where A is a generating matrix of W.

The Plücker embedding embeds the Grassmannian $\mathrm{G}(k+1, n+1)$ in $\mathbb{P}^{\binom{n+1}{k+1}-1}$ as an irreducible projective subvariety, as stated by the following theorem. See, e.g., [45, 32] for proofs.

► **Theorem 25.** *The Plücker embedding ϕ is a well-defined injective map whose image is an irreducible projective subvariety of $\mathbb{P}^{\binom{n+1}{k+1}-1}$.*

The homogeneous coordinates $(\det A_{[k+1],S})_{S \in \binom{[n+1]}{k+1}}$ of $\phi(W)$ are called the *(primal) Plücker coordinates* of W.

Denote by $R := \mathbb{F}\left[X_S : S \in \binom{[n+1]}{k+1}\right]$ the homogeneous coordinate ring of $\mathbb{P}^{\binom{n+1}{k+1}-1}$. The irreducible projective subvariety $\phi(\mathrm{G}(k+1, n+1))$ is defined by a homogeneous prime ideal of R, which we denoted by I. Then R/I is the homogeneous coordinate ring of $\phi(\mathrm{G}(k+1, n+1))$. The ideal I contains precisely the polynomial relations that the Plücker coordinates need to satisfy. It is also known that I is generated by certain quadratic forms, known as the *Plücker relations.* See [45, 32] for details.

Dual Plücker coordinates. Alternatively, we could represent a linear subspace $W \in \mathrm{G}(k+1, n+1)$ by an $(n-k) \times (n+1)$ matrix B over $\mathbb{F}$ whose rows specify the linear equations defining W. We call such a matrix B a *parity check matrix* of W. For convenience, we also say B is a parity check matrix of $\mathbb{P}W \in \mathbb{G}(k, n)$.

The entries of B are called the *dual Stiefel coordinates* of W. This gives another embedding $\phi^\vee : \mathrm{G}(k+1, n+1) \to \mathbb{P}^{\binom{n+1}{n-k}-1} = \mathbb{P}^{\binom{n+1}{k+1}-1}$, defined by

$$\phi^\vee(W) = (\det B_{[n-k],S})_{S \in \binom{[n+1]}{n-k}}.$$

The homogeneous coordinates $(\det B_{[n-k],S})_{S \in \binom{[n+1]}{n-k}}$ of $\phi^\vee(W)$ are called the *dual Plücker coordinates* of W.[6] In fact, it is known that dual Plücker coordinates are equivalent to primal Plücker coordinates. Namely, if $W \in \mathrm{G}(k+1, n+1)$ has primal Plücker coordinates $(c_S)_{S \in \binom{[n+1]}{k+1}}$, then it has dual Plücker coordinates $(c'_S)_{S \in \binom{[n+1]}{n-k}}$ with $c'_S = (-1)^{\sum_{i \in S} i - \sum_{i \in [k+1]} i} \cdot c_{[n+1] \setminus S}$ (see, e.g., [50]).

[6] Some authors use "primal" and "dual" in the opposite way (e.g., [15]).

Chow forms. Recall that we denote by $\mathbb{G}(k,n)$ the set of all k-subspaces of $\mathbb{P}^n$. By identifying $\mathrm{G}(k+1,n+1)$ with $\mathbb{G}(k,n)$ via $W \mapsto \mathbb{P}W$, we regard ϕ and $\phi^\vee$ as maps from $\mathbb{G}(k,n)$ to $\mathbb{P}^{\binom{n+1}{k+1}-1}$.

We also need the notion of *associated hypersurfaces*.

▶ **Definition 26** (Associated hypersurface [34]). *For an irreducible subvariety $\mathcal{V} \subseteq \mathbb{P}^n$ of dimension $n-k-1$, define the* associated hypersurfaces *$\mathcal{Z}_\mathcal{V}$ of $\mathcal{V}$ to be the set of k-subspaces intersecting $\mathcal{V}$, i.e.,*

$$\mathcal{Z}_\mathcal{V} := \{W \in \mathbb{G}(k,n) : \mathcal{V} \cap W \neq \emptyset\}.$$

The term "associated hypersurface" is justified by the following theorem.

▶ **Theorem 27.** *Let $\mathcal{V} \subseteq \mathbb{P}^n$ be an irreducible projective subvariety of dimension $n-k-1$ and degree $d \in \mathbb{N}^+$. Then there exists a nonzero homogeneous polynomial $P_\mathcal{V} \in R = \mathbb{F}\left[X_S : S \in \binom{[n+1]}{k+1}\right]$ of degree d such that $\phi(\mathcal{Z}_\mathcal{V})$ is defined by $P_\mathcal{V}$ as a subvariety of $\phi(\mathbb{G}(k,n))$. That is,*

$$\phi(\mathcal{Z}_\mathcal{V}) = \phi(\mathbb{G}(k,n)) \cap \mathcal{V}(P_\mathcal{V}).$$

Moreover, $\mathcal{R}_\mathcal{V} := P_\mathcal{V} + I \in (R/I)_d$ is uniquely determined by $\mathcal{V}$ up to scalars.

Theorem 27 is explicitly stated as [15, Theorem 1.1 and Corollary 2.1]. A proof can be found in [34, Section 3.2]. We briefly explain how to find a polynomial $P_\mathcal{V}$ satisfying Theorem 27: Firstly, it can be shown using the trick of dimension counting via incidence varieties that $\phi(\mathcal{Z}_\mathcal{V})$ is an irreducible projective subvariety of the Grassmannian $\phi(\mathbb{G}(k,n))$ of codimension one [34, Section 3.2, Proposition 2.2]. Secondly, the homogeneous coordinate ring R/I of the Grassmannian is known to be a *unique factorization domain* [32, Chapter 9]. These two facts imply that the homogeneous ideal of R/I defining $\phi(\mathcal{Z}_\mathcal{V})$ is a *principal ideal*. Choose $\mathcal{R}_\mathcal{V}$ to be a generator of this principal ideal, which is unique up to scalars. Then lift $\mathcal{R}_\mathcal{V} \in R/I$ to $P_\mathcal{V} \in R$.

Now we are ready to define the Chow form of projective subvarieties.

▶ **Definition 28** (Chow form). *Let $\mathcal{V} \subseteq \mathbb{P}^n$ be an irreducible subvariety of dimension $n-k-1$ and degree $d \in \mathbb{N}^+$. Define the* Chow form of $\mathcal{V}$ in Plücker coordinates*, or simply the* Chow form *of $\mathcal{V}$, to be $\mathcal{R}_\mathcal{V} \in (R/I)_d$ as in Theorem 27.*

More generally, for an equidimensional subvariety $\mathcal{V} = \bigcup_{i=1}^s \mathcal{V}_i \subseteq \mathbb{P}^n$ of dimension $n-k-1$ and degree d, where $\mathcal{V}_1, \ldots, \mathcal{V}_s$ are the irreducible components of $\mathcal{V}$, the Chow form *of $\mathcal{V}$ is $\mathcal{R}_\mathcal{V} := \prod_{i=1}^s \mathcal{R}_{\mathcal{V}_i} \in (R/I)_d$. It is uniquely determined by $\mathcal{V}$ up to scalars.*

As a k-subspace intersects $\mathcal{V} = \bigcup_{i=1}^s \mathcal{V}_i$ iff it intersects some $\mathcal{V}_i$, we see from Theorem 27 that the Chow form $\mathcal{R}_\mathcal{V}$ of an equidimensional projective subvariety $\mathcal{V}$ of dimension $n-k-1$ vanishes precisely at the set of k-subspaces that intersect $\mathcal{V}$.

▶ Example 29. Let $k = 0$. Let $\mathcal{V} \subseteq \mathbb{P}^n$ be a hypersurface defined by a nonzero homogeneous polynomial $P \in \mathbb{F}[X_1, \ldots, X_{n+1}] = R$. The ideal I of R is zero in this case. And the Chow form $\mathcal{R}_\mathcal{V}$ of $\mathcal{V}$ is simply P (up to a scalar).

▶ Example 30. Let $V \in \mathrm{G}(n-k, n+1)$ and $W \in \mathrm{G}(k+1, n+1)$. Choose matrices $A, B \in \mathbb{F}^{(k+1)\times(n+1)}$ such that A is a generating matrix of W and B is a parity check matrix of V. Then $\mathbb{P}V \cap \mathbb{P}W \neq \emptyset$ iff $\dim(V \cap W) > 0$, which holds iff $\det(AB^T) = 0$. On the other hand, we have

$$\det(AB^T) = \sum_{S \in \binom{[n+1]}{k+1}} \det(A_{[k+1],S}) \cdot \det((B^T)_{S,[k+1]}) = \sum_{S \in \binom{[n+1]}{k+1}} \det(A_{[k+1],S}) \cdot \det(B_{[k+1],S}),$$

where the first equation is known as the *Cauchy–Binet formula* (see, e.g., [28]). So $P_{\mathbb{P}V} \in R_1$ is a linear polynomial whose coefficients are given by the dual Plücker coordinates $(\det B_{[k+1],S})_{S \in \binom{[n+1]}{k+1}}$ of V (up to a scalar). The degree-one part I_1 of I is zero as I is generated by quadratic forms. So the Chow form $\mathcal{R}_{\mathbb{P}V} \in (R/I)_1 = R_1$ is simply $P_{\mathbb{P}V}$.

Chow forms in Stiefel coordinates. We may also express the Chow form in Stiefel coordinates, i.e., in the entries of a generating matrix of a linear subspace. This expression has the advantage that it is an actual polynomial rather than a member of the abstract vector space $(R/I)_d$.

Formally, let A^* be a $(k+1) \times (n+1)$ variable matrix whose (i,j)-th entry is a variable $Y_{i,j}$. Define the ring homomorphism

$$\phi^\sharp : R = \mathbb{F}\left[X_S : S \in \binom{[n+1]}{k+1}\right] \to \mathbb{F}[Y_{i,j} : i \in [k+1], j \in [n+1]]$$

that sends each variable X_S to $\det(A^*_{[k+1],S})$. Define the *Chow form of $\mathcal{V}$ in Stiefel coordinates* to be

$$\widetilde{\mathcal{R}}_{\mathcal{V}} := \phi^\sharp(P_{\mathcal{V}}) \in \mathbb{F}[Y_{i,j} : i \in [k+1], j \in [n+1]]$$

where $P_{\mathcal{V}} \in R_d$ is a lift of $\mathcal{R}_{\mathcal{V}} \in (R/I)_d$. Note that I is precisely the kernel of $\phi^\sharp$. So $\widetilde{\mathcal{R}}_{\mathcal{V}}$ is uniquely determined by $\mathcal{V}$ up to scalars. By construction, for any $W \in \mathrm{G}(k+1, n+1)$ and generating matrix $A = (a_{i,j})_{i \in [k+1], j \in [n+1]}$ of W, we have $P_{\mathcal{V}}(\phi(W)) = \widetilde{\mathcal{R}}_{\mathcal{V}}(A) := \widetilde{\mathcal{R}}_{\mathcal{V}}(a_{1,1}, \dots, a_{k+1,n+1})$. So $\widetilde{\mathcal{R}}_{\mathcal{V}}$ vanishes at A iff $\mathbb{P}W \in \mathbb{G}(k,n)$ intersects $\mathcal{V}$.

Chow forms in dual Stiefel coordinates. Similarly, we may express the Chow form in dual Stiefel coordinates, i.e., in the entries of a parity check matrix of a linear subspace.

More specifically, choose a homogeneous polynomial $Q_{\mathcal{V}} \in \mathbb{F}\left[X_S : S \in \binom{[n+1]}{n-k}\right]$ that defines the set of k-subspaces intersecting $\mathcal{V}$ in terms of dual Plücker coordinates. As primal and dual Plücker coordinates are equivalent, $Q_{\mathcal{V}}$ can be obtained from the polynomial $P_{\mathcal{V}}$ above by simply negating and renaming variables. Next, compose $Q_{\mathcal{V}}$ with a ring homomorphism that substitutes dual Plücker coordinates with dual Stiefel coordinates. The resulting polynomial, which we denote by $\widetilde{\mathcal{R}}^\vee_{\mathcal{V}} \in \mathbb{F}[Y_{i,j} : i \in [n-k], j \in [n+1]]$, is called the *Chow form of $\mathcal{V}$ in dual Stiefel coordinates.*

We note that the Chow form $\widetilde{\mathcal{R}}_{\mathcal{V}}$ in primal Stiefel coordinates is a homogeneous polynomial of degree $(k+1)d$ in $(k+1)(n+1)$ variables, whereas the Chow form $\widetilde{\mathcal{R}}^\vee_{\mathcal{V}}$ in dual Stiefel coordinates is a homogeneous polynomial of degree $(n-k)d$ in $(n-k)(n+1)$ variables. This suggests that it is more convenient to use the Chow form in primal (resp. dual) Stiefel coordinates when k is small (resp. $n-k$ is small).[7]

3.3 Explicit Constructions of Variety Evasive Subspace Families

Let $n, d \in \mathbb{N}^+$, $k \in \{0, 1, \dots, n\}$, and $\epsilon \in (0,1)$. In this subsection, we prove the Main Theorem (Theorem 6) by constructing explicit projective or affine k-subspace families that are (n, d, ϵ)-evasive. The problem is trivial when $k = n$, as we just need to choose the singleton $\{\mathbb{P}^n\}$ or $\{\mathbb{A}^n\}$. So assume $k < n$.

[7] While both $\mathcal{R}_{\mathcal{V}}$ and $\mathcal{R}^\vee_{\mathcal{V}}$ may be viewed as elements of $(R/I)_d$, the two (injective) maps $\mathcal{R}_{\mathcal{V}} \mapsto \widetilde{\mathcal{R}}_{\mathcal{V}}$ and $\mathcal{R}^\vee_{\mathcal{V}} \mapsto \widetilde{\mathcal{R}}^\vee_{\mathcal{V}}$ come from different linear embedding of $(R/I)_d$ in vector spaces of polynomials. As a result, the representation of $\mathcal{V}$ by the polynomial $\widetilde{\mathcal{R}}_{\mathcal{V}}$ and the representation by $\widetilde{\mathcal{R}}^\vee_{\mathcal{V}}$ are not equally succinct in general.

We first prove Theorem 6 in the projective case, and then derive the affine case from it by viewing $\mathbb{A}^n$ as an open subset of $\mathbb{P}^n$. For the projective case, we present two constructions. The first one is simple and only uses ϵ-hitting sets for low degree polynomials (Lemma 14). But the size of the resulting subspace family is polynomial only when both d and k (or $n-k$) are bounded. Next, we give a more sophisticated construction, which yields subspace families of polynomial size as long as d is bounded.

3.3.1 Simple Construction

We first present a simple construction of (n, d, ϵ)-evasive k-subspace families on $\mathbb{P}^n$.

First assume $k+1 \leq n-k$. In this case, construct a k-subspace family $\mathcal{H}$ on $\mathbb{P}^n$ as follows:

1. Use Lemma 14 to compute an ϵ-hitting set T for the family of polynomials $f \in \mathbb{F}[Y_{i,j} : i \in [k+1], j \in [n+1]]$ of degree at most $(k+1)d$ such that $|T| = \text{poly}\left(\binom{(k+1)(n+1+d)}{(k+1)d}, 1/\epsilon\right)$. Think of T as a collection of $(k+1) \times (n+1)$ matrices over $\mathbb{F}$.
2. Initialize $\mathcal{H} = \emptyset$. For each matrix $A \in T$, if A has full row rank $k+1$, add to $\mathcal{H}$ the k-subspace $W \in \mathbb{G}(k, n)$ with the generating matrix A.

Next, assume $k+1 > n-k$. In this case, construct $\mathcal{H}$ in a similar way, but use parity check matrices instead of generating matrices. Namely, compute an ϵ-hitting set T for the family of polynomials $f \in \mathbb{F}[Y_{i,j} : i \in [n-k], j \in [n+1]]$ of degree at most $(n-k)d$ such that $|T| = \text{poly}\left(\binom{(n-k)(n+1+d)}{(n-k)d}, 1/\epsilon\right)$. Think of T as a collection of $(n-k) \times (n+1)$ matrices over $\mathbb{F}$. For each matrix $A \in T$, add to $\mathcal{H}$ the k-subspace $W \in \mathbb{G}(k, n)$ with the parity check matrix A.

This construction does give an (n, d, ϵ)-evasive k-subspace family, as stated by the following lemma.

▶ **Lemma 31.** *The k-subspace family $\mathcal{H}$ constructed above is (n, d, ϵ)-evasive and has size polynomial in $\min\left\{\binom{(k+1)(n+1+d)}{(k+1)d}, \binom{(n-k)(n+1+d)}{(n-k)d}\right\}$ and $1/\epsilon$. Moreover, the total time complexity of computing the linear equations defining the k-subspaces in $\mathcal{H}$ is polynomial in $|\mathcal{H}|$ (and $\log p$, if $\text{char}(\mathbb{F}) = p > 0$).*

Proof. We only show that $\mathcal{H}$ is (n, d, ϵ)-evasive since the rest of the lemma is obvious from the construction. Let $\mathcal{F}$ be the family of all equidimensional subvarieties of $\mathbb{P}^n$ of dimension $n-k-1$ and degree at most d. By Lemma 21, it suffices to prove that $\mathcal{H}$ is $(\mathcal{F}, \epsilon)$-evasive. Consider any $\mathcal{V} \in \mathcal{F}$. We want to show that $\mathcal{V} \cap W = \emptyset$ for all but at most ϵ-fraction of $W \in \mathcal{H}$.

First assume $k+1 \leq n-k$. The Chow form $\widetilde{\mathcal{R}}_{\mathcal{V}}$ of $\mathcal{V}$ in Stiefel coordinates is a nonzero homogeneous polynomial in $\mathbb{F}[Y_{i,j} : i \in [k+1], j \in [n+1]]$ of degree $(k+1)\deg(\mathcal{V}) \leq (k+1)d$. By the choice of T, for all but at most ϵ-fraction of $A \in T$, we have $\widetilde{\mathcal{R}}_{\mathcal{V}}(A) \neq 0$, which implies $\mathcal{V} \cap W = \emptyset$, where A is a generating matrix of W.

By construction, $\mathcal{H}$ is the collection of k-subspaces corresponding to the matrices $A \in T$ of full row rank. So we have ignored the matrices that do not have full row rank. But this does not increase the fraction of "bad" $W \in \mathcal{H}$ since if A does not have full row rank, then the maximal minors of A are all zero, and $\widetilde{\mathcal{R}}_{\mathcal{V}}(A)$ must be zero. It follows that $\mathcal{V} \cap W = \emptyset$ for all but at most ϵ-fraction of $W \in \mathcal{H}$, as desired.

Now assume $k+1 > n-k$. The proof in this case is similar and we omit the details. The only difference is that we use the Chow form $\widetilde{\mathcal{R}}_{\mathcal{V}}^{\vee}$ in dual Stiefel coordinates instead of $\widetilde{\mathcal{R}}_{\mathcal{V}}$. ◀

3.3.2 Improved Construction

For a subvariety $\mathcal{V} \subseteq \mathbb{P}^n$, denote by $\operatorname{span}(\mathcal{V})$ the smallest projective subspace that contains $\mathcal{V}$. We say $\mathcal{V}$ is *nondegenerate* if it is not contained in a hyperplane of $\mathbb{P}^n$, or equivalently, $\operatorname{span}(\mathcal{V}) = \mathbb{P}^n$.

We need the following fact from algebraic geometry (see, e.g., [24, Proposition 0] or [45, Corollary 18.12]).

▶ **Lemma 32.** *The codimension of a nondegenerate irreducible subvariety $\mathcal{V}$ of $\mathbb{P}^n$ is at most* $\deg(\mathcal{V}) - 1$.

We now give an improved construction of (n, d, ϵ)-evasive k-subspace families on $\mathbb{P}^n$ as follows.

1. If $\min\{k+1, n-k\} \le d-1$, just use the previous simple construction. So assume $\min\{k+1, n-k\} > d-1$. Let $t = k-d+2$ and $\epsilon_0 = \epsilon/(2d)$.
2. Use Lemma 14 to construct an ϵ_0-hitting set $T \subseteq \mathbb{F}^{(d-1)(n+1)}$ for the family of polynomials $f \in \mathbb{F}[Y_{i,j} : i \in [d-1], j \in [n+1]]$ of degree at most $(d-1)d$ such that $|T| = \operatorname{poly}\left(\binom{(d-1)(n+1+d)}{(d-1)d}, d/\epsilon\right)$. Think of T as a collection of $(d-1) \times (n+1)$ matrices over $\mathbb{F}$.[8]
3. Use Corollary 16 to construct a collection U of $t \times (n+1)$ matrix over $\mathbb{F}$ such that $|U| = \operatorname{poly}(n, d/\epsilon)$ and for every $(n+1) \times t$ matrix M over $\mathbb{F}$ of rank t, all but at most ϵ_0-fraction of $B \in U$ satisfies $\operatorname{rank}(BM) = t$.
4. Initialize $\mathcal{H} = \emptyset$. For each $(A, B) \in T \times U$, if the $(k+1) \times (n+1)$ matrix $\binom{A}{B}$ has full row rank, add to $\mathcal{H}$ the k-subspace $W \in \mathbb{G}(k, n)$ with the generating matrix $\binom{A}{B}$.

   See below for an illustration of a matrix $\binom{A}{B}$, where $(A, B) \in T \times U$.

$$\begin{array}{r} d-1\Big\{ \\ t\Big\{ \end{array}\left(\begin{array}{c} \overbrace{\qquad\qquad A \qquad\qquad}^{n+1} \\ \hdashline B \end{array}\right)\Bigg\}k+1$$

We use the construction above to prove the Main Theorem (Theorem 6) in the projective case. For convenience, we restate it in the following form.

▶ **Theorem 33** (Main Theorem in the projective case)**.** *The k-subspace family $\mathcal{H}$ constructed above is (n, d, ϵ)-evasive and has size* $\operatorname{poly}(N(k, d, n), n, 1/\epsilon)$. *Moreover, the total time complexity of computing the linear equations defining the k-subspaces in $\mathcal{H}$ is polynomial in $|\mathcal{H}|$ (and $\log p$, if $\operatorname{char}(\mathbb{F}) = p > 0$).*

Proof. The theorem follows from Lemma 31 if $\min\{k+1, n-k\} \le d-1$. So assume $\min\{k+1, n-k\} > d-1$ and hence $t \ge 1$. We only show that $\mathcal{H}$ is (n, d, ϵ)-evasive since the rest of the theorem is obvious from the construction.

[8] When $d = 1$, just let T be the singleton $\mathbb{F}_q^{(d-1)(n+1)} = \mathbb{F}_q^0$, which consists of an “empty matrix”.

Let $\mathcal{F}$ be the family of all irreducible subvarieties of $\mathbb{P}^n$ of dimension $n-k-1$ and degree at most d. By Lemma 23, it suffices to prove that $\mathcal{H}$ is $(\mathcal{F}, 2\epsilon_0)$-evasive. Consider any $\mathcal{V} \in \mathcal{F}$. We want to show that $\mathcal{V} \cap W = \emptyset$ for all but at most $(2\epsilon_0)$-fraction of $W \in \mathcal{H}$.

By definition, $\mathcal{V}$ is a nondegenerate irreducible subvariety of $\operatorname{span}(\mathcal{V})$. By Lemma 32, the codimension of $\mathcal{V}$ in $\operatorname{span}(\mathcal{V})$ is at most $d-1$. Therefore,

$$\dim(\operatorname{span}(\mathcal{V})) \le \dim(\mathcal{V}) + d - 1 = (n-k-1) + (d-1) = n - t.$$

Let $\Lambda \subseteq \mathbb{P}^n$ be an $(n-t)$-subspace that contains $\operatorname{span}(\mathcal{V})$. Let $M \in \mathbb{F}^{t \times (n+1)}$ be a parity check matrix of Λ. By the choice of U, all but at most ϵ_0-fraction of $B \in U$ satisfies $\operatorname{rank}(BM) = t$. Fix $B \in U$ such that $\operatorname{rank}(BM) = t$. Let $W_0 \in \mathbb{G}(t-1, n)$ such that B is a generating matrix of W_0. The condition $\operatorname{rank}(BM) = t$ is equivalent to $W_0 \cap \Lambda = \emptyset$.

We make the following claim.

▷ Claim 34. For all but ϵ_0-fraction of $A \in T$, the matrix $\binom{A}{B}$ is a generating matrix of a k-subspace $W \in \mathbb{G}(k, n)$ that is disjoint from $\mathcal{V}$.

Note that Claim 34 implies that $\mathcal{V} \cap W = \emptyset$ holds for all but at most $(2\epsilon_0)$-fraction of $W \in \mathcal{H}$. So it remains to prove this claim.

A matrix $\binom{A}{B}$ is a generating matrix of a k-subspace disjoint from $\mathcal{V}$ as long as $\widetilde{\mathcal{R}}_{\mathcal{V}}(\binom{A}{B}) \neq 0$, where $\widetilde{\mathcal{R}}_{\mathcal{V}} \in \mathbb{F}[Y_{i,j} : i \in [k+1], j \in [n+1]]$ is the Chow form of $\mathcal{V}$ in Stiefel coordinates. Consider the polynomial

$$P = \widetilde{\mathcal{R}}_{\mathcal{V}}(\tbinom{\cdot}{B}) \in \mathbb{F}[Y_{i,j} : i \in [d-1], j \in [n+1]]$$

which is obtained from $\widetilde{\mathcal{R}}_{\mathcal{V}}$ by assigning the $t \times (n+1)$ entries of B to the variables $Y_{d,1}, \ldots, Y_{k+1,n+1}$ on the bottom t rows, with the top $d-1$ rows of variables $Y_{1,1}, \ldots, Y_{d-1,n+1}$ left free.

As $W_0 \cap \Lambda = \emptyset$ and $\operatorname{span}(\mathcal{V}) \subseteq \Lambda$, we know W_0 is disjoint from $\mathcal{V}$. By Lemma 18, W_0 extends to a k-subspace that is disjoint from $\mathcal{V}$. So the generating matrix B of W_0 extends to a matrix $\binom{A}{B}$ such that $\widetilde{\mathcal{R}}_{\mathcal{V}}(\binom{A}{B}) \neq 0$. In particular, the polynomial P is not identically zero. Also note $\deg(P) = (d-1)\deg(\mathcal{V}) \le (d-1)d$. By the choice of T, for all but ϵ_0-fraction of $A \in T$, we have $\widetilde{\mathcal{R}}_{\mathcal{V}}(\binom{A}{B}) = P(A) \neq 0$, and hence $\binom{A}{B}$ is a generating matrix of a k-subspace that is disjoint from $\mathcal{V}$. This proves Claim 34 and completes the proof of the theorem. ◀

3.3.3 The Affine Case

In this subsection, we prove Theorem 6 in the affine case. Recall that we may view $\mathbb{A}^n$ as an open subset of $\mathbb{P}^n$ via the map $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, 1)$. In this way, $\mathbb{P}^n$ becomes the disjoint union of $\mathbb{A}^n$ and the *hyperplane at infinity* H_∞ defined by $X_{n+1} = 0$.

We use the following lemma to reduce the affine case to the projective case.

▶ **Lemma 35.** *Let $n, d \in \mathbb{N}^+$, $k \in \{0, 1, \ldots, n-1\}$, and $\epsilon' \in (0, 1/2)$. Suppose $\mathcal{H}$ is an (n, d, ϵ')-evasive k-subspace family on $\mathbb{P}^n$. Then*

$$\mathcal{H}' = \{W \cap \mathbb{A}^n : W \in \mathcal{H}, W \not\subseteq H_\infty\}$$

is an (n, d, ϵ)-evasive affine k-subspace family on $\mathbb{A}^n$, where $\epsilon = \epsilon'/(1-\epsilon') \le 2\epsilon'$. Moreover,

$$\mathcal{H}'' = \{W \in \mathcal{H} : W \not\subseteq H_\infty\} = \{W_{\mathrm{cl}} : W \in \mathcal{H}'\}$$

is an (n, d, ϵ)-evasive k-subspace family on $\mathbb{P}^n$.

Proof. By (n,d,ϵ')-evasiveness of $\mathcal{H}$, at most ϵ'-fraction of $W\in\mathcal{H}$ are fully contained in H_∞. Throwing away those k-subspaces fully contained in H_∞ increases the error parameter ϵ' by at most a factor of $1/(1-\epsilon')$. Therefore, $\mathcal{H}''=\{W\in\mathcal{H}:W\not\subseteq H_\infty\}$ is (n,d,ϵ)-evasive. We want to prove that $\mathcal{H}'=\{W\cap\mathbb{A}^n:W\in\mathcal{H}''\}$ is also (n,d,ϵ)-evasive.

Consider a subvariety $\mathcal{V}\subseteq\mathbb{A}^n$ of degree at most d. Let $\mathcal{V}_1,\dots,\mathcal{V}_s$ be the irreducible components of $\mathcal{V}$. The projective closure $\mathcal{V}_{\mathrm{cl}}$ of $\mathcal{V}$ has the irreducible components $(\mathcal{V}_1)_{\mathrm{cl}},\dots,(\mathcal{V}_s)_{\mathrm{cl}}$. Consider a k-subspace $W\in\mathcal{H}''$ that evades $\mathcal{V}_{\mathrm{cl}}$. We just need to prove that $W\cap\mathbb{A}^n$ evades $\mathcal{V}$. This is true since for each $i\in[s]$,

$$\begin{aligned}\dim((W\cap\mathbb{A}^n)\cap\mathcal{V}_i)&\le\dim(W\cap(\mathcal{V}_i)_{\mathrm{cl}})\le\dim(W)+\dim((\mathcal{V}_i)_{\mathrm{cl}})-n\\&=\dim(W\cap\mathbb{A}^n)+\dim(\mathcal{V}_i)-n\end{aligned}$$

where the second inequality holds since W evades $\mathcal{V}_{\mathrm{cl}}$ and the last equality uses the fact $W\not\subseteq H_\infty$. ◀

The affine case of Theorem 6 now follows easily.

Proof of Theorem 6 in the affine case. If $k=n$, just choose $\mathcal{H}=\mathbb{A}^n$. Now assume $k<n$. Construct an $(n,d,\epsilon/2)$-evasive k-subspace family $\mathcal{H}$ on $\mathbb{P}^n$ using Theorem 33. Then

$$\mathcal{H}':=\{W\cap\mathbb{A}^n:W\in\mathcal{H},W\not\subseteq H_\infty\}$$

is an (n,d,ϵ)-evasive affine k-subspace family on $\mathbb{A}^n$ by Lemma 35. The nonhomogeneous linear equations defining $W\cap\mathbb{A}^n\in\mathcal{H}'$ can be easily computed from the homogeneous linear equations defining $W\in\mathcal{H}$ by letting $X_{n+1}=1$. ◀

The proof of Theorem 6 is now complete.

Strengthening Theorem 6 in the affine case. For projective subvarieties $\mathcal{V}_1,\mathcal{V}_2\subseteq\mathbb{P}^n$ such that $\dim(\mathcal{V}_1)+\dim(\mathcal{V}_2)\ge n$, the minimum possible dimension of $\mathcal{V}_1\cap\mathcal{V}_2$ is $\dim(\mathcal{V}_1)+\dim(\mathcal{V}_2)-n$, as stated by Lemma 20. Nevertheless, for two affine subvarieties $\mathcal{V}_1,\mathcal{V}_2\subseteq\mathbb{A}^n$, it is possible that the intersection of $\mathcal{V}_1$ and $\mathcal{V}_2$ is empty even if its expected dimension $\dim(\mathcal{V}_1)+\dim(\mathcal{V}_2)-n$ is nonnegative. For example, the intersection of two distinct and parallel affine hyperplanes $\mathcal{V}_1,\mathcal{V}_2\subseteq\mathbb{A}^n$ is always empty even if $n\ge 2$. The reason this happens is that, while the dimension of $(\mathcal{V}_1)_{\mathrm{cl}}\cap(\mathcal{V}_2)_{\mathrm{cl}}$ is $n-2$ (as expected), this intersection is fully contained in the hyperplane H_∞, which is excluded from $\mathbb{A}^n$.

One may strengthen the definition of evading (Definition 1) by requiring the intersection of $\mathcal{V}_1$ with every irreducible component of $\mathcal{V}_2$ to have *exactly* the expected dimension. It is possible to construct explicit affine k-subspace families satisfying Theorem 6 even under this stronger definition of evading. We sketch the ideas as follows but omit the details.

First construct an $(n-1,d,\epsilon')$-evasive $(k-1)$-subspace family $\mathcal{H}'$ on $H_\infty\cong\mathbb{P}^{n-1}$ for some sufficiently small ϵ' depending on ϵ. Then extend each $W\in\mathcal{H}'$ to a collection of k-subspaces by picking $p\in\mathbb{A}^n$ and taking the k-subspace $J(W,p)$, where the coordinates of p are chosen from an ϵ'-hitting set for polynomials of degree at most d given by Lemma 31. Call the resulting k-subspace family $\mathcal{H}$. It is easy to prove that $\mathcal{H}$ is $(n,d,O(\epsilon'))$-evasive.

Furthermore, the affine k-subspace family $\{W\cap\mathbb{A}^n:W\in\mathcal{H}\}$ is (n,d,ϵ)-evasive even under the stronger definition of evading. To see this, consider an affine subvariety $\mathcal{V}\subseteq\mathbb{A}^n$ of degree at most d. For most $W\in\mathcal{H}$, we have:

- For each irreducible component $\mathcal{V}_i$ of $\mathcal{V}$, the dimension of $(\mathcal{V}_i)_{\mathrm{cl}}\cap W$ is as expected by $(n,d,O(\epsilon'))$-evasiveness of $\mathcal{H}$ and Lemma 20. Call this dimension d_i, which is $-\infty$ if $(\mathcal{V}_i)_{\mathrm{cl}}\cap W=\emptyset$.

- Moreover, the dimension of $((\mathcal{V}_i)_{\mathrm{cl}} \cap H_\infty) \cap (W \cap H_\infty)$ is at most $d_i - 1$ by $(n-1, d, \epsilon')$-evasiveness of $\mathcal{H}'$.
- Therefore, $\mathcal{V}_i \cap (W \cap \mathbb{A}^n)$ has the expected dimension d_i for each irreducible component $\mathcal{V}_i$ of $\mathcal{V}$.

4 Lower Bound

We prove Theorem 7 in this section. The main tool is the notion of *Chow varieties*, which parameterize projective subvarieties. More precisely, they parametrize a generalization of projective subvarieties, called *(effective) algebraic cycles* on a projective space.

Algebraic cycles. An *algebraic r-cycle* (or simply *r-cycle*) on $\mathbb{P}^n$ is a formal linear combination $\sum c_i \mathcal{V}_i$ of finitely many irreducible subvarieties $\mathcal{V}_i \subseteq \mathbb{P}^n$ of dimension r, where the coefficients c_i are integers. The *degree* of $\sum c_i \mathcal{V}_i$ is $\sum c_i \deg(\mathcal{V}_i)$. An r-cycle is *effective* if all its coefficients are nonnegative. Denote by $C(r,d,n)$ the set of all effective r-cycles of degree d on $\mathbb{P}^n$.

Chow varieties. Let $k \in \{0, 1, \ldots, n-1\}$ and $r = n-k-1$. The definition of Chow forms naturally extends to effective r-cycles. Namely, for an effective r-cycle $D = \sum_{i=1}^r c_i \mathcal{V}_i$ of degree d on $\mathbb{P}^n$, define the Chow form of D to be $\mathcal{R}_D := \prod_{i=1}^r \mathcal{R}_{\mathcal{V}_i}^{c_i}$.

Note that $\mathcal{R}_D$ is a vector in $(R/I)_d$ and is uniquely determined by D up to scalars. Write $[\mathcal{R}_D]$ for the point in $\mathbb{P}(R/I)_d$ represented by $\mathcal{R}_D$. Then we have map $\psi : C(r,d,n) \to \mathbb{P}(R/I)_d$, given by

$$\psi : D \mapsto [\mathcal{R}_D],$$

called the *Chow embedding* of $C(r,d,n)$. Indeed, it embeds $C(r,d,n)$ in $\mathbb{P}(R/I)_d$ as a projective subvariety, as stated by the following theorem of Chow and van der Waerden [13].

▶ **Theorem 36** ([13]). *The map ψ is injective and its image is Zariski-closed.*

A proof can also be found in [34, Chapter 4]. We identify $C(r,d,n)$ with its image under ψ and view it as a projective variety. This variety is called the *Chow variety* of effective r-cycles of degree d on $\mathbb{P}^n$.

▶ Example 37. Let V be the subspace of homogeneous polynomials in $\mathbb{F}[X_1, \ldots, X_{n+1}]$ of degree d. Then $C(n-1,d,n)$ is simply the projective space $\mathbb{P}V$ (see Example 29).

▶ Example 38. $C(r,1,n)$ is the Grassmannian $\mathrm{G}(r+1, n+1)$ (or $\mathbb{G}(r,n)$) embedded in $\mathbb{P}^{\binom{n+1}{r+1}-1} = \mathbb{P}^{\binom{n+1}{k+1}-1}$ via $\phi^\vee$ (see Example 30).

The dimension of Chow varieties. When $d = 1$, the Chow variety $C(r,d,n)$ is just the Grassmannian $\mathrm{G}(r+1,n+1)$ (see Example 38) and its dimension is well known to be $(r+1)(n-r)$ [45]. When $d > 1$, the dimension of $C(r,d,n)$ was determined by Azcue in his Ph.D. thesis [5] and independently by Lehmann [59]. We state their result as follows.

▶ **Theorem 39** ([5, 59]). *For $d > 1$ and $0 \le r < n$, the dimension of $C(r,d,n)$ is*

$$\max\left\{d(r+1)(n-r), \binom{d+r+1}{r+1} - 1 + (r+2)(n-r-1)\right\}.$$

This theorem was previously proved by Eisenbud and Harris [25] for the special case $r = 1$.

▶ Remark. To prove Theorem 7, we only need a lower bound for the dimension of the Chow variety, which is much easier to prove than Theorem 39. Indeed, it is not difficult to see that $d(r+1)(n-r)$ is the dimension of the space of unions of d r-subspaces of $\mathbb{P}^n$, and $\binom{d+r+1}{r+1} - 1 + (r+2)(n-r-1)$ is the dimension of the space of degree-d hypersurfaces in $(r+1)$-subspaces of $\mathbb{P}^n$.

Lower bound via dimension counting. We now restate Theorem 7 and prove it using a dimension counting argument.

▶ **Theorem 7.** *Let $n, d \in \mathbb{N}^+$ and $k \in \{0, 1, \ldots, n-1\}$. Let $\mathcal{F}$ be the family of equidimensional projective subvarieties of $\mathbb{P}^n$ of dimension $n-k-1$ and degree at most d. Suppose $\mathcal{H}$ is an $\mathcal{F}$-evasive k-subspace family on $\mathbb{P}^n$. Then*

$$|\mathcal{H}| \geq \begin{cases} (n-k)(k+1)+1 & \text{if } d = 1, \\ \max\left\{d(n-k)(k+1)+1, \binom{d+n-k}{d} + (n-k+1)k\right\} & \text{if } d > 1. \end{cases}$$

In particular, $|\mathcal{H}|$ is superpolynomial in n when $n-k = \Omega(n)$ and $d = \omega(1)$.

Proof. Consider an arbitrary k-subspace $W \in \mathcal{H}$. We may think of each point in $\mathbb{P}(R/I)_d$ as a homogeneous polynomial of degree d in Plücker coordinates modulo scalars and the ideal I of Plücker relations. We know Plücker coordinates always satisfy the Plücker relations. So it makes sense to talk about if a point in $\mathbb{P}(R/I)_d$ vanishes at $\phi(W)$ or not, as it does not depend on the choice of the homogeneous polynomial representing this point. Note that the constraint of $p \in \mathbb{P}(R/I)_d$ vanishing at $\phi(W)$ is a linear equation in the homogeneous coordinates of p. So the set of points in $\mathbb{P}(R/I)_d$ vanishing at $\phi(W)$ is a hyperplane of $\mathbb{P}(R/I)_d$, which we denote by H_W.

Let $r = n-k-1$. Assume $|\mathcal{H}| \leq \dim(C(r,d,n))$. Then we have

$$\psi(C(r,d,n)) \cap \bigcap_{W \in \mathcal{H}} H_W \neq \emptyset$$

since taking the intersection with a hyperplane reduces the dimension of a projective subvariety by at most one (Lemma 19 or Lemma 20). So there exists an effective r-cycle $D \in C(r,d,n)$ such that $\psi(D) = [\mathcal{R}_D]$ vanishes at $\phi(W)$ for all $W \in \mathcal{H}$. Suppose $D = \sum_{i=1}^{s} c_i \mathcal{V}_i$ where $c_i \in \mathbb{N}^+$ for $i \in [s]$ and $\mathcal{V}_1, \ldots, \mathcal{V}_s$ are distinct irreducible varieties.

Let $\mathcal{V} = \bigcup_{i=1}^{s} \mathcal{V}_s$. Note $\mathcal{V} \in \mathcal{F}$ since $\deg(\mathcal{V}) = \sum_{i=1}^{s} \deg(\mathcal{V}_i) \leq \sum_{i=1}^{s} c_i \deg(\mathcal{V}_i) = d$. For all $W \in \mathcal{H}$, we know $\mathcal{R}_D = \prod_{i=1}^{s} \mathcal{R}_{\mathcal{V}_i}^{c_i}$ vanishes at $\phi(W)$, or equivalently, $\mathcal{R}_{\mathcal{V}} = \prod_{i=1}^{s} \mathcal{R}_{\mathcal{V}_i}$ vanishes at $\phi(W)$. This implies $\mathcal{V} \cap W \neq \emptyset$ for all $W \in \mathcal{H}$. As $\mathcal{V} \in \mathcal{F}$, this contradicts our assumption about $\mathcal{H}$. We conclude

$$|\mathcal{H}| \geq \dim(C(r,d,n)) + 1.$$

The dimension of $C(r,d,n)$ is $(r+1)(n-r)$ when $d = 1$ and is given by Theorem 39 when $d > 1$. Plugging in $r = n-k-1$ proves the theorem. ◀

▶ Remark. It is easy to show that the lower bound in Theorem 7 is optimal by reversing its proof. Namely, we add random k-subspaces $W \in \mathbb{G}(k,n)$ to $\mathcal{H}$ one by one, such that each time the dimension of $\psi(C(r,d,n)) \cap \bigcap_{W \in \mathcal{H}} H_W$ is reduced by one with high probability. It is easy to see that at each step, a general k-subspace W does reduce the dimension by one. However, it requires more work to prove a reasonable bound for the coefficients defining such a k-subspace W. This is because we need to apply a union bound over the irreducible components of $\psi(C(r,d,n)) \cap \bigcap_{W \in \mathcal{H}} H_W$. An upper bound for the number of these irreducible components can be shown by following [54, Exercise 3.28]. We postpone the details to the full version of this paper.

5 Applications

In this section, we use the explicit constructions of variety-evasive subspace families in Section 3 to derandomize Noether's Normalization Lemma (Theorem 8 and Theorem 9) and black-box PIT for special depth-4 circuits (Theorem 11). The proof of Theorem 11 only uses the simple construction of variety-evasive subspace families (Lemma 31).

5.1 Derandomization of Noether's Normalization Lemma

Suppose W is a k-subspace of $\mathbb{P}^n$, and $\ell_1,\dots,\ell_{n-k}\in\mathbb{F}[X_1,\dots,X_{n+1}]$ are $n-k$ homogeneous linear polynomials such that $W=\mathcal{V}(\ell_1,\dots,\ell_{n-k})$. Then we have a map $\pi_{\ell_1,\dots,\ell_{n-k}}:\mathbb{P}^n\backslash W\to\mathbb{P}^{n-k-1}$ defined by

$$\pi_{\ell_1,\dots,\ell_{n-k}}:\mathbf{x}\mapsto(\ell_1(\mathbf{x}),\dots,\ell_{n-k}(\mathbf{x}))$$

which is well-defined since $\ell_1,\dots,\ell_{n-k}$ never simultaneously vanish on $\mathbb{P}^n\setminus W$. We say $\pi_{\ell_1,\dots,\ell_{n-k}}$ is a *projection* from $\mathbb{P}^n\setminus W$ to $\mathbb{P}^{n-k-1}$ and W is its *center*.

The following lemma is crucial. Its proof can be found in [71].

▶ **Lemma 40** ([71, Section I.5.3, Theorem 7]). *Suppose $\pi:\mathbb{P}^n\setminus W\to\mathbb{P}^m$ is a projection with center W and $\mathcal{V}$ is a subvariety of $\mathbb{P}^n$ disjoint from W. Then π restricts to a finite morphism from $\mathcal{V}$ to $\mathbb{P}^m$.*

We are now ready to prove Theorem 8 and Theorem 9, which we restate below for convenience.

▶ **Theorem 8.** *Let $n,d\in\mathbb{N}^+$, $r\in\{0,1,\dots,n\}$, and $\epsilon\in(0,1)$. There exists an explicit collection $\mathcal{L}$ of linear maps $\mathbb{A}^{n+1}\to\mathbb{A}^{r+1}$ of size* $\mathrm{poly}(N(k,d,n),n,1/\epsilon)$ *such that for every subvariety $\mathcal{V}\subseteq\mathbb{P}^n$ of dimension r and degree at most d, all but at most ϵ-fraction of $\pi\in\mathcal{L}$ induce a surjective finite morphism from $\mathcal{V}$ to $\mathbb{P}^r$. Moreover, $\mathcal{L}$ can be computed in time polynomial in $|\mathcal{L}|$ (and* $\log p$*, if* $\mathrm{char}(\mathbb{F})=p>0$*).*

Proof. If $r=n$, we have $\mathcal{V}=\mathbb{P}^n$. Then just use the identity map $\mathbb{A}^{n+1}\to\mathbb{A}^{n+1}$. So assume $r<n$.

Let $k=n-r-1$. Construct an (n,d,ϵ)-evasive k-subspace family $\mathcal{H}$ on $\mathbb{P}^n$ using Theorem 6. Consider $W\in\mathcal{H}$. Pick $n-k=r+1$ homogeneous linear polynomials $\ell_1,\dots,\ell_{r+1}\in\mathbb{F}[X_1,\dots,X_{n+1}]$ such that $W=\mathcal{V}(\ell_1,\dots,\ell_{r+1})$. These $r+1$ linear polynomials determine a linear map $\widetilde{\pi}_{\ell_1,\dots,\ell_{r+1}}:\mathbb{A}^{n+1}\to\mathbb{A}^{r+1}$ sending $\mathbf{x}\in\mathbb{A}^{n+1}$ to $(\ell_1(\mathbf{x}),\dots,\ell_{r+1}(\mathbf{x}))$, and the latter induces the projection $\pi_{\ell_1,\dots,\ell_{r+1}}:\mathbb{P}^n\setminus W\to\mathbb{P}^r$. Let $\mathcal{L}$ be the collection of all these linear maps $\widetilde{\pi}_{\ell_1,\dots,\ell_{r+1}}$, one from each $W\in\mathcal{H}$.

Let $\mathcal{V}$ be a subvariety of $\mathbb{P}^n$ of dimension r and degree at most d. We know all but at most ϵ-fraction of $W\in\mathcal{H}$ are disjoint from $\mathcal{V}$. So we just need to prove that for every $W\in\mathcal{H}$ disjoint from $\mathcal{V}$, the corresponding projection $\pi:=\pi_{\ell_1,\dots,\ell_{r+1}}:\mathbb{P}^n\setminus W\to\mathbb{P}^r$ restricts to a surjective finite morphism from $\mathcal{V}$ to $\mathbb{P}^r$. The restriction $\pi|_{\mathcal{V}}:\mathcal{V}\to\mathbb{P}^r$ is indeed finite by Lemma 40. So its image $\pi(\mathcal{V})$ is closed and has dimension $\dim(\mathcal{V})=r$. The only r-dimensional closed subset of $\mathbb{P}^r$ is $\mathbb{P}^r$ itself. So π is surjective. ◀

▶ **Theorem 9.** *Let $n,d\in\mathbb{N}^+$ and $r\in\{0,1,\dots,n\}$, and $\epsilon\in(0,1)$. There exists an explicit collection $\mathcal{L}$ of linear maps $\mathbb{A}^n\to\mathbb{A}^r$ of size* $\mathrm{poly}(N(k,d,n),n,1/\epsilon)$ *such that for every subvariety $\mathcal{V}\subseteq\mathbb{A}^n$ of dimension r and degree at most d, all but at most ϵ-fraction of $\pi\in\mathcal{L}$ restrict to a surjective finite morphism from $\mathcal{V}$ to $\mathbb{A}^r$. Moreover, $\mathcal{L}$ can be computed in time polynomial in $|\mathcal{L}|$ (and* $\log p$*, if* $\mathrm{char}(\mathbb{F})=p>0$*).*

Proof. If $r = n$, we have $\mathcal{V} = \mathbb{A}^n$. Then just use the identity map $\mathbb{A}^n \to \mathbb{A}^n$. If $r = 0$, use the only map $\mathbb{A}^n \to \mathbb{A}^0$. So assume $0 < r < n$. Regard $\mathbb{A}^n$ as an open subset of $\mathbb{P}^n$ via $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, 1)$. Similarly, regard $\mathbb{A}^r$ as an open subset of $\mathbb{P}^r$ via $(x_1, \ldots, x_r) \mapsto (x_1, \ldots, x_r, 1)$.

Let $k = n-r-1$. Construct an $(n-1, d, \epsilon)$-evasive k-subspace family $\mathcal{H}$ on $H_\infty = \mathbb{P}^n \backslash \mathbb{A}^n \cong \mathbb{P}^{n-1}$ using Theorem 6. Consider $W \in \mathcal{H}$. Pick $n-k = r+1$ homogeneous linear polynomials $\ell_1, \ldots, \ell_{r+1} \in \mathbb{F}[X_1, \ldots, X_{n+1}]$ such that $\ell_{r+1} = X_{n+1}$, $\ell_1, \ldots, \ell_r \in \mathbb{F}[X_1, \ldots, X_n]$, and $W = \mathcal{V}(\ell_1, \ldots, \ell_{r+1})$. This is possible as $W \subseteq H_\infty = \mathcal{V}(X_{n+1})$. These $r+1$ linear polynomials determine the projection $\pi_{\ell_1,\ldots,\ell_{r+1}} : \mathbb{P}^n \setminus W \to \mathbb{P}^r$, defined by

$$\mathbf{x} = (x_1, \ldots, x_{n+1}) \mapsto (\ell_1(\mathbf{x}), \ldots, \ell_{r+1}(\mathbf{x})) = (\ell_1(\mathbf{x}), \ldots, \ell_r(\mathbf{x}), x_{n+1}).$$

As $x_{n+1} = 1$ for $\mathbf{x} \in \mathbb{A}^n$, we have $\pi_{\ell_1,\ldots,\ell_{r+1}}(\mathbb{A}^n) \subseteq \mathbb{A}^r$. Restricting $\pi_{\ell_1,\ldots,\ell_{r+1}}$ on $\mathbb{A}^n$ yields a map $\pi_{\ell_1,\ldots,\ell_{r+1}}|_{\mathbb{A}^n} : \mathbb{A}^n \to \mathbb{A}^r$, which is a linear map as $\ell_1, \ldots, \ell_r$ are homogeneous linear polynomials in $\mathbb{F}[X_1, \ldots, X_n]$. Let $\mathcal{L}$ be the collection of all these linear maps $\pi_{\ell_1,\ldots,\ell_{r+1}}|_{\mathbb{A}^n}$, one from each $W \in \mathcal{H}$.

Let $\mathcal{V}$ be a subvariety of $\mathbb{A}^n$ of dimension r and degree at most d. Its projective closure $\mathcal{V}_{\mathrm{cl}}$ has dimension $\dim(\mathcal{V}) = r$ and degree $\deg(\mathcal{V}) \leq d$. By the definition of $\mathcal{V}_{\mathrm{cl}}$, none of the irreducible components of $\mathcal{V}_{\mathrm{cl}}$ is fully contained in H_∞. So by Lemma 19, the projective subvariety $\mathcal{V}_{\mathrm{cl}} \cap H_\infty$ has dimension $r-1$ and degree at most d.

By the choice of $\mathcal{H}$, all but at most ϵ-fraction of $W \in \mathcal{H}$ are disjoint from $\mathcal{V}_{\mathrm{cl}} \cap H_\infty$ and hence from $\mathcal{V}_{\mathrm{cl}}$. So we just need to prove that for every $W \in \mathcal{H}$ disjoint from $\mathcal{V}_{\mathrm{cl}}$ and the corresponding projection $\pi := \pi_{\ell_1,\ldots,\ell_{r+1}}$, the map $\pi|_{\mathcal{V}} : \mathcal{V} \to \mathbb{A}^r$ is a surjective finite morphism. We have already seen from the proof of Theorem 8 that, as the center W of π is disjoint from $\mathcal{V}_{\mathrm{cl}}$, the projection π restricts to a surjective finite morphism $\pi|_{\mathcal{V}_{\mathrm{cl}}} : \mathcal{V}_{\mathrm{cl}} \to \mathbb{P}^r$. As $\mathcal{V} = \mathcal{V}_{\mathrm{cl}} \cap \mathbb{A}^n = \mathcal{V}_{\mathrm{cl}} \cap \pi^{-1}(\mathbb{A}^r)$, the map $\pi|_{\mathcal{V}}$ is precisely the restriction of $\pi|_{\mathcal{V}_{\mathrm{cl}}}$ to $(\pi|_{\mathcal{V}_{\mathrm{cl}}})^{-1}(\mathbb{A}^r)$. As $\pi|_{\mathcal{V}_{\mathrm{cl}}}$ is a surjective finite morphism, so is $\pi|_{\mathcal{V}}$. ◀

▶ Remark. For simplicity, we have restricted to the category of varieties over an algebraically closed field $\mathbb{F}$ when stating Theorem 8 and Theorem 9. We now mention some generalizations without proofs, which lead to the usual algebraic formulation of Noether's normalization lemma and its derandomization:

- As mentioned in the remark after Theorem 6, the coefficients of the linear maps that we use live in a non-algebraically closed field $\mathbb{K}_0 \subseteq \mathbb{F}$, which is either $\mathbb{Q}$ or a finite extension of $\mathbb{F}_p$. For any field $\mathbb{K} \supseteq \mathbb{K}_0$, we have actually constructed explicit families of linear maps that are defined over $\mathbb{K}$. Theorem 8 and Theorem 9 then hold for projective/affine varieties over $\mathbb{K}$ (which we have not defined) as well.
- Furthermore, Theorem 8 and Theorem 9 hold for *closed subschemes* of projective/affine spaces over $\mathbb{K}$ as well. In fact, it suffices to consider the variety $\mathcal{V}_{\mathrm{red}} := \mathcal{V}(\sqrt{I(\mathcal{V})})$ in place of a closed subscheme $\mathcal{V}$ when checking if a linear map gives a valid surjective finite morphism. This is because the evading property that we need is set-theoretic.
- A generalization of Theorem 9 then translates into the following derandomization of Noether's normalization lemma: Let $\mathbb{K}$ be a field containing all the coefficients of the linear maps in $\mathcal{L}$, where $\mathcal{L}$ is as constructed in Theorem 9. Let $A \neq 0$ be a finitely generated commutative $\mathbb{K}$-algebra with generators $b_1, \ldots, b_n$ such that the *Krull dimension* of A is r and the variety $\mathcal{V} \subseteq \mathbb{A}^n_{\mathbb{K}}$ has degree at most d, where $\mathcal{V} = \mathcal{V}(\sqrt{I})$ and $I \subseteq \mathbb{K}[X_1, \ldots, X_n]$ is the set of polynomial relations that $b_1, \ldots, b_n$ satisfy. For a linear map $\pi \in \mathcal{L}$ defined by $(x_1, \ldots, x_n) \mapsto (\sum_{i=1}^n c_{i,1} x_i, \ldots, \sum_{i=1}^n c_{i,r} x_i)$, let $y_j^\pi = \sum_{i=1}^n c_{i,j} b_i$ for $j \in [r]$. Then

for all but at most ϵ-fraction of $\pi \in \mathcal{L}$, the corresponding $y_1^\pi, \dots, y_r^\pi$ are algebraically independent and A is a finitely-generated module over $\mathbb{K}[y_1^\pi, \dots, y_r^\pi]$. The existence of such $y_1^\pi, \dots, y_r^\pi$ is the content of the usual algebraic formulation of Noether's normalization lemma [4, Chapter 5, Exercise 16].

5.2 Black-Box PIT for Non-SG Depth-4 Circuits

We first define $\Sigma\Pi\Sigma\Pi(k,r)$ circuits and non-SG $\Sigma\Pi\Sigma\Pi(k,r)$ circuits.

▶ **Definition 41** ($\Sigma\Pi\Sigma\Pi(k,r)$ circuit)**.** *An algebraic circuit C over $\mathbb{F}$ is a $\Sigma\Pi\Sigma\Pi(k,r)$* circuit *if it has the form*

$$C(X_1,\dots,X_n) = \sum_{i=1}^{k'} F_i = \sum_{i=1}^{k'} \prod_{j=1}^{d_i} Q_{i,j} \tag{1}$$

where $k' \le k$, $d_1,\dots,d_{k'} \in \mathbb{N}^+$, $F_i = \prod_{j=1}^{d_i} Q_{i,j}$ for $j \in [k']$, and each $Q_{i,j}$ is a polynomial in $X_1,\dots,X_n$ of degree at most r over $\mathbb{F}$. The degree *of the circuit C is defined to be* $\max\{\deg(F_i) : i \in [k']\}$*. In addition:*

- *C is* minimal *if $\sum_{i\in I} F_i \neq 0$ for all nonempty proper subset $I \subsetneq [k']$.*
- *C is* homogeneous *if all the polynomials F_i are homogeneous of the same degree.*
- *Let* $\gcd(C) := \gcd(F_1,\dots,F_{k'})$*. We say C is* simple *if* $\gcd(C) = 1$*. In general, we have* $C = \gcd(C)\cdot \mathrm{sim}(C)$ *where* $\mathrm{sim}(C)$ *is a simple $\Sigma\Pi\Sigma\Pi(k,r)$ circuit, called the* simple part *of C. Note the simple part of a minimal $\Sigma\Pi\Sigma\Pi(k,r)$ circuit is still minimal.*

The polynomial computed by C is again denoted by C by an abuse of notation.

▶ **Definition 42** (Non-SG circuit)**.** *We say a minimal, simple, and homogeneous $\Sigma\Pi\Sigma\Pi(k,r)$ circuit $C(X_1,\dots,X_n) = \sum_{i=1}^{k'} F_i$ as in* (1) *is* non-SG *if there exists $i \in [k']$ such that*

$$\bigcap_{j\in[k']\setminus i} \mathcal{V}(F_j) \not\subseteq \mathcal{V}(F_i)$$

where $\mathcal{V}(F)$ denotes the subvariety of $\mathbb{P}^n$ defined by F. More generally, a minimal and simple $\Sigma\Pi\Sigma\Pi(k,r)$ circuit $C(X_1,\dots,X_n) = \sum_{i=1}^{k'} F_i$ of degree d is non-SG if its homogenization

$$\widetilde{C}(X_1,\dots,X_{n+1}) = \sum_{i=1}^{k'} F_i(X_1/X_{n+1},\dots,X_n/X_{n+1}) \cdot X_{n+1}^d = \sum_{i=1}^{k'} \prod_{j=1}^{d_i'} \widetilde{Q}_{i,j}$$

is non-SG, where each $\widetilde{Q}_{i,j}$ is either the homogenization of $Q_{i,j}$ or X_{n+1}. A minimal $\Sigma\Pi\Sigma\Pi(k,r)$ circuit C is non-SG if $\mathrm{sim}(C)$ *is non-SG. Finally, a $\Sigma\Pi\Sigma\Pi(k,r)$ circuit is non-SG if it has an equivalent minimal non-SG $\Sigma\Pi\Sigma\Pi(k,r)$ circuit.*

We restate our result (Theorem 11) and then give a proof.

▶ **Theorem 11.** *There exists a deterministic black-box PIT algorithm with time complexity polynomial in $d \cdot \binom{k(n+1+r^k)}{kr^k} \cdot \binom{k-1+d}{k-1} \le \mathrm{poly}(d^k, n^{r^k}, r^{k^2 r^k})$ (and $\log p$, if $\mathrm{char}(\mathbb{F}) = p > 0$) for non-SG $\Sigma\Pi\Sigma\Pi(k,r)$ circuits of degree at most d in $X_1,\dots,X_n$ over an algebraically closed field $\mathbb{F}$.*

Proof. If $n \le k-1$, we may simply use Lemma 14 to construct a $\frac{1}{2}$-hitting set of size polynomial in $\binom{n+d}{n} \le \binom{k-1+d}{k-1}$ for n-variate polynomials of degree at most d, and then run the corresponding black-box PIT algorithm. So assume $n > k-1$.

Consider a nonzero non-SG $\Sigma\Pi\Sigma\Pi(k,r)$ circuit C of degree at most d. We want to design a black-box PIT algorithm for C. By replacing C with an equivalent minimal non-SG circuit, we may assume C is minimal. Let $D = \gcd(C)$ and $E = \mathrm{sim}(C)$. Let $\widetilde{C}$, $\widetilde{D}$, and $\widetilde{E}$ be the homogenization of C, D, and E respectively. Then $\widetilde{D} = \gcd(\widetilde{C})$, $\widetilde{E} = \mathrm{sim}(\widetilde{C})$, and $\widetilde{C} = \widetilde{D} \cdot \widetilde{E}$.

Let $\mathcal{H}$ be an affine $(k-1)$-subspace family on $\mathbb{A}^n$ of size $\mathrm{poly}(\binom{k(n+1+r^k)}{kr^k}, d)$ such that $\mathcal{H}' := \{W_{\mathrm{cl}} : W \in \mathcal{H}\}$ is an $(n, r^k, \frac{1}{4d})$-evasive $(k-1)$-subspace family on $\mathbb{P}^n$. Such a family $\mathcal{H}$ can be computed using Lemma 35 and Lemma 31. We claim

1. $\widetilde{D}|_W \neq 0$ for all but at most $\frac{1}{4}$-fraction of $W \in \mathcal{H}'$, and
2. $\widetilde{E}|_W \neq 0$ for all but at most $\frac{1}{4}$-fraction of $W \in \mathcal{H}'$.

Assume these two claims hold. Then for at least half of $W \in \mathcal{H}$, we have $\widetilde{C}|_{W_{\mathrm{cl}}} \neq 0$ and hence $C|_W = \widetilde{C}|_{W_{\mathrm{cl}} \cap \mathbb{A}^n} \neq 0$, where we use the facts that $\widetilde{C}(X_1, \ldots, X_n, 1)$ equals $C(X_1, \ldots, X_n)$ and $W_{\mathrm{cl}} \cap \mathbb{A}^n$ is dense in W_{cl}. The restriction of C to each $W \cong \mathbb{A}^{k-1}$ is a $(k-1)$-variate polynomial of degree at most d. So to test if $C|_W$ is zero, we just need to use Lemma 14 to construct a hitting set in W of size $\mathrm{poly}(\binom{k-1+d}{k-1})$ for $(k-1)$-variate polynomials of degree at most d. Take the union of these hitting sets to obtain a hitting set of size $\mathrm{poly}(\binom{k(n+1+r^k)}{kr^k}, d, \binom{k-1+d}{k-1})$ and we are done.

So it remains to prove the two claims. Note $\widetilde{D}$ is the product of at most d factors whose degrees are bounded by r. The first claim then follows from the $(n, r^k, \frac{1}{4d})$-evasiveness of $\mathcal{H}'$ and the union bound.

Now we prove the second claim. By definition, $\widetilde{E}$ is a non-SG $\Sigma\Pi\Sigma\Pi(k,r)$ circuit. Suppose it has the form

$$\widetilde{E} = \sum_{i=1}^{k'} F_i = \sum_{i=1}^{k'} \prod_{j=1}^{d_i} Q_{i,j} \tag{2}$$

where each $Q_{i,j}$ is a homogeneous polynomial of degree at most r. As $\widetilde{E}$ is non-SG, there exists $i_0 \in [k']$ such that

$$\bigcap_{i \in [k'] \setminus i_0} \mathcal{V}(F_i) \not\subseteq \mathcal{V}(F_{i_0})$$

Without loss of generality, we may assume $i_0 = k'$. Note $\mathcal{V}(F_i) = \bigcup_{j=1}^{d_i} \mathcal{V}(Q_{i,j})$ for $i \in [k']$. So there exists $(j_1, \ldots, j_{k'-1}) \in [d_1] \times \cdots \times [d_{k'-1}]$ such that

$$\bigcap_{i=1}^{k'-1} \mathcal{V}(Q_{i,j_i}) \not\subseteq \mathcal{V}(F_{k'}).$$

Let $\mathcal{V}_0$ be an irreducible component of $\bigcap_{i=1}^{k'-1} \mathcal{V}(Q_{i,j_i})$ such that $\mathcal{V}_0 \not\subseteq \mathcal{V}(F_{k'})$. Let $d_0 = \dim(\mathcal{V}_0) \geq 0$. By Lemma 19, we have $d_0 \geq n - k' + 1$ and the variety $\mathcal{V}_0 \cap \mathcal{V}(F_{k'}) = \bigcup_{j=1}^{d_{k'}} (\mathcal{V}_0 \cap \mathcal{V}(Q_{k',j}))$ has dimension at most $d_0 - 1$. For each $j \in [d_{k'}]$, the degree of $\mathcal{V}_0 \cap \mathcal{V}(Q_{k',j})$ is at most r^k by Lemma 19 (or by Bézout's inequality [46]). By $(n, r^k, \frac{1}{4d})$-evasiveness of $\mathcal{H}'$ and the union bound, all but at most $\frac{1}{4}$-fraction of $W \in \mathcal{H}'$ evade $\mathcal{V}_0 \cap \mathcal{V}(Q_{k',j})$ for $j = 1, 2, \ldots, d_{k'}$.

Consider any $W \in \mathcal{H}'$ that evades $\mathcal{V}_0 \cap \mathcal{V}(Q_{k',j})$ for $j = 1, 2, \ldots, d_{k'}$. We just need to prove $\widetilde{E}|_W \neq 0$, or equivalently, $W \not\subseteq \mathcal{V}(\widetilde{E})$. Assume to the contrary that $W \subseteq \mathcal{V}(\widetilde{E})$. Then $W \cap \mathcal{V}_0 \subseteq \mathcal{V}(\widetilde{E})$. So

$$W \cap \mathcal{V}_0 = W \cap \mathcal{V}_0 \cap \mathcal{V}(\widetilde{E}) = W \cap \mathcal{V}_0 \cap \mathcal{V}\left(\prod_{j=1}^{d_{k'}} Q_{k',j}\right) = \bigcup_{j=1}^{d_{k'}} (W \cap \mathcal{V}_0 \cap \mathcal{V}(Q_{k',j})) \tag{3}$$

where the second equality holds since $\widetilde{E} \equiv \prod_{j=1}^{d_{k'}} Q_{k',j}$ modulo the ideal

$$I_0 := \langle Q_{1,j_1}, \dots, Q_{k'-1,j_{k'-1}} \rangle$$

by (2) and $\mathcal{V}_0 \subseteq \bigcap_{i=1}^{k'-1} \mathcal{V}(Q_{i,j_i}) = \mathcal{V}(I_0)$. We know the dimension of $\bigcup_{j=1}^{d_{k'}} (\mathcal{V}_0 \cap \mathcal{V}(Q_{k',j}))$ is at most $d_0 - 1$. So by the choice of W, the dimension of $\bigcup_{j=1}^{d_{k'}} (W \cap \mathcal{V}_0 \cap \mathcal{V}(Q_{k',j}))$ is at most $(k-1) + (d_0 - 1) - n$. However, by Lemma 20, the dimension of $W \cap \mathcal{V}_0$ is at least $(k-1) + d_0 - n \geq 0$, where we use the fact $d_0 \geq n - k' + 1 \geq n - k + 1$. This contradicts (3). So $\widetilde{E}|_W \neq 0$. ◀

6 Open Problems and Future Directions

We have seen that constructing explicit variety evasive subspace families is a natural problem that generalizes important problems in algebraic pseudorandomness and algebraic complexity theory, including deterministic black-box polynomial identity testing (evading varieties of codimension one) and constructing explicit lossless rank condensers (evading varieties of degree one). It is closely connected with advanced topics in algebraic geometry such as Chow forms and Chow varieties, and has applications to derandomizing PIT and non-explicit results in algebraic geometry like Noether's normalization lemma.

There are many interesting open problems and potential future directions. We list some of them here.

1. Theorem 6 focuses on subvarieties of bounded degree in a projective or affine space. Are there other interesting families of varieties for which we could construct explicit variety evasive subspace families? Families that are defined computation-theoretically may be particularly interesting, as many results of this kind are already known for polynomial identity testing.
2. Can explicit variety evasive subspace families be used to derandomize other non-explicit results in algebraic geometry?
3. Can our explicit construction in Theorem 6 be improved? In the case $k = 0$ and the case $d = 1$, there are optimal or essentially optimal constructions, and our construction indeed degenerates into these constructions. In general, however, there is a significant gap between the upper bound in Theorem 6 and the lower bound in Theorem 7.
4. Extending the notion of *strong* lossless rank condensers [27], one could strengthen the definition of $(\mathcal{F}, \epsilon)$-evasive subspace families in Definition 3 by bounding the total deviation of the dimension instead of the number of bad subspaces. At the same time, one could consider the setting where there is gap between $\dim(\mathcal{V}_1)$ and $\operatorname{codim}(\mathcal{V}_2)$, as in typical applications of *subspace designs* [43, 37, 41]. Alternatively, one could relax the definition by allowing $\dim(\mathcal{V}_1 \cap \mathcal{V}_2)$ to be slightly greater than $\dim(\mathcal{V}_1) + \dim(\mathcal{V}_2) - n$, which is related to the notion of *lossy* rank condensers in [27]. It is natural to study explicit constructions of these variants and their applications, which can be seen as extensions of the theory of "linear-algebraic pseudorandomness" [27] to a nonlinear setting.
5. Could our lower bound (Theorem 7) be extended to the affine case or to a "lossy" relaxation of the problem?

6. When $n-k=O(1)$, our lower bound (Theorem 7) is only polynomial in n and d. So one question is if there are explicit constructions of polynomial size when $n-k=O(1)$. As a concrete special case, consider the problem of constructing an explicit affine $(n-2)$-subspace family $\mathcal{H}$ on $\mathbb{A}^n$ such that $\mathcal{H}$ is evasive for degree-d curves that are images of morphisms $\mathbb{A}^1 \to \mathbb{A}^n$. Note that for $\varphi : \mathbb{A}^1 \to \mathbb{A}^n$ corresponding to a ring homomorphism $\varphi^\sharp : \mathbb{F}[X_1, \ldots, X_n] \to \mathbb{F}[Y]$, an affine $(n-2)$-subspace defined by affine linear polynomials ℓ_1 and ℓ_2 evades the curve $\mathrm{Im}(\varphi)$ iff $\varphi^\sharp(\ell_1)$ and $\varphi^\sharp(\ell_2)$ have no common root. Using *resultants*, we could reduce this problem to black-box PIT for symbolic determinants. We are not aware of any *unconditional* derandomization whose time complexity is subexponential in $\min\{n, d\}$, however.

References

1 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697, 2015. doi:10.1137/140975103.

2 Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM Journal on Computing*, 45(4):1533–1562, 2016.

3 Manindra Agrawal and V Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 67–75, 2008.

4 Michael F. Atiyah and I. G. MacDonald. *Introduction to Commutative Algebra*. Addison-Wesley-Longman, 1969.

5 Pablo Azcue. *On the dimension of the Chow varieties*. PhD thesis, Harvard University, 1992.

6 Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Information and Computation*, 222:2–19, 2013.

7 Markus Bläser and Anurag Pandey. Polynomial identity testing for low degree polynomials with optimal randomness. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 8:1–8:13, 2020.

8 Andrej Bogdanov. Pseudorandom generators for low degree polynomials. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 21–30, 2005.

9 Juliette Bruce and Daniel Erman. A probabilistic approach to systems of parameters and Noether normalization. *Algebra & Number Theory*, 13(9):2081–2102, 2019.

10 Nader Bshouty. Testers and their applications. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, pages 327–352, 2014.

11 Arthur Cayley. On a new analytical representation of curves in space. *The Quarterly Journal of Pure and Applied Mathematics*, 3:225–236, 1860.

12 Arkadev Chattopadhyay, Ankit Garg, and Suhail Sherif. Towards stronger counterexamples to the log-approximate-rank conjecture. *arXiv preprint*, 2020. arXiv:2009.02717.

13 Wei-Liang Chow and B.L. van der Waerden. Zur algebraischen Geometrie. IX. *Mathematische Annalen*, 113(1):692–704, 1937.

14 Gil Cohen and Amnon Ta-Shma. Pseudorandom generators for low degree polynomials from algebraic geometry codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 20, page 155, 2013.

15 John Dalbec and Bernd Sturmfels. Introduction to Chow forms. *Invariant Methods in Discrete and Computational Geometry*, pages 37–58, 1995.

16 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978.

17 Thomas W Dubé. A combinatorial proof of the effective Nullstellensatz. *Journal of Symbolic Computation*, 15(3):277–296, 1993.

18 Zeev Dvir. Extractors for varieties. *Computational Complexity*, 21(4):515–572, 2012.

19 Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18(1):1–58, 2009.

20 Zeev Dvir, János Kollár, and Shachar Lovett. Variety evasive sets. *Computational Complexity*, 23(4):509–529, 2014.

21 Zeev Dvir and Shachar Lovett. Subspace evasive sets. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 351–358, 2012.

22 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007.

23 David Eisenbud. *Commutative Algebra: with a View Toward Algebraic Geometry*, volume 150. Springer Science & Business Media, 2013.

24 David Eisenbud and Joe Harris. On varieties of minimal degree. In *Proceedings of Symposia in Pure Mathematics*, volume 46, pages 3–13, 1987.

25 David Eisenbud and Joe Harris. The dimension of the Chow variety of curves. *Compositio Mathematica*, 83(3):291–310, 1992.

26 Michael A. Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs.* PhD thesis, Massachusetts Institute of Technology, 2014.

27 Michael A. Forbes and Venkatesan Guruswami. Dimension expanders via rank condensers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2015.

28 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 867–875, 2014.

29 Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 163–172, 2012.

30 Michael A. Forbes and Amir Shpilka. Explicit Noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 527–542, 2013.

31 Michael A. Forbes and Amir Shpilka. A PSPACE construction of a hitting set for the closure of small algebraic circuits. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*, pages 1180–1192, 2018.

32 William Fulton. *Young Tableaux: With Applications to Representation Theory and Geometry*, volume 35. Cambridge University Press, 1997.

33 Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008.

34 Israel M. Gelfand, Mikhail M. Kapranov, and Andrei V. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants.* Birkhäuser, 1994.

35 Marc Giusti, Klemens Hägele, Grégoire Lecerf, Joël Marchand, and Bruno Salvy. The projective Noether Maple package: computing the dimension of a projective variety. *Journal of Symbolic Computation*, 30(3):291–307, 2000.

36 Marc Giusti and Joos Heintz. La d etermination des points isol es et de la dimension d'une vari et e alg ebrique peut se faire en temps polynomial. *Computational Algebraic Geometry and Commutative Algebra (Cortona, 1991)*, pages 216–256, 1993.

37 Zeyu Guo and Noga Ron-Zewi. Efficient list-decoding with constant alphabet and list sizes. *arXiv preprint*, 2020. To appear in STOC 2021. `arXiv:2011.05884`.

38 Zeyu Guo, Nitin Saxena, and Amit Sinhababu. Algebraic dependencies and PSPACE algorithms in approximative complexity over any field. *Theory of Computing*, 15(16):1–30, 2019.

39 Ankit Gupta. Algebraic geometric techniques for depth-4 PIT & Sylvester-Gallai conjectures for varieties. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 21, page 130, 2014.

40 Venkatesan Guruswami and Swastik Kopparty. Explicit subspace designs. *Combinatorica*, 36(2):161–185, 2016.

41 Venkatesan Guruswami, Nicolas Resch, and Chaoping Xing. Lossless dimension expanders via linearized polynomials and subspace designs. *Combinatorica*, pages 1–35, 2021.

42 Venkatesan Guruswami, Carol Wang, and Chaoping Xing. Explicit list-decodable rank-metric and subspace codes via subspace designs. *IEEE Transactions on Information Theory*, 62(5):2707–2718, 2016.

43 Venkatesan Guruswami and Chaoping Xing. List decoding Reed-Solomon, Algebraic-Geometric, and Gabidulin subcodes up to the Singleton bound. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 843–852, 2013.

44 Venkatesan Guruswami, Chaoping Xing, and Chen Yuan. Subspace designs based on algebraic function fields. *Transactions of the American Mathematical Society*, 370(12):8757–8775, 2018.

45 Joe Harris. *Algebraic Geometry: A First Course*, volume 133. Springer Science & Business Media, 2013.

46 Joos Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 24(3):239–277, 1983.

47 Joos Heintz and Malte Sieveking. Absolute primality of polynomials is decidable in random polynomial time in the number of variables. In *International Colloquium on Automata, Languages, and Programming*, pages 16–28, 1981.

48 David Hilbert. Ueber die Theorie der algebraischen Formen. *Mathematische Annalen*, 36(4):473–534, 1890.

49 Gabriela Jeronimo, Teresa Krick, Juan Sabia, and Martín Sombra. The computational complexity of the Chow form. *Foundations of Computational Mathematics*, 4(1):41–117, 2004.

50 Michael Joswig and Thorsten Theobald. *Polyhedral and Algebraic Methods in Computational Geometry*. Springer Science & Business Media, 2013.

51 Zohar S. Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333, 2011.

52 Adam R. Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 216–223, 2001.

53 János Kollár. Sharp effective Nullstellensatz. *Journal of the American Mathematical Society*, pages 963–975, 1988.

54 János Kollár. *Rational Curves on Algebraic Varieties*, volume 32. Springer Science & Business Media, 2013.

55 Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters. Improved decoding of folded Reed-Solomon and multiplicity codes. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science*, pages 212–223, 2018.

56 Teresa Krick. Straight-line programs in polynomial equation solving. *Foundations of Computational Mathematics*, 312:96–136, 2002.

57 Joseph M. Landsberg. Tensors: geometry and applications. *Representation theory*, 381(402):3, 2012.

58 Joseph M. Landsberg. Geometric complexity theory: an introduction for geometers. *Annali dell'universita'di Ferrara*, 61(1):65–117, 2015.

59 Brian Lehmann. Asymptotic behavior of the dimension of the Chow variety. *Advances in Mathematics*, 308:815–835, 2017.

60 Chi-Jen Lu. Hitting set generators for sparse polynomials over any finite fields. In *Proceedings of the 27th Conference on Computational Complexity*, pages 280–286, 2012.

61 Partha Mukhopadhyay. Depth-4 identity testing and Noether's normalization lemma. In *Proceedings of the 11th International Computer Science Symposium in Russia*, pages 309–323, 2016.

62 Ketan Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

63 Masayoshi Nagata. Local rings. *Interscience Tracts in Pure and Applied Mathematics*, 1962.

64 Emmy Noether. Der Endlichkeitssatz der Invarianten endlicher linearer Gruppen der Charakteristik p. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1926:28–35, 1926.

65 Shir Peleg and Amir Shpilka. A generalized Sylvester-Gallai type theorem for quadratic polynomials. In *Proceedings of the 35th Computational Complexity Conference*, 2020.

66 Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via Edelstein-Kelly type theorem for quadratic polynomials. *arXiv preprint*, 2020. arXiv:2006.08263.

67 Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009.

68 Nitin Saxena. Progress on polynomial identity testing - II. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:186, 2013. URL: http://eccc.hpi-web.de/report/2013/186.

69 Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM Journal on Computing*, 41(5):1285–1298, 2012.

70 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

71 Igor R. Shafarevich. *Basic Algebraic Geometry 1: Varieties in Projective Space*. Springer Science & Business Media, 2013.

72 Amir Shpilka. Sylvester-Gallai type theorems for quadratic polynomials. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing*, pages 1203–1214, 2019.

73 Amir Shpilka and Amir Yehudayoff. *Arithmetic Circuits: A Survey of Recent Results and Open Questions*. Now Publishers Inc, 2010.

74 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International Symposium on Symbolic and Algebraic Manipulation*, pages 216–226, 1979.

A Lower Bound for Polynomial Calculus with Extension Rule

Yaroslav Alekseev
Chebyshev Laboratory, St. Petersburg State University, Russia

Abstract

A major proof complexity problem is to prove a superpolynomial lower bound on the length of Frege proofs of arbitrary depth. A more general question is to prove an Extended Frege lower bound. Surprisingly, proving such bounds turns out to be much easier in the algebraic setting. In this paper, we study a proof system that can simulate Extended Frege: an extension of the Polynomial Calculus proof system where we can take a square root and introduce new variables that are equivalent to arbitrary depth algebraic circuits. We prove that an instance of the subset-sum principle, the binary value principle $1 + x_1 + 2x_2 + \ldots + 2^{n-1}x_n = 0$ (BVP_n), requires refutations of exponential bit size over $\mathbb{Q}$ in this system.

Part and Tzameret [18] proved an exponential lower bound on the size of Res-Lin (Resolution over linear equations [22]) refutations of BVP_n. We show that our system p-simulates Res-Lin and thus we get an alternative exponential lower bound for the size of Res-Lin refutations of BVP_n.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases proof complexity, algebraic proofs, polynomial calculus

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.21

Related Version *Full Version*: https://arxiv.org/abs/2010.05660

Funding *Yaroslav Alekseev*: research is supported by «Native towns», a social investment program of PJSC «Gazprom Neft».

Acknowledgements I would like to thank Edward A. Hirsch for guidance and useful discussions at various stages of this work. Also, I wish to thank Dmitry Itsykson and Dmitry Sokolov for very helpful comments concerning this work.

1 Introduction

In essence, the study of propositional proof complexity started with the work of Cook and Reckhow [7], which states that if there is a propositional proof system in which any unsatisfiable formula F has a short proof of unsatisfiability, then NP = CoNP. The first superpolynomial bound on the proof size was proved in a pioneering work of Tseitin [27] for regular resolution. Since then, many proof systems have been studied, some of them are logic-style (working with disjunctions, conjunctions, and other Boolean operations) and some of them are algebraic (working with arbitrary polynomials).

In this work, we consider extensions of two systems, an algebraic one and a logic-style one.

Logic-style systems

As it was mentioned before, the first superpolynomial bound on the proof size was proved in a work of Tseitin for regular resolution, which is a popular logic proof system. Lately, Haken [11] proved an exponential lower bound on the size of (unrestricted) Resolution refutation of the pigeonhole principle (PHP), expressing that there is no (total) injective map from a set with cardinality m to a set with cardinality n if $m > n$.

Since then, a stronger logic proof systems such as Frege systems were considered. But while exponential lower bounds for low-depth proof systems (both algebraic and logical ones) are known for decades, the situation with the higher depth proof systems is much worse. The present knowledge is limited to superpolynomial lower bounds for Frege systems over de Morgan basis (that is, without xor's or equivalences) of depth up to $\Theta(\log(n)/\log\log(n))$ [12] (see also [21] where a superpolynomial lower bound for systems of depth up to $\Theta(\sqrt{\log(n)})$ is proved).

Resolution with counting

Another approach to strengthen resolution is to use weak extensions in order to do some sort of counting. Res-Lin (defined in [22]) is a system working with disjunctions of linear equations, and can be viewed as a generalization of Resolution (we consider this system in the present paper). However, no truly exponential lower bounds are known for the size of refutations of formulas in CNF in (dag-like) systems that work over disjunctions of equations or inequalities (see [16] as the first paper defining these systems and containing partial results). Part and Tzameret [18] proved an exponential lower bound for (dag-like) Res-Lin refutations over $\mathbb{Q}$ for the binary value principle BVP_n. Although this is the first exponential lower bound for this system, the instance does not correspond to a translation of formula in CNF.

Itsykson and Sokolov [15] considered another extension of the resolution proof system that operates with disjunctions of linear equalities over $\mathbb{F}_2$ named Res($\oplus$) and proved an exponential lower bound on the size of tree-like Res($\oplus$)-proofs.

Algebraic proof systems

Algebraic proof systems such as Nullstellensatz were developed to use some algebraic techniques of Razborov and Smolensky [23, 25] in the proof complexity case. Lower bounds for algebraic systems started with an exponential lower bound for the Nullstellensatz [2] system. The main system considered in this paper is based on the Polynomial Calculus system [6], which is a dynamic version of Nullstellensatz. Many exponential lower bounds are known for the size of Polynomial Calculus proofs for tautologies like the Pigeonhole Principle [24, 14] and Tseitin tautologies [3]. While most results concern the representation of Boolean values by 0 and 1, there are also exponential lower bounds over the $\{-1, +1\}$ basis [26].

However, simple algebraic proof systems such as Nullstellensatz and Polynomial Calculus cannot simulate strong logic systems like Frege systems and thus cannot provide lower bounds for these systems. In order to fix this issue, strong extensions were considered: Grigoriev and Hirsch [9] considered algebraic systems over formulas. Grochow and Pitassi [10] introduced the Ideal Proof System, IPS, which can be considered as the version of Nullstellensatz where all polynomials are written as algebraic circuits (see also [19, 20] for earlier versions of this system).

Many other extensions of Polynomial Calculus and Nullstellensatz have been considered also. Buss, Impagliazzo, Krajíček, Pudlák, Razborov and Sgall [4] showed that there is a tight connection between the lengths of constant-depth Frege proofs with MOD_p gates and the length of Nullstellensatz refutations using extension axioms. Impagliazzo, Mouli and Pitassi [13] showed that a depth-3 extension of Polynomial Calculus called $\Sigma\Pi\Sigma$-PC p-simulates semantic CP* (an inequalities-based system, Cutting Planes [8, 5] with coefficients written in unary) over $\mathbb{Q}$. Also, they showed that a stronger extension of Polynomial Calculus, called Depth-k-PC, p-simulates Cutting Planes and another inequalities-based system Sum-of-Squares; the simulations can be conducted over $\mathbb{F}_{p^m}$ for an arbitrary prime number p if m is sufficiently large. However, the question about proving a superpolynomial lower bound even on the size of $\Sigma\Pi\Sigma$-PC refutations over any field remains open since it is not clear how to extend lower bound techniques such as size-degree tradeoff to this system.

1.1 Our results

We extend Polynomial Calculus with two additional rules. One rule allows us to take a square root (it was introduced by Grigoriev and Hirsch [9] in the context of transforming refutation proofs of non-Boolean formulas into derivation proofs; our motivation to take square roots is to consider an algebraic system that is at least as strong as Res-Lin even for non-Boolean formulas, see below). Another rule is an algebraic version of Tseitin's extension rule, which allows us to introduce new variables that are equivalent to arbitrary depth algebraic circuits. We will denote our generalization of Polynomial Calculus as Ext-PC$^{\surd}$. Note that Ext-PC$^{\surd}$ p-simulates Extended Frege system (since Ext-PC$^{\surd}$ p-simulates Extended Resolution and Extended Resolution p-simulates Extended Frege [17]), but it's not obvious how to p-simulate IPS refutations in Ext-PC$^{\surd}$ (since IPS refutations polynomials are written as algebraic circuits and Ext-PC$^{\surd}$ refutations are written explicitly as a sum of monomials).

In this work we give a partial positive answer to the question raised in [13] asking for a technique for proving size lower bounds on Polynomial Calculus without proving any degree lower bounds. However, our lower bound works only for field $\mathbb{Q}$ and the question about proving lower bounds over finite fields remains open. Also, we give a partial answer to another question raised in [13] by proving an exponential lower bound for the system with an extension rule even stronger than that in $\Sigma\Pi\Sigma$-PC, which is another extension of Polynomial Calculus presented in the aforementioned work.

We consider the following subset-sum instance, called Binary Value Principle (BVP_n) [1, 18]:

$$1 + x_1 + 2x_2 + \dots 2^{n-1}x_n = 0,$$

and prove an exponential lower bound for the size of Ext-PC$^{\surd}_{\mathbb{Q}}$ refutations of BVP_n. Note that Binary Value Principle does not correspond to the translation of any CNF formula and thus the question about proving the size lower bound on the refutation of formulas in CNF without proving degree lower bounds **remains open**.

▶ **Theorem 1.** *Any* Ext-PC$^{\surd}_{\mathbb{Q}}$ *refutation of* BVP_n *requires size* $2^{\Omega(n)}$.

The technique we use for proving this lower bound is similar to the technique for proving the conditional IPS lower bound in [1]. However, since Ext-PC proof system is weaker than Ideal Proof System, we get an unconditional lower bound. The main idea of the conditional lower bound in [1] is to prove the complexity lower bound on the free term in the end of the IPS-refutation of BVP_n over $\mathbb{Z}$ and then show that $\mathsf{IPS}_{\mathbb{Z}}$ simulates $\mathsf{IPS}_{\mathbb{Q}}$. One difference is that instead of concentrating on the *complexity* of computing the free term of the proof, we concentrate on the *prime numbers* being mentioned in the proof (and thus appearing as factors of the free term).

Then we consider Res-Lin and show that Ext-PC$^{\surd}_{\mathbb{Q}}$ simulates Res-Lin and thus get an alternative lower bound for Res-Lin.

▶ **Corollary 2.** *Any* $\mathsf{Res\text{-}Lin}_{\mathbb{Q}}$ *refutation of* BVP_n *requires size* $2^{\Omega(n)}$.

Note that while Part and Tzameret [18] prove an exponential lower bound on the number of lines in the proof, we prove a bound on the proof size (essentially, on the bit size of scalars appearing in the proof).

1.2 Organization of the paper

In Section 2 we recall the definition of Polynomial Calculus (PC) and give the definitions of Polynomial Calculus with square root ($\mathsf{PC}^{\surd}$) and Extended Polynomial Calculus with square root ($\mathsf{Ext\text{-}PC}^{\surd}$).

In Section 3 we prove an exponential lower bound on the size of $\mathsf{Ext\text{-}PC}^{\surd}_{\mathbb{Q}}$ refutations of BVP_n. We start with considering derivations with integer coefficients ($\mathsf{Ext\text{-}PC}^{\surd}_{\mathbb{Z}}$) and show that the free term in the end of such refutation of BVP_n is not just large but also is divisible by all primes less than 2^n (see Theorem 9). Then, in Theorem 11, we convert proofs over $\mathbb{Q}$ into proofs over $\mathbb{Z}$ without changing the set of primes mentioned in the proof and thus get an $\mathsf{Ext\text{-}PC}^{\surd}_{\mathbb{Q}}$ lower bound.

In Section 4 we show that $\mathsf{Ext\text{-}PC}^{\surd}_{\mathbb{Q}}$ simulates Res-Lin and thus we get an alternative lower bound for the size of Res-Lin refutations of BVP_n.

2 Preliminaries

In this paper we are going to work with polynomials over integers or rationals. We define the size of a polynomial roughly as the total length of the bit representation of its coefficients:

▶ **Definition 3** (Size of a polynomial). *Let f be an arbitrary integer or rational polynomial in variables $\{x_1, \dots, x_n\}$.*

- *If $f \in \mathbb{Z}[x_1, \dots, x_n]$ then $Size(f) = \sum(\lceil \log |a_i| \rceil + 1)$ where a_i are the coefficients of f.*
- *If $f \in \mathbb{Q}[x_1, \dots, x_n]$ then $Size(f) = \sum(\lceil \log |p_i| \rceil + \lceil \log |q_i| \rceil + 1)$ where $p_i \in \mathbb{Z}$, $q_i \in \mathbb{N}$ and $\frac{p_i}{q_i}$ are the coefficients of f.*

▶ **Definition 4** (Polynomial Calculus). *Let $\Gamma = \{P_1, \dots, P_m\} \subset \mathbb{F}[x_1, \dots, x_n]$ be a set of polynomials in variables $\{x_1, \dots, x_n\}$ over a field $\mathbb{F}$ such that the system of equations $P_1 = 0, \dots, P_m = 0$ has no solution. A Polynomial Calculus refutation of Γ is a sequence of polynomials $R_1, \dots, R_s$ where $R_s = 1$ and for every l in $\{1, \dots, s\}$, $R_l \in \Gamma$ or is obtained through one of the following derivation rules for $j, k < l$*

- *$R_l = \alpha R_j + \beta R_k$ for $\alpha, \beta \in \mathbb{F}$*
- *$R_l = x_i R_k$*

The size of the refutation is $\sum_{l=1}^{s} Size(R_l)$. The degree of the refutation is $\max_l deg(R_l)$.

Now we consider a variant of Polynomial Calculus proof system with additional **square root derivation rule** (see [9]). Moreover, we extend our definition from fields to **rings**.

▶ **Definition 5** (Polynomial Calculus with square root). *Let $\Gamma = \{P_1, \dots, P_m\} \subset R[x_1, \dots, x_n]$ be a set of polynomials in variables $\{x_1, \dots, x_n\}$ over a domain R such that the system of equations $P_1 = 0, \dots, P_m = 0$ has no solution. A $\mathsf{PC}^{\surd}_{R}$ refutation of Γ is a sequence of polynomials $R_1, \dots, R_s$ where $R_s = M$ for some constant $M \in R, M \neq 0$ and for every l in $\{1, \dots, s\}$, $R_l \in \Gamma$ or is obtained through one of the following derivation rules for $j, k < l$*

- *$R_l = \alpha R_j + \beta R_k$ for $\alpha, \beta \in R$*
- *$R_l = x_i R_k$ for some $i \in \{1, \dots, n\}$*
- *$R_l^2 = R_k$ (which means that we can take square root of a polynomial if and only if it is a square of some other polynomial)*

The size of the refutation is $\sum_{l=1}^{s} Size(R_l)$, where $Size(R_l)$ is the size of the polynomial R_l. The degree of the refutation is $\max_l deg(R_l)$.

▶ Note 6. We will consider $\mathbb{Q}$ or $\mathbb{Z}$ as the ring R. For both of those rings, if we consider **Boolean** case, where axioms $x_i^2 - x_i = 0$ added, our system will be complete, which means that for every unsatisfiable over $\{0, 1\}$ assignment system $\{f_i(\vec{x}) = 0\}$ there is a $\mathsf{PC}^{\surd}_{R}$ refutation. Also, note that if R is a domain and $P^2 = 0$ for some $P \in R[\vec{x}]$, then $P = 0$.

We now define a variant of $\mathsf{PC}_R^{\surd}$, $\mathsf{Ext\text{-}PC}_R^{\surd}$ where the proof system is additionally allowed to introduce new variables y_i corresponding to arbitrary polynomials in the original variables x_i.

▶ **Definition 7** (Extended Polynomial Calculus with square root)**.** *Let $\Gamma = \{P_1, \dots, P_m\} \subset R[x_1, \dots, x_n]$ be a set of polynomials in variables $\{x_1, \dots, x_n\}$ over a domain R such that the system of equations $P_1 = 0, \dots, P_m = 0$ has no solution. A $\mathsf{Ext\text{-}PC}_R^{\surd}$ refutation of Γ is a $\mathsf{PC}_R^{\surd}$ refutation of a set*

$$\Gamma' = \{P_1, \dots, P_m, y_1 - Q_1(x_1, \dots, x_n), y_2 - Q_2(x_1, \dots, x_n, y_1), \dots,\\ y_m - Q_m(x_1, \dots, x_n, y_1, \dots, y_{m-1})\}$$

where $Q_i \in R[\vec{x}, y_1, \dots, y_{i-1}]$ are arbitrary polynomials.

The size of the $\mathsf{Ext\text{-}PC}_R^{\surd}$ refutation is equal to the size of the $\mathsf{PC}_R^{\surd}$ refutation of Γ'.

3 Lower bound

In order to prove the lower bound for the $\mathsf{Ext\text{-}PC}_{\mathbb{Q}}^{\surd}$ proof system, we consider the following subset-sum instance [1, 18]:

▶ **Definition 8** (Binary Value Principle BVP_n)**.** *The* ***binary value principle*** *over the variables $x_1, \dots, x_n$, BVP_n for short, is the following unsatisfiable system of equations:*

$$x_1 + 2x_2 + \dots 2^{n-1}x_n + 1 = 0,$$

$$x_1^2 - x_1 = 0,\ x_2^2 - x_2 = 0,\ \dots,\ x_n^2 - x_n = 0.$$

▶ **Theorem 9.** *Any $\mathsf{Ext\text{-}PC}_{\mathbb{Z}}^{\surd}$ refutation of BVP_n requires size $\Omega(2^n)$. Moreover, the absolute value of the constant in the end of our $\mathsf{Ext\text{-}PC}_{\mathbb{Z}}^{\surd}$ refutation consists of at least $C \cdot 2^n$ bits for some constant $C > 0$. Also, the constant in the end of our $\mathsf{Ext\text{-}PC}_{\mathbb{Z}}^{\surd}$ refutation is divisible by every prime number less than 2^n.*

Proof. Assume that $\{R_1, \dots, R_t\}$ is an $\mathsf{Ext\text{-}PC}_{\mathbb{Z}}^{\surd}$ refutation of BVP_n. Then we know that $\{R_1, \dots, R_t\}$ is a $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ refutation of some set

$$\Gamma' = \{G(\vec{x}), F_1(\vec{x}), \dots, F_n(\vec{x}), y_1 - Q_1(\vec{x}), \dots y_m - Q_m(\vec{x}, y_1, \dots, y_{m-1})\}$$

where $G(\vec{x}) = 1 + \sum_{i=1}^{i=n} 2^{(i-1)} x_i$, $F_i(\vec{x}) = x_i^2 - x_i$ and $Q_i \in \mathbb{Z}[\vec{x}, y_1, \dots, y_{i-1}]$.

By the definition of an $\mathsf{Ext\text{-}PC}_{\mathbb{Z}}^{\surd}$ refutation we know that there exists an integer constant $M \neq 0$ such that $R_t = M$.

▷ Claim 10. M is divisible by every prime number less than 2^n.

Proof of claim. Consider arbitrary integer number $0 \le k < 2^n$ and its binary representation $b_1, \dots, b_n$. Let $k+1$ be **prime**. Then $G(b_1, \dots, b_n) = k + 1$, $F_i(b_1, \dots, b_n) = b_i^2 - b_i = 0$. Also consider integers $c_1, \dots, c_m$ such that $c_i = Q_i(b_1, \dots, b_n, c_1, c_2, \dots, c_{i-1})$. Now we will prove by induction that every integer number $R_i(b_1, \dots, b_n, c_1, \dots, c_m)$ is divisible by $k+1$ and thus M is divisible by every prime number less than 2^n.

Base case: if $i = 1$, then

$$R_i = G(b_1, \ldots, b_n, c_1, \ldots, c_m) = k + 1$$

or

$$R_i = F_i(b_1, \ldots, b_n, c_1, \ldots, c_m) = 0$$

or

$$R_i(b_1, \ldots, b_n, c_1, \ldots, c_m) = c_i - Q_i(b_1, \ldots, b_n, c_1, \ldots, c_{i-1}) = 0$$

which means that R_i is divisible by $k + 1$.

Induction step: suppose we know that R_j is divisible by $k + 1$ for any $j \leq i$. Now we will show it for R_{i+1}. There are four cases:

1. If $R_{i+1} \in \Gamma'$, then this case is equivalent to the base case and $R_{i+1}(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by $k + 1$.
2. If $R_{i+1} = \alpha R_j + \beta R_s$ for $\alpha, \beta \in \mathbb{Z}$ and $j, s \leq i$, then $R_{i+1}(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by $k + 1$ because $R_j(b_1, \ldots, b_n, c_1, \ldots, c_m)$ and $R_s(b_1, \ldots, b_n, c_1, \ldots, c_m)$ are divisible by $k + 1$ and α and β are integers.
3. If $R_{i+1} = x_j R_s$ or $R_{i+1} = y_j R_s$, then $R_{i+1}(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by $k + 1$ because $R_s(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by $k + 1$ and b_i and c_i are integers.
4. If $R_{i+1}^2 = R_s$, then we know that $R_s(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by $k + 1$. Suppose $R_{i+1}(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is not divisible by $k + 1$. Then $R_{i+1}(b_1, \ldots, b_n, c_1, \ldots, c_m)^2$ is not divisible by $k + 1$ since $k + 1$ is **prime**. But $R_{i+1}(b_1, \ldots, b_n, c_1, \ldots, c_m)^2 = R_s(b_1, \ldots, b_n, c_1, \ldots, c_m)$ which leads us to a contradiction.

Since every $R_i(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by $k + 1$, we know that $M = R_t(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by every $k + 1$ less than 2^n, and in particular M is divisible by every prime number less than 2^n.

So we know that M is divisible by the product of all prime numbers less than 2^n. Then we know that $|M| > (\pi(2^n))!$ where $\pi(2^n)$ is the number of all prime numbers less than 2^n. By the prime number theorem $\pi(2^n) > C\frac{2^n}{n}$. By Stirling's approximation we get

$$|M| > \left(C\frac{2^n}{n}\right)! > C' \cdot \left(C\frac{2^n}{e \cdot n}\right)^{C\frac{2^n}{n}} > C'' \left(2^{\frac{n}{2}}\right)^{C\frac{2^n}{n}} > C'' 2^{(2^n C_0)}$$

which means that M consists of at least $C_1 \cdot 2^n$ bits and therefore any $\text{Ext-PC}_{\mathbb{Z}}^{\surd}$ refutation of BVP_n requires size $\Omega(2^n)$. ◁

◀

In order to prove a lower bound over $\mathbb{Q}$, we need to convert an $\text{Ext-PC}_{\mathbb{Q}}^{\surd}$ proof into an $\text{Ext-PC}_{\mathbb{Z}}^{\surd}$ proof. The key idea of this translation is that we can create an $\text{Ext-PC}_{\mathbb{Z}}^{\surd}$ proof in which the constant in the end is a multiplication of some constants occurring in the original $\text{Ext-PC}_{\mathbb{Q}}^{\surd}$ refutation. Since the constant in the end of the $\text{Ext-PC}_{\mathbb{Z}}^{\surd}$ refutation is divisible by all prime numbers less then 2^n, we get a lower bound on the size of constants occurring in the $\text{Ext-PC}_{\mathbb{Q}}^{\surd}$ refutation and hence on the size of the refutation itself.

▶ **Theorem 11.** *Any $\text{Ext-PC}_{\mathbb{Q}}^{\surd}$ refutation of BVP_n requires size $\Omega(2^n)$.*

Proof. Assume that $\{R_1, \ldots, R_t\}$ is an $\mathsf{Ext\text{-}PC}^{\surd}_{\mathbb{Q}}$ refutation of Γ of the size S. Then we know that $\{R_1, \ldots, R_t\}$ is a $\mathsf{PC}^{\surd}_{\mathbb{Q}}$ refutation of some set $\Gamma' = \Gamma \cup \{y_1 - Q_1(\vec{x}), \ldots, y_m - Q_m(\vec{x}, y_1, \ldots, y_{m-1})\}$ where $Q_i \in \mathbb{Q}[\vec{x}, \vec{y}]$. Also, we know that $R_t = M$ for some $M \in \mathbb{Q}$.

Consider integers $M_1, \ldots, M_m$ where M_i is equal to the product of denominators of all coefficients of polynomial Q_i. Also consider all polynomials $R_j(\vec{x}, \vec{y})$ which was derived by using linear combination rule which means that $R_j = \alpha R_i + \beta R_k$. Then we consider **all** constants α and β occurring in linear combination derivations in our proof. Let's denote the set of those constants as $\{\gamma_1, \gamma_2, \ldots, \gamma_l\} \subset \mathbb{Q}$. Now consider the set of all **denominators** of the constants in $\{\gamma_1, \gamma_2, \ldots, \gamma_l\}$ and denote this set as $\{\delta_1, \delta_2, \ldots, \delta_l\} \subset \mathbb{N}$.

Also consider the products of all denominators of coefficients of polynomials $\{R_1, \ldots, R_t\}$. We will denote the set of those integers as $\{L_1, \ldots, L_t\} \subset \mathbb{N}$.

Now we will construct the $\mathsf{Ext\text{-}PC}^{\surd}_{\mathbb{Z}}$ refutation of Γ such that the constant in the end of this proof is equal to $M_1^{c_1} \cdot M_2^{c_2} \cdots M_m^{c_m} \cdot \delta_1^{c_{m+1}} \cdots \delta_l^{c_{m+l}} \cdot L_1^{c_{m+l+1}} \cdots L_t^{c_{m+l+t}} \cdot M$ where $\{c_1, c_2, \ldots, c_{m+l+t}\} \subset \mathbb{N} \cup \{0\}$.

Firstly, we will translate polynomials Q_i into some integer polynomials Q'_i. Consider $Q'_1(\vec{x}) = M_1 \cdot Q_1(\vec{x})$ where M_1 is equal to the product of denominators of all coefficients of the polynomial Q_1. Then $Q'_1 \in \mathbb{Z}[\vec{x}]$ and $T_1 = M_1$. Then consider $Q'_2(\vec{x}, y'_1) = T_2 \cdot Q_2(\vec{x}, \frac{y'_1}{T_1})$ where T_2 is equal to $T_1^{\alpha_{11}} \cdot M_2$ where α_{11} is an **arbitrary** non-negative integer such that $Q'_2 \in \mathbb{Z}[\vec{x}, y'_1]$. Then for every i we consider $Q'_i(\vec{x}, y'_1, \ldots, y'_{i-1}) = T_i \cdot Q_i(\vec{x}, \frac{y'_1}{T_1}, \ldots, \frac{y'_{i-1}}{T_{i-1}})$ where $T_i = T_1^{\alpha_{i1}} \cdot T_2^{\alpha_{i2}} \cdots T_{i-1}^{\alpha_{ii-1}} \cdot M_i$ where $\alpha_{i1}, \ldots, \alpha_{ii-1}$ are **arbitrary** integers such that $Q'_i \in \mathbb{Z}[\vec{x}, y'_1, \ldots, y'_{i-1}]$. Note that we are not interested in the size of the integers α_{ij} so they could be arbitrary large.

Now we will construct a $\mathsf{PC}^{\surd}_{\mathbb{Q}}$ refutation $\{R'_1, \ldots, R'_s\}$ of the set $\Gamma'' = \Gamma \cup \{y'_1 - Q'_1(\vec{x}), \ldots y'_m - Q'_m(\vec{x}, y'_1, \ldots, y'_{m-1})\}$ of the following form: this refutation duplicates the original refutation $\{R_1, \ldots, R_t\}$ in all cases except when the polynomial R_i was derived by multiplying by some variable y_j from some polynomial R_k. In this case we will multiply corresponding polynomial by y'_j and then multiply it by $\frac{1}{T_j}$.

Formally, we will prove the following claim:

▷ Claim 12. There is a $\mathsf{PC}^{\surd}_{\mathbb{Q}}$ refutation $\{R'_1, \ldots, R'_s\}$ of the set $\Gamma'' = \Gamma \cup \{y'_1 - Q'_1(\vec{x}), \ldots y'_m - Q'_m(\vec{x}, y'_1, \ldots, y'_{m-1})\}$ for which the following properties holds:

- For every polynomial $R'_i(\vec{x}, y'_1, \ldots, y'_m)$ one of the following equations holds: $R'_i(\vec{x}, y_1 \cdot T_1, \ldots, y_m \cdot T_m) = R_j(\vec{x}, y_1, \ldots, y_m)$ for some j or $R'_i(\vec{x}, y_1 \cdot T_1, \ldots, y_m \cdot T_m) = T_k \cdot R_j(\vec{x}, y_1, \ldots, y_m)$ for some k and j.
- If $R'_i(\vec{x}, y'_1, \ldots, y'_m)$ was derived from $R'_j(\vec{x}, y'_1, \ldots, y'_m)$ and $R'_k(\vec{x}, y_1, \ldots, y_m)$ by taking linear combination with rational constants α and β (which means that $R'_i = \alpha R'_j + \beta R'_k$), then $\alpha = \frac{1}{T_f}$ and $\beta = 0$ for some f or there is some polynomial $R_h(\vec{x}, y'_1, \ldots, y'_m)$ which was derived from some polynomials R_k and R_l by using linear combination with constants α and β.

Proof of claim. The proof is an easy (but lengthy) inductive argument and is given in the Appendix A. ◁

Now we will show that Γ'' has a $\mathsf{PC}^{\surd}_{\mathbb{Z}}$ refutation in which the constant in the end is equal to

$$M_1^{c_1} \cdot M_2^{c_2} \cdots M_m^{c_m} \cdot \delta_1^{c_{m+1}} \cdots \delta_l^{c_{m+l}} \cdot L_1^{c_{m+l+1}} \cdots L_t^{c_{m+l+t}} \cdot M.$$

In order to do this we will fix a $\mathsf{PC}_{\mathbb{Q}}^{\surd}$ refutation $\{R'_1, \dots, R'_s\}$ of Γ'' with the properties from the Claim 12 and construct a $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ refutation of Γ'' by induction. Moreover, we will construct a $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ refutation $\{R''_1, \dots, R''_f\}$ in which every polynomial R''_i is equal to $M_1^{d_1} \cdot M_2^{d_2} \cdots M_m^{d_m} \cdot \delta_1^{d_{m+1}} \cdots \delta_l^{d_{m+l}} \cdot L_1^{d_{m+l+1}} \cdots L_t^{d_{m+l+t}} \cdot R'_i$ for some non-negative integers $d_1, \dots, d_{m+l+t}$ and some polynomial R'_i.

Informally, we are going to multiply each line in our $\mathsf{PC}_{\mathbb{Q}}^{\surd}$ refutation by some constant in order to get a correct $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ refutation. But since we cannot divide polynomials in our $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ refutation by any constant, we will duplicate original $\mathsf{PC}_{\mathbb{Q}}^{\surd}$ refutation multiplied by some constant of the form $M_1^{d_1} \cdot M_2^{d_2} \cdots M_m^{d_m} \cdot \delta_1^{d_{m+1}} \cdots \delta_l^{d_{m+l}} \cdot L_1^{d_{m+l+1}} \cdots L_t^{d_{m+l+t}}$ every time we would like to simulate derivation in the original proof.

Induction statement. Let $\{R'_1, \dots, R'_i\}$ be a $\mathsf{PC}_{\mathbb{Q}}^{\surd}$ derivation from Γ'' with the properties from the Claim 12. Then there exists a $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ derivation $\{R''_1, \dots, R''_f\}$ from Γ'' such that

- $f \leq 2i^2$.
- There is some constant $F_i = M_1^{b_1} \cdot M_2^{b_2} \cdots M_m^{b_m} \cdot \delta_1^{b_{m+1}} \cdots \delta_l^{b_{m+l}} \cdot L_1^{b_{m+l+1}} \cdots L_t^{b_{m+l+t}} \in \mathbb{N}$ such that

$$F_i \cdot R'_1 = R''_{f-i+1},\; F_i \cdot R'_2 = R''_{f-i+2},\; \dots,\; F_i \cdot R'_i = R''_f$$

Both *base case of induction* and *induction step* are straightforward derivations and are given in the Appendix B.

So now we have a $\mathsf{Ext\text{-}PC}_{\mathbb{Z}}^{\surd}$ refutation of Γ such that the constant in the end of this refutation is equal to $M_1^{c_1} \cdot M_2^{c_2} \cdots M_m^{c_m} \cdot \delta_1^{c_{m+1}} \cdots \delta_l^{c_{m+l}} \cdot L_1^{c_{m+l+1}} \cdots L_t^{c_{m+l+t}} \cdot M$. Suppose that $M = \frac{p'}{q'}$ where $p \in \mathbb{Z}$ and $q \in \mathbb{N}$. Then, from Theorem 9 we know that $M_1^{c_1} \cdot M_2^{c_2} \cdots M_m^{c_m} \cdot \delta_1^{c_{m+1}} \cdots \delta_f^{c_{m+l}} \cdot L_1^{c_{m+l+1}} \cdots L_t^{c_{m+l+t}} \cdot p'$ is divisible by every prime number less than 2^n. Since $M_1, \dots, M_m$, $\delta_1, \dots, \delta_l$, $L_1, \dots, L_t$ are positive integers we know that $M_1 \cdot M_2 \cdots M_m \cdot \delta_1 \cdots \delta_l \cdot L_1 \cdots L_t \cdot p'$ is divisible by every prime number less than 2^n. Also we know that

$$\log\lceil M_1\rceil + \cdots + \log\lceil M_m\rceil + \log\lceil \delta_1\rceil + \cdots + \log\lceil \delta_l\rceil + \log\lceil L_1\rceil + \cdots + \log\lceil L_t\rceil + \log\lceil p\rceil \leq O(Size(S))$$

because all constants $M_1, \dots, M_m, L_1, \dots, L_t$ are products of denominators in the lines of our refutation $\{R_1, \dots, R_t\}$ and constants $\delta_1, \dots, \delta_l$ are denominators of rationals in linear combinations used in our derivation.

On the other hand, we know that

$$M_1 \cdot M_2 \cdots M_m \cdot \delta_1 \cdots \delta_l \cdot L_1 \cdots L_t \cdot p' \geq 2^{2^{\Omega(n)}}$$

since our product is divisible by every prime number less than 2^n. Then we know that $S \geq 2^{\Omega(n)}$. ◀

4 Connection between Res-Lin, $\mathsf{Ext\text{-}PC}_{\mathbb{Q}}^{\surd}$ and $\mathsf{Ext\text{-}PC}_{\mathbb{Q}}$

Following [22], we define Res-Lin proof system.

▶ **Definition 13.** *A* ***disjunction of linear equations*** *is of the following general form:*

$$(a_1^{(1)}x_1 + \ldots + a_n^{(1)}x_n = a_0^{(1)}) \vee \cdots \vee (a_1^{(t)}x_1 + \ldots + a_n^{(t)}x_n = a_0^{(t)}) \qquad (1)$$

where $t \geq 0$ and the coefficients a_i^j are ***integers*** *(for all $0 \leq i \leq n$, $1 \leq j \leq t$). The semantics of such a disjunction is the natural one: We say that an assignment of integral values to the variables $x_1, \dots, x_n$ satisfies (1) if and only if there exists $j \in \{1, \dots, t\}$ so that the equation $a_1^{(j)}x_1 + \ldots + a_n^{(j)}x_n = a_0^{(j)}$ holds under the given assignment.*

*The **size** of the disjunction of linear equations is $\sum_{i=1}^{n}\sum_{j=1}^{t}|a_i^{(j)}|$ if all coefficients are written in **unary** notation. If all coefficients are written in **binary** notation then the **size** is equal to $\sum_{i=1}^{n}\sum_{j=1}^{t}(\lceil\log|a_i^{(j)}|\rceil+1)$.*

▶ **Definition 14.** *Let $K := \{K_1, \ldots, K_m\}$ be a collection of disjunctions of linear equations. An* Res-Lin *proof from K of a disjunction of linear equations D is a finite sequence $\pi = (D_1, \ldots, D_l)$ of disjunctions of linear equations, such that $D_l = D$ and for every $i \in \{1, \ldots, l\}$, either $D_i = K_j$ for some $j \in \{1, \ldots, m\}$, or D_i is a Boolean axiom $(x_h = 0) \vee (x_h = 1)$ for some $h \in \{1, \ldots, n\}$, or D_i was deduced by one of the following* Res-Lin *inference rules, using D_j, D_k for some $j, k < i$:*

- ***Resolution**: Let A, B be two, possibly empty, disjunctions of linear equations and let L_1, L_2 be two linear equations. From $A \vee L_1$ and $B \vee L_2$ derive $A \vee B \vee (\alpha L_1 + \beta L_2)$ where $\alpha, \beta \in \mathbb{Z}$.*
- ***Weakening**: From a (possibly empty) disjunction of linear equations A derive $A \vee L$, where L is an arbitrary linear equation over $\{x_1, \ldots, x_n\}$.*
- ***Simplification**: From $A \vee (k = 0)$ derive A, where A is a, possibly empty, disjunction of linear equations and $k \neq 0$ is a constant.*
- ***Contraction**: From $A \vee L \vee L$ derive $A \vee L$, where A is a, possibly empty, disjunction of linear equations and L is some linear equation.*

Note that we assume that the order of equations in the disjunction is not significant, while we contract identical equations, especially.

An Res-Lin ***refutation** of a collection of disjunctions of linear equations K is a proof of the empty disjunction from K. The **size** of an* Res-Lin *proof π is the total size of all the disjunctions of linear equations in π.*

If all coefficients in our Res-Lin *proof π are written in the **unary** notation then we denote this proof an* Res-Lin$_U$ *derivation. Otherwise, if all coefficients are written in the **binary** notation then we denote this proof an* Res-Lin$_B$ *derivation.*

▶ Note 15. In the original Res-Lin proof system duplicate linear equations can be discarded from the disjunction. Instead, we will use **contraction** rule explicitly. It is easy to see that both these variants of Res-Lin system are equivalent.

▶ **Definition 16.** *Let D be a disjunction of linear equations:*

$$(a_1^{(1)}x_1 + \ldots + a_n^{(1)}x_n = a_0^{(1)}) \vee \cdots \vee (a_1^{(t)}x_1 + \ldots + a_n^{(t)}x_n = a_0^{(t)})$$

We denote by $\widehat{D}$ its translation into the following system of polynomial equations:

$$y_1 \cdot y_2 \cdots y_t = 0$$

$$y_1 = a_1^{(1)}x_1 + \ldots + a_n^{(1)}x_n - a_0^{(1)},\ y_2 = a_1^{(2)}x_1 + \ldots + a_n^{(2)}x_n - a_0^{(2)},\ \ldots,$$
$$y_t = a_1^{(t)}x_1 + \ldots + a_n^{(t)}x_n - a_0^{(t)}$$

If D is the empty disjunction, we define $\widehat{D}$ to be the single polynomial equation 1 = 0.

Now we will prove that Ext-PC$_{\mathbb{Q}}^{\surd}$ p-simulates Res-Lin$_B$ and $\Sigma\Pi\Sigma$-$PC_{\mathbb{Q}}$ p-simulates Res-Lin$_U$.

▶ **Theorem 17.** *Let $\pi = (D_1, \ldots, D_l)$ be an* Res-Lin$_B$ *proof sequence of D_l from some collection of initial disjunctions of linear equations $Q_1, \ldots, Q_m$. Also consider $L_1, \ldots, L_t$ – all affine forms that we have in all disjunctions in our* Res-Lin$_B$ *proof sequence.*

Then, there exists a PC$_{\mathbb{Q}}^{\surd}$ *proof of $\widehat{D}_l$ from $\widehat{Q}_1 \cup \ldots \cup \widehat{Q}_m \cup \{y_1 = L_1, y_2 = L_2, \ldots, y_t = L_t\}$ of size at most $O(p(Size(\pi)))$ for some polynomial p.*

Proof. The proof is a straightforward induction and is given in the Appendix C. ◀

Following [13], we define the $\Sigma\Pi\Sigma$-PC_R proof system.

▶ **Definition 18** ([13]). *Let $\Gamma = \{P_1, \ldots, P_m\} \subset R[x_1, \ldots, x_n]$ be a set of polynomials in variables $\{x_1, \ldots, x_n\}$ over a ring R such that the system of equations $P_1 = 0, \ldots, P_m = 0$ has no solution. A $\Sigma\Pi\Sigma$-PC_R refutation of Γ is a PC_R refutation of a set $\Gamma' = \{P_1, \ldots, P_m, Q_1, \ldots, Q_m\}$ where Q_i are polynomials of the form $Q_i = y_i - (a_{i0} + \sum_j a_{ij}x_j)$ for some constants $a_{ij} \in R$.*

The size of the $\Sigma\Pi\Sigma$-PC_R refutation is equal to the size of the PC_R refutation of Γ'.

▶ **Theorem 19.** *Let $\pi = (D_1, \ldots, D_l)$ be an $\mathsf{Res\text{-}Lin}_U$ proof sequence of D_l, from some collection of initial disjunctions of linear equations $Q_1, \ldots, Q_m$. Then, there exists a $\Sigma\Pi\Sigma$-$PC_{\mathbb{Q}}$ proof of $\widehat{D}_l$ from $\widehat{Q}_1 \cup \ldots \cup \widehat{Q}_m$ of size at most $O(p(Size(\pi)))$ for some polynomial p.*

Proof. To prove this theorem we will use the following lemma from [13]:

▶ **Lemma 20** ([13], revision 2 of the ECCC report, lemma 7, p.32). *Let $\Gamma = \{P_1, \ldots, P_a, Q_1, \ldots, Q_b, X, Y\}$ be a set of polynomials such that*

$$P_1 = x_1 - (x-1),\ P_2 = x_2 - (x-2),\ \ldots, P_a = x_a - (x-a),$$

$$Q_1 = y_1 - (y-1),\ Q_2 = y_2 - (y-2),\ \ldots, Q_b = y_b - (y-b),$$

$$X = x \cdot x_1 \cdot x_2 \cdots x_a,\ Y = y \cdot y_1 \cdot y_2 \cdots y_b.$$

Then we can introduce new variables $z, z_1, \ldots, z_{a+b}$ using the $\Sigma\Pi\Sigma$-$PC_{\mathbb{Q}}$ extension rule and derive Γ' from Γ in $\Sigma\Pi\Sigma$-$PC_{\mathbb{Q}}$ with a derivation of size $poly(ab)$, where $\Gamma' = \{Z_0, Z_1, \ldots, Z_{a+b}, Z\}$ and

$$Z_0 = z-(x+y),\ Z_1 = z_1-(x+y+1),\ Z_2 = z_2-(x+y+2),\ \ldots, Z_{a+b} = z_{a+b}-(x+y+a+b),$$

$$Z = z \cdot z_1 \cdot z_2 \cdots z_{a+b}.$$

Now we will prove the theorem by induction on lines in π.
Base case: An $\mathsf{Res\text{-}Lin}_U$ axiom Q_i is translated into $\widehat{Q_i}$ and $\mathsf{Res\text{-}Lin}_U$ Boolean axiom $(x_i = 0) \vee (x_i = 1)$ is translated into PC axiom $x_i^2 - x_i = 0$.
Induction step: Now we will simulate all $\mathsf{Res\text{-}Lin}_U$ derivation rules in the $\Sigma\Pi\Sigma$-$PC_{\mathbb{Q}}$ proof.

- **Resolution**, **Weakening**, **Simplification** rules simulation is the same as in Theorem 17.
- **Contraction**: Assume that $D_i = A \vee L$ and $D_j = A \vee L \vee L$ where L is a linear equation. Then, we have already derived polynomial equations

  $$y_{j1} = (a_{j1}^{(1)}x_1 + \ldots + a_{jn}^{(1)}x_n - a_{j0}^{(1)}),\ \ldots,\ y_{jt_j-1} = y_{jt_j} = (a_{j1}^{(t_j)}x_1 + \ldots + a_{jn}^{(t_j)}x_n - a_{j0}^{(t_j)}),$$

  $$y_{j1} \cdot y_{j2} \cdots y_{jt_j-1} \cdot y_{jt_j} = 0.$$

  Then we can derive $y_{jt_j-1} = y_{jt_j}$ and $y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot (y_{jt_j-1}^2) = 0$. Using lemma we can introduce new variables $\{z_{-M}, \ldots, z_M\}$ and derive

  $$z_{-M} = y_{jt_{j-1}} + M,\ , z_{-M+1} = y_{jt_{j-1}} + M - 1, \ldots, z_0 = y_{jt_{j-1}},\ z_M = y_{jt_{j-1}} - M,$$

  $$z_{-M} \cdot z_{-M+1} \cdots z_{M-1} \cdot z_M = 0,$$

  where $M = |a_{j1}^{(t_{j-1})}| + |a_{j2}^{(t_{j-1})}| + \ldots + |a_{jn}^{(t_{j-1})}|$. Then we can substitute $y_{jt_j} - k$ for each z_k one by one and get equation

  $$f(y_{jt_{j-1}}) = 0$$

where $f(y_{jt_{j-1}}) = b_1 \cdot y_{jt_{j-1}} + b_2 \cdot y^2_{jt_{j-1}} + \ldots + b_{2M+1} \cdot y^{2M+1}_{jt_{j-1}}$ is some polynomial from $\mathbb{Z}[y_{jt_{j-1}}]$ and $b_1 = (M!)^2 \cdot (-1)^M$. Then we can derive the following equation by using multiplication rule:

$$y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot f(y_{jt_{j-1}}) = b_1 \cdot y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot y_{jt_j-1} + \\ + y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot (y^2_{jt_j-1}) \cdot (b_2 + b_3 \cdot y_{jt_{j-1}} + \ldots + b_{2M+1} \cdot y^{2M-1}_{jt_{j-1}}) = 0.$$

Now, using the equation $y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot (y^2_{jt_j-1}) = 0$ we can derive $b_1 \cdot y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot y_{jt_j-1} = 0$ and since $b_1 \neq 0$ we can derive $y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot y_{jt_j-1} = 0$. This equation is the last part of $\widehat{D}_i$ because other parts were derived earlier.

◀

Now we will show that our lower bound provides an interesting counterpart to a result from [18].

▶ **Theorem 21** ([18]). *Any* Res-Lin$_B$ *refutation of* $1 + 2x_1 + \ldots + 2^n x_n = 0$ *is of the size* $2^{\Omega(n)}$.

Proof. From Theorem 11 we know that any Ext-PC$^{\surd}_{\mathbb{Q}}$ refutation of BVP$_n$ requires size $2^{\Omega(n)}$ and thus from Theorem 17 we know that there is some polynomial p such that for any Res-Lin$_B$ refutation of BVP$_n$ of size S the equation $p(S) \geq C_0 \cdot 2^{C_1 \cdot n}$ holds. Then we know that for some constant C the equation $S \geq 2^{C \cdot n}$ holds. ◀

Also we will show that there is no straightforward translation of Res-Lin$_B$ derivations into Ext-PC$_{\mathbb{Q}}$ refutations.

▶ **Theorem 22.** *Any* Ext-PC$_{\mathbb{Q}}$*-derivation of* $1 + x_1 + \ldots + 2^{n-1}x_n = 0$ *from equation* $(1 + x_1 + \ldots + 2^{n-1}x_n)^2 = 0$ *requires size* $2^{\Omega(n)}$.

Proof. The proof of this theorem essentially copies the proof of Theorem 11 and consists of two parts. In the first part we prove that if we have an Ext-PC$_{\mathbb{Z}}$-derivation of $M \cdot (1 + x_1 + \ldots + 2^{n-1}x_n) = 0$ from equation $(1 + x_1 + \ldots + 2^{n-1}x_n)^2 = 0$ where $M \in \mathbb{Z}, M \neq 0$, then M is divisible by every prime number less than 2^n.

In the second part we prove that for every Ext-PC$_{\mathbb{Q}}$-derivation of $(1+x_1+\ldots+2^{n-1}x_n) = 0$ from equation $(1+x_1+\ldots+2^{n-1}x_n)^2 = 0$ there is an Ext-PC$_{\mathbb{Z}}$-derivation of $M_1^{\alpha_1} \cdots M_k^{\alpha_k} \cdot (1+x_1+\ldots+2^{n-1}x_n) = 0$ from equation $(1+x_1+\ldots+2^{n-1}x_n)^2 = 0$ where $M_i \in \mathbb{Z}, M_i \neq 0$ and M_i are denominators from the original Ext-PC$_{\mathbb{Q}}$-derivation. Then we know that $M_1 \cdots M_k$ is divisible by all prime numbers less than 2^n and thus the size of the original Ext-PC$_{\mathbb{Q}}$-derivation was $2^{\Omega(n)}$.

For the full proof see Appendix D. ◀

Open Problems

1. Theorem 17 says that Ext-PC$^{\surd}_{\mathbb{Q}}$ p-simulates any Res-Lin$_B$ derivation. However, from Theorem 22 we know that simulation from Theorem 17 doesn't work for Ext-PC$_{\mathbb{Q}}$. Is the square root rule necessary, that is, can we p-simulate the Res-Lin$_B$ refutation in the Ext-PC$_{\mathbb{Q}}$ proof system?
2. A major question is to prove an exponential lower bound on the size of the $\Sigma\Pi\Sigma$-PC$_{\mathbb{Q}}$ refutation of a translation of formula in CNF.

3. Theorem 21 says that any $\mathsf{Res\text{-}Lin}_B$ refutation of BVP_n requires size $2^{\Omega(n)}$. Does the exponential lower bound on the size of the $\mathsf{Res\text{-}Lin}_B$ refutation imply the exponential lower bound on the number of lines in the $\mathsf{Res\text{-}Lin}_B$ refutation? Do we necessarily need large coefficients in some $\mathsf{Res\text{-}Lin}_B$ refutations with a small number of lines? Or if there is a $\mathsf{Res\text{-}Lin}_B$ refutation with a small number of lines then there is a $\mathsf{Res\text{-}Lin}_B$ refutation with a small number of lines and small coefficients?

References

1 Yaroslav Alekseev, Dima Grigoriev, Edward A. Hirsch, and Iddo Tzameret. Semi-algebraic proofs, IPS lower bounds and the τ-conjecture: Can a natural number be negative? In *Proceedings of the 52th Annual ACM Symposium on Theory of Computing (STOC 2020)*, pages 54–67, 2020.

2 Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc. (3)*, 73(1):1–26, 1996. doi:10.1112/plms/s3-73.1.1.

3 Sam Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *Journal of Computer and System Sciences*, 62(2):267–289, 2001. doi:10.1006/jcss.2000.1726.

4 Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiří Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computational Complexity*, 6(3):256–298, 1996. doi:10.1007/BF01294258.

5 V. Chvátal, W. Cook, and M. Hartmann. On cutting-plane proofs in combinatorial optimization. *Linear Algebra and its Applications*, 114-115:455–499, 1989. Special Issue Dedicated to Alan J. Hoffman. doi:10.1016/0024-3795(89)90476-X.

6 Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 174–183, New York, 1996. ACM.

7 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. doi:10.2307/2273702.

8 W. Cook, C. R. Coullard, and G. Turan. On the complexity of cutting plane proofs. *Discrete Applied Mathematics*, 18:25–38, 1987.

9 Dima Grigoriev and Edward A. Hirsch. Algebraic proof systems over formulas. *Theoret. Comput. Sci.*, 303(1):83–102, 2003. Logic and complexity in computer science (Créteil, 2001).

10 Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018. doi:10.1145/3230742.

11 Armin Haken. The intractability of resolution. *Theoret. Comput. Sci.*, 39(2-3):297–308, 1985.

12 Johan Håstad. On small-depth Frege proofs for tseitin for grids. *J. ACM*, 68(1), 2020. doi:10.1145/3425606.

13 Russell Impagliazzo, Sasank Mouli, and Toniann Pitassi. The surprising power of constant depth algebraic proofs. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 591–603, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394754.

14 Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999. doi:10.1007/s000370050024.

15 Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Annals of Pure and Applied Logic*, 171(1):102722, 2020. doi:10.1016/j.apal.2019.102722.

16 Jan Krajíček. Discretely ordered modules as a first-order extension of the cutting planes proof system. *The Journal of Symbolic Logic*, 63(4):1582–1596, 1998.

17 Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *Journal of Symbolic Logic*, 54(3):1063–1079, 1989. doi:10.2307/2274765.

18 Fedor Part and Iddo Tzameret. Resolution with counting: Different moduli and dag-like lower bounds. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2020, January, 2020, Seattle, WA, USA*, 2020.

19 Toniann Pitassi. Algebraic propositional proof systems. In *Descriptive complexity and finite models (Princeton, NJ, 1996)*, volume 31 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 215–244. Amer. Math. Soc., Providence, RI, 1997.

20 Toniann Pitassi. Unsolvable systems of equations and proof complexity. In *Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998)*, pages 451–460, 1998.

21 Toniann Pitassi, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. Poly-logarithmic Frege depth lower bounds via an expander switching lemma. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 644–657, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897637.

22 Ran Raz and Iddo Tzameret. Resolution over linear equations and multilinear proofs. *Ann. Pure Appl. Logic*, 155(3):194–224, 2008. doi:10.1016/j.apal.2008.04.001.

23 Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987. doi:10.1007/BF01137685.

24 Alexander A. Razborov. Lower bounds for the polynomial calculus. *Comput. Complexity*, 7(4):291–324, 1998.

25 Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987)*, pages 77–82, 1987. doi:10.1145/28395.28404.

26 Dmitry Sokolov. (semi)algebraic proofs over {± 1} variables. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 78–90, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384288.

27 Grigori Tseitin. *On the complexity of derivations in propositional calculus*, pages 466–483. Studies in constructive mathematics and mathematical logic Part II. Consultants Bureau, New-York-London, 1968.

A Proof of the Claim 12

▶ **Claim 12.** *There is a* PC$^{\surd}_{\mathbb{Q}}$ *refutation* $\{R'_1,\dots,R'_s\}$ *of the set* $\Gamma'' = \Gamma \cup \{y'_1 - Q'_1(\vec{x}), \dots y'_m - Q'_m(\vec{x}, y'_1, \dots, y'_{m-1})\}$ *for which the following properties holds:*

- *For every polynomial* $R'_i(\vec{x}, y'_1, \dots, y'_m)$ *one of the following equations holds:* $R'_i(\vec{x}, y_1 \cdot T_1, \dots, y_m \cdot T_m) = R_j(\vec{x}, y_1, \dots, y_m)$ *for some* j *or* $R'_i(\vec{x}, y_1 \cdot T_1, \dots, y_m \cdot T_m) = T_k \cdot R_j(\vec{x}, y_1, \dots, y_m)$ *for some* k *and* j*.*
- *If* $R'_i(\vec{x}, y'_1, \dots, y'_m)$ *was derived from* $R'_j(\vec{x}, y'_1, \dots, y'_m)$ *and* $R'_k(\vec{x}, y_1, \dots, y_m)$ *by taking linear combination with rational constants* α *and* β *(which means that* $R'_i = \alpha R'_j + \beta R'_k$*), then* $\alpha = \frac{1}{T_f}$ *and* $\beta = 0$ *for some* f *or there is some polynomial* $R_h(\vec{x}, y'_1, \dots, y'_m)$ *which was derived from some polynomials* R_k *and* R_l *by using linear combination with constants* α *and* β*.*

Proof of claim. We will construct PC$^{\surd}_{\mathbb{Q}}$ refutation $\{R'_1, R'_2, \dots, R'_s\}$ of the set Γ'' by induction.

Induction statement. Let $\{R_1, \dots, R_i\}$ be a $\mathsf{PC}^{\surd}_{\mathbb{Q}}$ derivation from Γ'. Then there exists a $\mathsf{PC}^{\surd}_{\mathbb{Q}}$ derivation $\{R'_1, \dots, R'_p\}$ from Γ'' such that

- $p \leq 2i$.
- For every $R_j(x_1, \dots, x_n, y_1, \dots, y_m)$ there exists some $R'_k(x_1, \dots, x_n, y'_1, \dots, y'_m)$ such that

$$R'_k(x_1, \dots, x_n, T_1 \cdot y_1, \dots, T_m \cdot y_m) = R_j(x_1, \dots, x_n, y_1, \dots, y_m).$$

- All the properties mentioned in the claim are true for our derivation $\{R'_1, \dots, R'_p\}$.

Base case: If $i = 1$ then $R_i \in \Gamma'$. If $R_i \in \Gamma$ then we can take $R'_1 = R_1$. Otherwise, if $R_i = y_j - Q_j(\vec{x})$ then we can take $R'_1 = y'_j - Q'_j(\vec{x}, y'_1, \dots, y'_{j-1})$ and $R'_2 = \frac{y'_j - Q'_j(\vec{x}, y'_1, \dots, y'_{j-1})}{T_j}$. Then it's obvious that

$$R'_2(\vec{x}, T_1 \cdot y_1, \dots, T_m \cdot y_m) = R_1(\vec{x}, y_1, \dots, y_m).$$

Induction step: Suppose we have already constructed the $\mathsf{PC}^{\surd}_{\mathbb{Q}}$ refutation $\{R'_1, R'_2, \dots, R'_p\}$ for which the induction statement is true. Now we have five cases depending on the way the R_{i+1} is derived.

Case 1: If $R_{i+1} \in \Gamma'$ then this case is equivalent to the base case of induction.

Case 2: If $R_{i+1} = \alpha R_j + \beta R_l$ then $R'_{p+1} = \alpha R'_{j'} + \beta R'_{l'}$ where $R'_{j'}(x_1, \dots, x_n, T_1 \cdot y_1, \dots, T_m \cdot y_m) = R_j(x_1, \dots, x_n, y_1, \dots, y_m)$ and $R'_{l'}(x_1, \dots, x_n, T_1 \cdot y_1, \dots, T_m \cdot y_m) = R_l(x_1, \dots, x_n, y_1, \dots, y_m)$.

Case 3: If $R_{i+1} = x_l \cdot R_j$ then $R'_{p+1} = x_l \cdot R'_{j'}$ where $R'_{j'}(x_1, \dots, x_n, T_1 \cdot y_1, \dots, T_m \cdot y_m) = R_j(x_1, \dots, x_n, y_1, \dots, y_m)$.

Case 4: If $R^2_{i+1} = R_j$ then we take

$$R'_{p+1}(x_1, \dots, x_n, y'_1, \dots, y'_m) = R_{i+1}(x_1, \dots, x_n, \frac{y'_1}{T_1}, \dots, \frac{y'_m}{T_m})$$

By the induction statement we know that

$$R_j(x_1, \dots, x_n, y_1, \dots, y_m) = R'_{j'}(x_1, \dots, x_n, T_1 \cdot y'_1, \dots, T_m \cdot y'_m)$$

for some $R'_{j'}$. Thus we know that

$$R_j(x_1, \dots, x_n, \frac{y'_1}{T_1}, \dots, \frac{y'_m}{T_m}) = R'_{j'}(x_1, \dots, x_n, y'_1, \dots, y'_m).$$

So we know that

$$R'_{p+1}(x_1, \dots, x_n, y'_1, \dots, y'_m)^2 = R_{i+1}(x_1, \dots, x_n, \frac{y'_1}{T_1}, \dots, \frac{y'_m}{T_m})^2 =$$
$$= R_j(x_1, \dots, x_n, \frac{y'_1}{T_1}, \dots, \frac{y'_m}{T_m}) = R'_{j'}(x_1, \dots, x_n, y'_1, \dots, y'_m)$$

and R'_{p+1} is derived from $R'_{j'}$.

Case 5: If $R_{i+1} = y_l \cdot R_j$ then we take $R'_{p+1} = y'_l \cdot R'_{j'}$ and $R'_{p+2} = \frac{R'_{p+1}}{T_l}$ where $R'_{j'}(x_1, \dots, x_n, T_1 \cdot y_1, \dots, T_m \cdot y_m) = R_j(x_1, \dots, x_n, y_1, \dots, y_m)$.

It's easy to see that in all these cases the induction statement stays true. ◁

B Induction form the Theorem 11

Induction statement. Let $\{R'_1, \dots, R'_i\}$ be a $\mathsf{PC}^{\surd}_{\mathbb{Q}}$ derivation from Γ'' with the properties from the Claim 12. Then there exists a $\mathsf{PC}^{\surd}_{\mathbb{Z}}$ derivation $\{R''_1, \dots, R''_f\}$ from Γ'' such that

- $f \le 2i^2$.
- There is some constant $F_i = M_1^{b_1} \cdot M_2^{b_2} \cdots M_m^{b_m} \cdot \delta_1^{b_{m+1}} \cdots \delta_l^{b_{m+l}} \cdot L_1^{b_{m+l+1}} \cdots L_t^{b_{m+l+t}} \in \mathbb{N}$ such that

$$F_i \cdot R'_1 = R''_{f-i+1},\ F_i \cdot R'_2 = R''_{f-i+2},\ \dots,\ F_i \cdot R'_i = R''_f$$

Base case: If $i = 1$ then $R'_i \in \Gamma''$. Then we can take $R''_1 = R'_i$.

Induction step: Suppose we have already constructed the $\mathsf{PC}^{\surd}_{\mathbb{Z}}$ refutation $\{R''_1, R''_2, \dots, R''_f\}$ for which the induction statement is true. Then there are four cases depending on the way the R'_{i+1} is derived.

Case 1: If $R'_{i+1} \in \Gamma''$ then $F_{i+1} = F_i$ and

$$R''_{f+1} = R'_{i+1},\ R''_{f+2} = F_{i+1} \cdot R'_1,\ R''_{f+3} = F_{i+1} \cdot R'_2,\ \dots,\\ R''_{f+i+1} = F_{i+1} \cdot R'_i,\ ,R''_{f+i+2} = F_{i+1} \cdot R'_{i+1}$$

Case 2: If $R'_{i+1} = x_j R'_l$ or $R'_{i+1} = y'_j R'_l$ then $F_{i+1} = F_i$,

$$R''_{f+1} = F_{i+1} \cdot R'_1,\ R''_{f+2} = F_{i+1} \cdot R'_2,\ \dots,\ R''_{f+i} = F_{i+1} \cdot R'_i$$

and $R''_{f+i+1} = x_j R''_{f-i+l} = F_{i+1} \cdot R'_{i+1}$ or $R''_{f+i+1} = y_j R''_{f-i+l} = F_{i+1} \cdot R''_{i+1}$.

Case 3: If $R_{i+1} = \alpha R_j + \beta R_k$ where $\alpha = \frac{p_1}{q_1}$ and $\beta = \frac{p_2}{q_2}$ where $\{p_1, q_1, p_2, q_2\} \subset \mathbb{Z}$. Then we can take $F_{i+1} = q_1 q_2 F_i$,

$$R''_{f+1} = q_1 q_2 \cdot R''_{f-i+1} = F_{i+1} \cdot R'_1,\ R''_{f+2} = q_1 q_2 \cdot R''_{f-i+2} = F_{i+1} \cdot R'_2,\ \dots,\\ R''_{f+i} = q_1 q_2 \cdot R''_f = F_{i+1} R'_i$$

and $R''_{f+i+1} = p_1 q_2 \cdot R''_{f-i+j} + p_2 q_1 \cdot R''_{f-i+k} = M_{i+1} R'_{i+1}$. From the Claim 12 we know that $\alpha = \frac{1}{T_k}$ for some k and $\beta = 0$, or q_2 and q_1 are equal to some δ_k and δ_r. From the induction statement we know that

$$F_i = M_1^{b_1} \cdot M_2^{b_2} \cdots M_m^{b_m} \cdot \delta_1^{b_{m+1}} \cdots \delta_l^{b_{m+l}} \cdot L_1^{b_{m+l+1}} \cdots L_t^{b_{m+l+t}}.$$

Then, since $T_k = M_1^{r_{1k}} \cdots M_m^{r_{mk}}$, we know that

$$F_{i+1} = M_1^{b'_1} \cdot M_2^{b'_2} \cdots M_m^{b'_m} \cdot \delta_1^{b'_{m+1}} \cdots \delta_l^{b'_{m+l}} \cdot L_1^{b'_{m+l+1}} \cdots L_t^{b'_{m+l+t}},$$

and the induction statement stays true.

Case 4: Suppose $R'^2_{i+1} = R'_j$. We know that

$$R'_{i+1}(x_1, \dots, x_n, y'_1, \dots, y'_m) = R_k(x_1, \dots, x_n, \frac{y'_1}{T_1}, \dots, \frac{y'_m}{T_m})$$

or

$$R'_{i+1}(x_1, \dots, x_n, y'_1, \dots, y'_m) = T_h \cdot R_k(x_1, \dots, x_n, \frac{y'_1}{T_1}, \dots, \frac{y'_m}{T_m})$$

for some h. Then we can take $M' = L_k \cdot T_1^{\alpha_1} \cdot T_2^{\alpha_2} \cdots T_m^{\alpha_m} = L_k \cdot M_1^{\alpha'_1} \cdot M_2^{\alpha'_2} \cdots M_m^{\alpha'_m}$ for some non-negative integers $\alpha_1, \dots, \alpha_m$, such that $M' \cdot R'_{i+1}$ is an integer polynomial. We know that such integers $\alpha_1, \dots, \alpha_m$ exist since L_k is the product of all denominators of coefficients of polynomial R_k.

Then we can take $F_{i+1} = M' \cdot F_i$. It's obvious that $F_{i+1} \cdot R'_{i+1}$ is an integer polynomial. Then we can make the following $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ derivation:

$$R''_{f+1} = F_i (M')^2 \cdot R''_{f-i+j} = (F_i M')^2 \cdot R'_j,\ R'_{f+2} = M' \cdot R'_{f-i+1} = F_{i+1} \cdot R_1,$$
$$R'_{f+3} = M' \cdot R'_{f-i+2} = F_{i+1} \cdot R_2,\ \dots,\ R'_{f+i+1} = M' \cdot R'_f = F_{i+1} R_i.$$

Then we can take $R''_{f+i+2} = F_i M' \cdot R'_{i+1}$ and since $R''_{f+1} = (F_i M')^2 \cdot R'_j$ we know that $(R''_{f+i+2})^2 = R''_{f+1}$ and we get a correct $\mathsf{PC}_{\mathbb{Z}}^{\surd}$ derivation.

Since $M' = L_p \cdot M_1^{\alpha'_1} \cdot M_2^{\alpha'_2} \cdots M_m^{\alpha'_m}$ we know that

$$F_{i+1} = M_1^{b'_1} \cdot M_2^{b'_2} \cdots M_m^{b'_m} \cdot \delta_1^{b'_{m+1}} \cdots \delta_f^{b'_{m+l}} \cdot L_1^{b'_{m+l+1}} \cdots L_t^{b'_{m+l+t}},$$

and the induction statement stays true.

C Proof of the Theorem 17

▶ **Theorem 17.** *Let $\pi = (D_1, \dots, D_l)$ be an* $\mathsf{Res\text{-}Lin}_B$ *proof sequence of D_l from some collection of initial disjunctions of linear equations $Q_1, \dots, Q_m$. Also consider $L_1, \dots, L_t$ – all affine forms that we have in all disjunctions in our* $\mathsf{Res\text{-}Lin}_B$ *proof sequence.*

Then, there exists a $\mathsf{PC}_{\mathbb{Q}}^{\surd}$ *proof of $\widehat{D}_l$ from $\widehat{Q}_1 \cup \ldots \cup \widehat{Q}_m \cup \{y_1 = L_1, y_2 = L_2, \dots, y_t = L_t\}$ of size at most $O(p(Size(\pi)))$ for some polynomial p.*

Proof. We proceed by induction on the number of lines in π.
Base case: An $\mathsf{Res\text{-}Lin}_B$ axiom Q_i is translated into $\widehat{Q_i}$ and $\mathsf{Res\text{-}Lin}_B$ Boolean axiom $(x_i = 0) \vee (x_i = 1)$ is translated into PC axiom $x_i^2 - x_i = 0$.
Induction step: Now we will simulate all $\mathsf{Res\text{-}Lin}_B$ derivation rules in the $\mathsf{PC}_{\mathbb{Q}}^{\surd}$ proof.

- **Resolution**: Assume that $D_i = A \vee B \vee (\alpha L_1 + \beta L_2)$ where $D_j = A \vee L_1$ and $D_k = B \vee L_2$. Then, we have already derived polynomial equations

$$y_{j1} = (a_{j1}^{(1)} x_1 + \ldots + a_{jn}^{(1)} x_n - a_{j0}^{(1)}),\ \dots,\ y_{jt_j} = (a_{j1}^{(t_j)} x_1 + \ldots + a_{jn}^{(t_j)} x_n - a_{j0}^{(t_j)}),$$
$$y_{k1} = (a_{k1}^{(1)} x_1 + \ldots + a_{kn}^{(1)} x_n - a_{k0}^{(1)}),\ \dots,\ y_{kt_k} = (a_{k1}^{(t_k)} x_1 + \ldots + a_{kn}^{(t_k)} x_n - a_{k0}^{(t_k)}),$$
$$y_{j1} \cdot y_{j2} \cdots y_{jt_j} = 0,\ y_{k1} \cdot y_{k2} \cdots y_{kt_k} = 0$$

where

$$A = (a_{j1}^{(2)} x_1 + \ldots + a_{jn}^{(2)} x_n = a_{j0}^{(2)}) \vee \cdots \vee (a_{j1}^{(t_j)} x_1 + \ldots + a_{jn}^{(t_j)} x_n = a_{j0}^{(t_j)}),$$
$$B = (a_{k1}^{(2)} x_1 + \ldots + a_{kn}^{(2)} x_n = a_{k0}^{(2)}) \vee \cdots \vee (a_{k1}^{(t_k)} x_1 + \ldots + a_{kn}^{(t_k)} x_n = a_{k0}^{(t_k)})$$
$$L_1 = (a_{j1}^{(1)} x_1 + \ldots + a_{jn}^{(1)} x_n = a_{j0}^{(1)}),\ L_2 = (a_{k1}^{(1)} x_1 + \ldots + a_{kn}^{(1)} x_n = a_{k0}^{(1)}).$$

Then we can derive $y_{j1} \cdot y_{j2} \cdots y_{jt_j} \cdot y_{k2} \cdots y_{kt_k} = 0$, $y_{k1} \cdot y_{j2} \cdots y_{jt_j} \cdot y_{k2} \cdots y_{kt_k} = 0$ and thus $(\alpha y_{j1} + \beta y_{k1}) \cdot y_{j2} \cdots y_{jt_j} \cdot y_{k2} \cdots y_{kt_k} = 0$. Then there is some equation $y_i = L_i$ from the set $\{y_1 = L_1, y_2 = L_2, \dots, y_t = L_t\}$, for which holds

$$L_i = \alpha(a_{j1}^{(1)} x_1 + \ldots + a_{jn}^{(1)} x_n - a_{j0}^{(1)}) + \beta(a_{k1}^{(1)} x_1 + \ldots + a_{kn}^{(1)} x_n - a_{k0}^{(1)}).$$

Then we can derive $y_i = \alpha y_{j1} + \beta y_{k1}$ and $y_i \cdot y_{j2} \cdots y_{jt_j} \cdot y_{k2} \cdots y_{kt_k} = 0$ which is part of $\widehat{D}_i$.

- **Weakening**: Assume that $D_i = D_j \vee L$ where L is a linear equation. Then, we have already derived polynomial equations

  $$y_{j1} = (a_{j1}^{(1)}x_1 + \ldots + a_{jn}^{(1)}x_n - a_{j0}^{(1)}),\ \ldots,\ y_{jt_j} = (a_{j1}^{(t_j)}x_1 + \ldots + a_{jn}^{(t_j)}x_n - a_{j0}^{(t_j)}),$$
  $$y_{j1} \cdot y_{j2} \cdots y_{jt_j} = 0.$$

  We know that there is some variable y_0 for which $y_0 = b_1x_1 + \ldots b_nx_n - b_0$ where L is a linear equation $b_1x_1 + \ldots b_nx_n = b_0$. From $y_{j1} \cdot y_{j2} \cdots y_{jt_j} = 0$ we can derive $y_0 \cdot y_{j1} \cdot y_{j2} \cdots y_{jt_j} = 0$ which is part of $\widehat{D}_i$.
- **Simplification**: Suppose that $D_i = A$ and $D_j = A \vee (k = 0)$ where $k \in \mathbb{Z}$, $k \neq 0$. Then, we have already derived polynomial equations

  $$y_{j1} = (a_{j1}^{(1)}x_1 + \ldots + a_{jn}^{(1)}x_n - a_{j0}^{(1)}),\ \ldots,$$
  $$y_{jt_j-1} = (a_{j1}^{(t_j-1)}x_1 + \ldots + a_{jn}^{(t_j-1)}x_n - a_{j0}^{(t_j-1)}),\ y_{jt_j} = k,$$
  $$y_{j1} \cdot y_{j2} \cdots y_{jt_j} = 0.$$

  From equation $y_{j1} \cdot y_{j2} \cdots y_{jt_j} = 0$ we can derive equation $y_{j1} \cdot y_{j2} \cdots y_{jt_j-1} \cdot k = 0$ from which we can derive $y_{j1} \cdot y_{j2} \cdots y_{jt_j-1} = 0$ which is part of $\widehat{D}_i$.
- **Contraction**: Assume that $D_i = A \vee L$ and $D_j \vee L \vee L$ where L is a linear equation. Then, we have already derived polynomial equations

  $$y_{j1} = (a_{j1}^{(1)}x_1 + \ldots + a_{jn}^{(1)}x_n - a_{j0}^{(1)}),\ \ldots,\ y_{jt_j-1} = y_{jt_j} = (a_{j1}^{(t_j)}x_1 + \ldots + a_{jn}^{(t_j)}x_n - a_{j0}^{(t_j)}),$$
  $$y_{j1} \cdot y_{j2} \cdots y_{jt_j-1} \cdot y_{jt_j} = 0.$$

  Then we can derive $y_{jt_j-1} = y_{jt_j}$ and $y_{j1} \cdot y_{j2} \cdots y_{jt_j-2} \cdot (y_{jt_j-1}^2) = 0$. Using multiplication we can derive $y_{j1}^2 \cdot y_{j2}^2 \cdots y_{jt_j-2}^2 \cdot (y_{jt_j-1}^2) = 0$ from which we can derive the equation $y_{j1} \cdot y_{j2} \cdots y_{jt_j-1} = 0$ by using the square root rule. This equation is the last part of $\widehat{D}_i$ because other parts were derived earlier. ◀

D Proof of the theorem 22

▶ **Theorem 22.** *Any* Ext-PC$_{\mathbb{Q}}$*-derivation of* $1 + x_1 + \ldots + 2^{n-1}x_n = 0$ *from equation* $(1 + x_1 + \ldots + 2^{n-1}x_n)^2 = 0$ *requires size* $2^{\Omega(n)}$.

Proof. Firstly, we need the following claim:

▷ Claim. For any Ext-PC$_{\mathbb{Z}}$-derivation of $M \cdot (1 + x_1 + \ldots + 2^{n-1}x_n) = 0$ from equation $(1 + x_1 + \ldots + 2^{n-1}x_n)^2 = 0$ where $M \in \mathbb{Z}, M \neq 0$, constant M is divisible by every prime number less than 2^n.

Proof of claim. Assume that $\{R_1, \ldots, R_t\}$ is an Ext-PC$_{\mathbb{Z}}$-derivation of $M \cdot (1 + x_1 + \ldots + 2^{n-1}x_n) = 0$ from equation $(1 + x_1 + \ldots + 2^{n-1}x_n)^2 = 0$. Then we know that $\{R_1, \ldots, R_t\}$ is a PC$_{\mathbb{Z}}$ refutation of some set

$$\Gamma' = \{G(\vec{x}), F_1(\vec{x}), \ldots, F_n(\vec{x}), y_1 - Q_1(\vec{x}), \ldots y_m - Q_m(\vec{x}, y_1, \ldots, y_{m-1})\}$$

where $G(\vec{x}) = (1 + \sum_{i=1}^{i=n} 2^{(i-1)}x_i)^2$, $F_i(\vec{x}) = x_i^2 - x_i$, $Q_i \in \mathbb{Z}[\vec{x}, y_1, \ldots, y_{i-1}]$ and $R_t = M \cdot (1 + x_1 + \ldots + 2^{n-1}x_n)$.

Now consider arbitrary integer number $0 \leq k < 2^n$ and its binary representation $b_1, \ldots, b_n$. Then $G(b_1, \ldots, b_n) = (k+1)^2$, $F_i(b_1, \ldots, b_n) = b_i^2 - b_i = 0$. Also consider integers $c_1, \ldots, c_m$ such that $c_i = Q_i(b_1, \ldots, b_n, c_1, c_2, \ldots, c_{i-1})$. Now we will prove by induction that every integer number $R_i(b_1, \ldots, b_n, c_1, \ldots, c_m)$ is divisible by $(k+1)^2$ and thus M is divisible by every prime number less than 2^n since $1 + b_1 + \ldots + 2^{n-1}b_n = k + 1$.

Base case: if $i = 1$, then $R_i = G(b_1, \dots, b_n, c_1, \dots, c_m) = (k+1)^2$ or $R_i = F_i(b_1, \dots, b_n, c_1, \dots, c_m) = 0$ or $R_i(b_1, \dots, b_n, c_1, \dots, c_m) = c_i - Q_i(b_1, \dots, b_n, c_1, \dots, c_{i-1}) = 0$ which means that R_i is divisible by $(k+1)^2$.

Induction step: suppose we know that R_j is divisible by $(k+1)^2$ for any $j \leq i$. Now we will show it for R_{i+1}. There are three cases:

1. If $R_{i+1} \in \Gamma'$, then this case is equivalent to the base case and $R_{i+1}(b_1, \dots, b_n, c_1, \dots, c_m)$ is divisible by $(k+1)^2$.
2. If $R_{i+1} = \alpha R_j + \beta R_s$ for $\alpha, \beta \in \mathbb{Z}$ and $j, s \leq i$, then $R_{i+1}(b_1, \dots, b_n, c_1, \dots, c_m)$ is divisible by $(k+1)^2$ because $R_j(b_1, \dots, b_n, c_1, \dots, c_m)$ and $R_s(b_1, \dots, b_n, c_1, \dots, c_m)$ are divisible by $(k+1)^2$ and α and β are integers.
3. If $R_{i+1} = x_j R_s$ or $R_{i+1} = y_j R_s$, then $R_{i+1}(b_1, \dots, b_n, c_1, \dots, c_m)$ is divisible by $(k+1)^2$ because $R_s(b_1, \dots, b_n, c_1, \dots, c_m)$ is divisible by $(k+1)^2$ and b_i and c_i are integers.

Since every $R_i(b_1, \dots, b_n, c_1, \dots, c_m)$ is divisible by $(k+1)^2$, we know that $R_t(b_1, \dots, b_n, c_1, \dots, c_m) = M \cdot (k+1)$ is divisible by $(k+1)^2$. Then we know that M is divisible by $k+1$ and thus M is divisible by every prime number less than 2^n.

Now assume that $\{R_1, \dots, R_t\}$ is an Ext-PC$_{\mathbb{Q}}$-derivation from arbitrary set of equations $\Gamma \subset \mathbb{Z}[\vec{x}]$ of the size S. Then we know that $\{R_1, \dots, R_t\}$ is a PC$_{\mathbb{Q}}^{\surd}$ refutation of some set $\Gamma' = \Gamma \cup \{y_1 - Q_1(\vec{x}), \dots, y_m - Q_m(\vec{x}, y_1, \dots, y_{m-1})\}$ where $Q_i \in \mathbb{Q}[\vec{x}, \vec{y}]$. Like in the proof of Theorem 11 we can consider all products of denominators of polynomials Q_i, R_i and all denominators in linear combination rule. Let's denote those constants as T_i. We know that $\prod T_i \leq 2^{\Omega(S)}$. From the proof of Theorem 11 we know that there is an Ext-PC$_{\mathbb{Z}}$-derivation $\{R'_1, \dots, R'_f\}$ from the set Γ for which $R'_f = T_1^{\alpha_1} \cdots T_r^{\alpha_r} R_t$ where $\alpha_i \in \mathbb{N}$.

Then we can consider

$$\Gamma = \{(1 + x_1 + \dots + 2^{n-1}x_n)^2 = 0, x_1^2 - x_1 = 0, \dots, x_n^2 - x_n = 0\}$$

and $R_t = 1 + x_1 + \dots + 2^{n-1}x_n$. Then we know that for every Ext-PC$_{\mathbb{Q}}$-derivation of $1 + x_1 + \dots + 2^{n-1}x_n = 0$ from equation $(1 + x_1 + \dots + 2^{n-1}x_n)^2 = 0$ of size S there is an Ext-PC$_{\mathbb{Z}}$-derivation of $M \cdot (1 + x_1 + \dots + 2^{n-1}x_n) = 0$ from equation $(1 + x_1 + \dots + 2^{n-1}x_n)^2 = 0$ where $M = T_1^{\alpha_1} \cdots T_r^{\alpha_r}$ and $T_1 \cdots T_r \leq 2^{\Omega(S)}$. However, from previous claim we know that M is divisible by all prime numbers less than 2^n. Then $T_1^{\alpha_1} \cdots T_r^{\alpha_r}$ is divisible by all prime numbers less than 2^n which means that $T_1 \cdots T_r$ is divisible by all prime numbers less than 2^n. Then $2^{2^{\Omega(n)}} \leq T_1 \cdots T_r \leq 2^{\Omega(S)}$ which means that $S \geq 2^{\Omega(n)}$. ◁

◀

Error Reduction for Weighted PRGs Against Read Once Branching Programs

Gil Cohen ✉
School of Computer Science, Tel Aviv University, Israel

Dean Doron ✉
Department of Computer Science, Stanford University, CA, USA

Oren Renard ✉
School of Computer Science, Tel Aviv University, Israel

Ori Sberlo ✉
School of Computer Science, Tel Aviv University, Israel

Amnon Ta-Shma ✉
School of Computer Science, Tel Aviv University, Israel

Abstract

Weighted pseudorandom generators (WPRGs), introduced by Braverman, Cohen and Garg [5], are a generalization of pseudorandom generators (PRGs) in which arbitrary real weights are considered, rather than a probability mass. Braverman et al. constructed WPRGs against read once branching programs (ROBPs) with near-optimal dependence on the error parameter. Chattopadhyay and Liao [6] somewhat simplified the technically involved BCG construction, also obtaining some improvement in parameters.

In this work we devise an error reduction procedure for PRGs against ROBPs. More precisely, our procedure transforms any PRG against length n width w ROBP with error $1/\mathrm{poly}(n)$ having seed length s to a WPRG with seed length $s + O(\log \frac{w}{\varepsilon} \cdot \log\log \frac{1}{\varepsilon})$. By instantiating our procedure with Nisan's PRG [17] we obtain a WPRG with seed length $O(\log n \cdot \log(nw) + \log \frac{w}{\varepsilon} \cdot \log\log \frac{1}{\varepsilon})$. This improves upon [5] and is incomparable with [6].

Our construction is significantly simpler on the technical side and is conceptually cleaner. Another advantage of our construction is its low space complexity $O(\log nw) + \mathrm{poly}(\log\log \frac{1}{\varepsilon})$ which is logarithmic in n for interesting values of the error parameter ε. Previous constructions (like [5, 6]) specify the seed length but not the space complexity, though it is plausible they can also achieve such (or close) space complexity.

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Pseudorandom generators, Read once branching programs, Space-bounded computation

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.22

Funding *Gil Cohen*: Supported by ERC starting grant 949499 and by Israel Science Foundation grant 1569/18.
Dean Doron: Supported by NSF award CCF-1763311.
Oren Renard: Supported by the Azrieli Faculty Fellowship.
Ori Sberlo: Supported by ERC starting grant 949499 and by ISF grant 952/18.
Amnon Ta-Shma: Supported by Israel Science Foundation grant 952/18.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 22; pp. 22:1–22:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 A brief account of space-bounded derandomization

Understanding the role that randomness plays in computation is of central importance in complexity theory. While randomness is provably necessary in many computational settings such as cryptography, PCPs and distributed computing, it is widely believed that randomness adds no significant computational power to neither time- nor space-bounded algorithms. Remarkably, proving such a statement for time-bounded algorithms implies circuit lower bounds which seem to be out of reach of current proof techniques [19, 14, 16].

On the other hand, there is no known barrier for proving such a statement in the space-bounded setting. Indeed, while we cannot even rule out a scenario in which randomness "buys" exponential time, the space-bounded setting is much better understood. Savitch's theorem [23] already implies that any one-sided error randomized algorithm can be simulated deterministically with only a quadratic overhead in space, namely $\mathbf{RL} \subseteq \mathbf{L}^2$. The (possibly) stronger inclusion $\mathbf{BPL} \subseteq \mathbf{L}^2$ can be proven easily through a variant of Savitch's theorem and also follows from [4]. Using pseudorandom generators, Nisan [17, 18] devised a time-efficient derandomization with quadratic overhead in space, concretely, $\mathbf{BPL} \subseteq \mathbf{DTISP}(\text{poly}(n), \log^2 n)$. Focusing solely on space, the state of the art result was obtained by Saks and Zhou [22] that build on Nisan's work to deterministically simulate two-sided error space s randomized algorithms in space $O(s^{3/2})$, thus, establishing that $\mathbf{BPL} \subseteq \mathbf{L}^{3/2}$.

1.2 Pseudorandom generators for ROBPs

Space-bounded algorithms are typically studied by considering their non-uniform counterparts. A length n, width w *read-once branching program* (ROBP) is a directed graph whose nodes, called states, are partitioned to $n+1$ layers, each consists of at most w states. The first layer contains a designated "start" state, and the last layer consists of two states labeled 'accept' and 'reject'. From every state but for the latter two, there are two outgoing edges, labeled by 0 and 1, to the following layer.[1] On input $x \in \{0,1\}^n$, the computation proceeds by following the edges according to the labels given by the bits of x starting from the start state. The string x is accepted by the program if the computation ends in the accept state.

A well-known fact (see, e.g., [10, Chapter 5], and [3, Chapter 14.4.4]) is that any space s randomized algorithm in the Turing model can be simulated by a length n, width w ROBP with $n, w = 2^{O(s)}$. Thus, one approach to derandomize two-sided error space-bounded algorithms is to construct, in bounded space, a distribution of small support that "looks random" to any such ROBP. We say that a distribution $\mathcal{D}$ on n-bit strings is (n, w, ε) *pseudorandom* if for every length n, width w ROBP, the path induced by an instruction sequence that is sampled from $\mathcal{D}$ has, up to an additive error ε, the same probability to end in the accept state as a truly random path. A truly random path corresponds to a path picked uniformly at random from the 2^n possible paths. An (n, w, ε) *pseudorandom generator* (PRG) is an algorithm $\mathsf{PRG}\colon \{0,1\}^s \to \{0,1\}^n$ that when fed with s uniformly random bits has an output distribution that is (n, w, ε) pseudorandom. We refer to the input to PRG as the *seed*.

[1] For simplicity, here we only consider ROBPs with two outgoing edges. Larger out-degrees (or alphabet) can also be considered and is in fact crucial for obtaining our result even if one is only interested in the binary case.

Derandomizing using a PRG is straightforward. By iterating over all seeds and generating the corresponding instruction sequences, one can calculate the fraction of those paths that end in the accept state. This way, one obtains an ε-approximation to the probability of reaching the accept state while taking a truly random path in the program. The space overhead consists of the seed length s (as an iterator is maintained) and the space of the PRG.

One can prove the existence of an (n, w, ε) PRG with seed length $O(\log(nw/\varepsilon))$. The proof is via the probabilistic method and has no guarantee on the space complexity of the PRG. As such, it is not useful for the purpose of derandomization. In his seminal work, Nisan [17] devised a PRG with seed length $s = O(\log n \cdot \log(nw/\varepsilon))$ and space complexity $O(\log(nw/\varepsilon))$. Setting $n, w = 2^{\Theta(s)}$ and ε to a small constant, the seed length is $O(s^2)$ indeed yields derandomization with quadratic overhead in space. Saks and Zhou [22] applied Nisan's generator in a far more sophisticated way than the naïve derandomization, in particular exploiting its low space complexity, so to obtain their result.

1.3 Pseudorandom pseudo-distributions for ROBPs

Braverman et al. [5] introduced the notion of a *pseudorandom pseudo-distribution* (PRPD) generalizing pseudorandom distributions.

▶ **Definition 1** (pseudorandom pseudo-distribution). *Let $\rho_1, \ldots, \rho_{2^s} \in \mathbb{R}$ and $p_1, \ldots, p_{2^s} \in \{0,1\}^n$. The sequence $\widetilde{\mathcal{D}} = ((\rho_1, p_1), \ldots, (\rho_{2^s}, p_{2^s}))$ is an (n, w, ε)* pseudorandom pseudo-distribution *(PRPD) if for every length n, width w ROBP, the sum of all ρ_i-s for which the respective paths p_i end in the accept state is an ε-approximation to the probability of ending at the accept state by taking a truly random path in the program.*

Note that Definition 1 allows the weights ρ_i to take both positive and negative values. These values are not necessarily bounded by 1 in absolute value, nor by any constant for that matter, and they do not necessarily sum up to 1. Nevertheless, the definition requires that the numbers cancel out nicely so that summing the weights of the respective paths that arrive to the accept state yields an ε-approximation for the probability of arriving to the accept state by taking a truly random path (and, in particular, the sum is a number in $[-\varepsilon, 1+\varepsilon]$). Analogous to a PRG, an (n, w, ε) *weighted pseudorandom generator* (WRPG) is an algorithm $\mathsf{WPRG}\colon \{0,1\}^s \to \mathbb{R} \times \{0,1\}^n$ whose output, when fed with a uniform seed, is an (n, w, ε) PRPD.

A WPRG that can be computed in bounded space suffices to derandomize two-sided error randomized algorithms. Indeed, the straightforward derandomization using a pseudorandom (proper) distribution, which sums the probability mass of the relevant paths, works just as well for pseudo-distributions as one can sum up the weights ρ_i which, in a sense, generalize the probability mass. Of course, the space requirement now depends on the bit complexity of the weights as well.

1.4 The error parameter

Braverman et al. [5] constructed a WPRG that has seed length with an improved–in fact near-optimal–dependence on the error parameter ε. Their WPRG has seed length $O(\log^2 n \cdot \log\log_n \frac{1}{\varepsilon} + \log n \cdot \log w + \log \frac{w}{\varepsilon} \cdot \log\log \frac{w}{\varepsilon})$. For the purpose of derandomization, the error parameter is anyhow taken to be constant, and so the necessity of such an improvement may seem moot. However, by inspecting Nisan's recursive construction one can see that the $\log^2 n$ term in the seed length appears due to the way the error evolves throughout the recursion. Hence, a construction which allows for a more delicate error analysis is called for. Furthermore, the Saks-Zhou construction applies Nisan's PRG in a setting in which $\varepsilon \ll 1/n$ for obtaining their result. It was observed [5] that improving upon [22] can be obtained by constructing a PRG having seed length with better dependence on both w, ε, even when retaining the $\log^2 n$ dependence.

Interestingly (and unfortunately), the $\log^2 n$ term in the BCG construction appears for a completely different reason. In short, unlike prior works [17, 15] that maintain a list of instructions throughout the recursion, BCG maintains a more involved structure consisting of several lists of lists. Maintaining the invariant on this complex structure is the reason for the $\log^2 n$ term in the seed of BCG's construction.

As hinted above, the BCG construction is quite involved. In a subsequent work Chattopadhyay and Liao [6] somewhat simplified the BCG construction also obtaining slight improvement in parameters. In particular, the seed length obtained by [6] is $O(\log n \cdot \log nw \cdot \log\log nw + \log \frac{1}{\varepsilon})$. Additionally, Hoza and Zuckerman [13] obtained a significantly simpler construction of hitting sets against ROBPs. Their construction has seed length $O(\frac{1}{\max(1, \log\log w - \log\log n)} \cdot \log n \cdot \log nw + \log \frac{1}{\varepsilon})$. Although hitting sets are weaker objects than PRPDs that are aimed for the derandomization of one sided error randomized algorithms, a subsequent work by Cheng and Hoza [7] showed how to derandomize two sided error randomized algorithms using hitting sets. While this is an illuminating result, we stress that most known constructions of PRGs, WPRGs and hitting sets make use of compositions (either directly or indirectly) and HSGs do not compose well, and so it is very much desired to devise new techniques for constructing PRGs and WPRGs.

1.5 Our contribution

This work further focuses on the error parameter of PRPDs. As our main result, we obtain an *error reduction procedure*. That is, we devise an algorithm that transforms, in a black-box manner, a PRG with a modest error parameter ε_0 to a WPRG with a desired error parameter ε, having comparable seed length and with a near optimal dependence on ε.

▶ **Theorem 2** (main result, see also Corollary 15). *Suppose* PRG *is an* (n, w, n^{-2}) *PRG with seed length* s_0*, computable in space* m*. Then, for every* ε *there exists an* (n, w, ε) *WPRG with seed length*

$$s = s_0 + O\left(\log \frac{w}{\varepsilon} \cdot \log\log_n \frac{1}{\varepsilon}\right).$$

that is computable in space $O(m + (\log\log \frac{w}{\varepsilon})^3)$*.*

When instantiated with Nisan's PRG [17] our error reduction procedure yields WPRGs with a seed that is slightly shorter than [5] and is incomparable to [6].

▶ **Corollary 3** (see also Corollary 16). *There exists an* (n, w, ε) *WPRG with seed length*

$$O\left(\log n \cdot \log nw + \log \frac{w}{\varepsilon} \cdot \log\log_n \frac{1}{\varepsilon}\right)$$

computable in space $O\left(\log nw + \left(\log\log \frac{w}{\varepsilon}\right)^3\right)$*.*

Our error reduction procedure as well as the resulting WPRG are significantly simpler than [5, 6]. Moreover, the underlying ideas are different and conceptually cleaner. More generally, it is much preferred to have a black-box error reduction procedure rather than a specific explicit construction. On top of the insights obtained, such a modularization has the potential of being instantiated in different settings such as for regular and permutation ROBPs or for bounded-width ROBPs.

Our error reduction procedure borrows ideas from the line of work concerning deterministic space-efficient graph algorithms, in particular a recent work by Ahmadinejad, Kelner, Murtagh, Peebles, Sidford and Vadhan [1] (which, in turn, is based on an exciting line of work on nearly-linear time graph algorithms, deterministic or otherwise. See [9, 8] and references therein).

Independently, Pyne and Vadhan [20] also used the Richardson iteration to obtain a WPRG for polynomial-width branching programs, and furthermore used that to obtain new results for permutation BPs.

1.6 An overview of our construction

Let $\mathsf{PRG}\colon \{0,1\}^s \to \{0,1\}^n$ be an (n,w,ε_0) PRG whose error we wish to reduce. Let $\overline{A} = (A_1,\dots,A_n)$ be the $w \times w$ stochastic matrices that correspond to a length n width w ROBP. That is, $A_i = \frac{1}{2}(A_i^{(0)} + A_i^{(1)})$ where $A_i^{(0)}$ is the Boolean stochastic matrix that encodes the edges leaving layer i that are labeled with 0 and $A_i^{(1)}$ encodes the edges labeled with 1. Define the $(n+1)w \times (n+1)w$ lower triangular block matrix B as follows. For $a,b \in [n+1]$, $a > b$, and $\sigma \in \{0,1\}^s$, let

$$B[a,b] = \mathop{\mathbb{E}}_{\sigma \in \{0,1\}^s} \left[A_a^{(\mathsf{PRG}(\sigma)_{a-b})} \cdots A_b^{(\mathsf{PRG}(\sigma)_1)} \right].$$

Further, $B[a,a] = I_w$. Since PRG has error ε_0, for every block $B[a,b]$ with $a > b$, $\|B[a,b] - A_a \cdots A_b\| \le \varepsilon_0$. Following [1] we observe that by denoting

$$L = \begin{pmatrix} I & 0 & \dots & 0 & 0 \\ -A_1 & I & \dots & 0 & 0 \\ 0 & -A_2 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -A_n & I \end{pmatrix},$$

one has that

$$L^{-1} = \begin{pmatrix} I & 0 & \dots & 0 & 0 \\ A_1 & I & \dots & 0 & 0 \\ A_2A_1 & A_2 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_n \dots A_1 & A_n \dots A_2 & \dots & A_n & I \end{pmatrix}.$$

Thus, $\|B - L^{-1}\| \le (n+1)\varepsilon_0$. That is, the crude error PRG can be used to approximate L^{-1} by applying it to all subprograms of the original ROBP.

Richardson iteration is a method for improving a given approximation to an inverse of a matrix. This method is frequently used to construct a preconditioner to a Laplacian system. To describe this method, let $L = I - A$. For $k \ge 1$ define the matrix

$$R_k = \sum_{i=0}^{k} (I - BL)^i B. \tag{1}$$

It can be shown that $\left\|R_k - L^{-1}\right\| \leq (n+1)\left(2(n+1)\varepsilon_0\right)^{k+1}$. Thus, by taking $\varepsilon_0 = n^{-2}$ and $k = O(\log_n \frac{1}{\varepsilon})$, one obtains approximation $\|R_k - L^{-1}\| \leq \varepsilon$. In particular, the lower left block of R_k is an ε-approximation of the desired product $A_n \cdots A_1$.

We further develop Equation (1). Let $\Delta = I - BL$. One can show that

$$\Delta[a,b] = \begin{cases} B[a,b+1]\cdot A_b - B[a,b] & a > b, \\ 0 & a \leq b. \end{cases} \tag{2}$$

Substituting this back to R_k, for $a > b$ we have that

$$R_k[a,b] \;=\; B[a,b] + \sum_{i=1}^{k} \sum_{a > \ell_i > \cdots > \ell_1 \geq b} \Delta[a,\ell_i]\cdot\Delta[\ell_i,\ell_{i-1}]\cdots\Delta[\ell_2,\ell_1]\cdot B[\ell_1,b].$$

If we further let $C_0[a,b] = B[a,b+1]\cdot A_b$ and $C_1[a,b] = B[a,b]$ then

$$R_k[a,b] = B[a,b] + \sum_{i=1}^{k} \sum_{a > \ell_1 > \cdots > \ell_i \geq b} \sum_{t_1,\ldots,t_i \in \{0,1\}} (-1)^{t_1+\cdots+t_i}\cdot C_{t_i}[a,\ell_i]\cdots C_{t_1}[\ell_2,\ell_1]\cdot B[\ell_1,b]. \tag{3}$$

By extending the definition of ROBPs to arbitrary alphabets (rather than binary) we observe that each summand in Equation (3) can be realized by a ROBP. Our construction thus uses an auxiliary PRG that ε' fools each summand and hence $\varepsilon' n^{O(k)} \approx \varepsilon' \cdot \mathrm{poly}(\frac{1}{\varepsilon})$ approximates R_k which, in turn, ε approximates L^{-1} yielding overall an $O(\varepsilon)$ approximation. As the ROBP that correspond to each summand is short (recall $i \leq k = O(\log_n \frac{1}{\varepsilon}) \ll n$), a short seed is sufficient even for the high accuracy $\varepsilon' = \mathrm{poly}(\varepsilon)$ that we require. We invoke [15] as our auxiliary PRG as it has good dependence on the alphabet size which, in our case, is comparable to the seed of the crude PRG that we started with. We remark that the weights in our PRPD are used so to mimic Equation (3). Indeed, on top of the sign, there are $\binom{n}{i}$ summands that correspond to partition to $i+1$ segments and so the weights are used for creating the appropriate scaling between different values of i.

Discussion

While $C_1[a,b] = B[a,b]$ is obtained by PRG, $C_0[a,b]$ is computed by following the instructions of PRG for all but the first step. For the latter, we use a fresh random bit. Namely, consider a thought experiment in which we use a new–more expensive–PRG $\mathsf{PRG}'\colon \{0,1\}^{s+1} \to \{0,1\}^{\ell}$ that is defined by $\mathsf{PRG}'(\sigma,p) = p \circ \mathsf{PRG}(\sigma)_{[1,\ell-1]}$, where $\sigma\colon \{0,1\}^s$ and $p \in \{0,1\}$. The matrix $\Delta[a,b] = C_1[a,b] - C_0[a,b]$ then compares the better approximation $C_1[a,b]$ with the "actual" approximation $C_0[a,b]$. From this perspective, Equation (3) suggests interpreting the Richardson iteration as a linear combination with ±1 coefficients (as determined by $(-1)^{t_1+\cdots+t_i}$) of approximations of $A_n \cdots A_1$ where each approximation is partition to segments (encoded by $\ell_1 > \cdots > \ell_i$). In segment j, according to the value t_j, the relevant sequence of instructions is obtained either from the original PRG or via the refined one PRG'.

1.7 A comparison with [5]

It is worthwhile to explore the differences between the BCG construction [5] (and the followup work of Chattopadhyay and Liao [6] which uses similar ideas) and ours and to point out the aspects of our work that we find similar to the work of Cheng and Hoza [7], and of Hoza and Zuckerman [13]. We start by giving a brief overview of the BCG construction.

1.7.1 A brief overview of BCG

In constructions prior to [5] (e.g., [17, 15]), a list of instructions is maintained with the property that given a ROBP $A_1, \dots, A_n$, averaging over the products corresponding to the instructions yields the desired approximation to the product $A_n \cdots A_1$. The key idea suggested in [5] is to maintain not a single list whose average yields the desired approximation but rather several lists of instructions $L_0, L_1, \dots, L_k$ such that averaging according to the instructions in L_0 yields a modest approximation; averaging according to $L_0 \cup L_1$ yields a more refined approximation, and so forth. Averaging according to the instructions given by $L_0 \cup \cdots \cup L_k$ gives the desired approximation. Thus, L_0 can be thought of as a crude approximation, L_1 a first order correction term, L_2 a second order correction term, etc.

To implement this idea, weights were introduced and, moreover, each list but for L_0 was in itself a list of lists, or bundles. The different instructions in a bundle did not carry useful information by themselves and it is the bundle which has the desired properties. Lists that correspond to higher error terms requires the expensive use of bigger bundles and larger weights, and so a delicate use of balanced and unbalanced samplers is employed in [5] in order to maintain the desired invariant throughout the recursion and assuring that the bundles and weights do not get too large.

1.7.2 Comparison with BCG

Our work, in comparison, goes back to the use of a single list as in [17, 15]. We do not need to maintain several lists, let alone lists of bundles. This makes our construction significantly simpler and, in particular, spares us from the delicate application of different types of samplers. The only component we do need are weights, both positive and negative that are unbounded in absolute value. However, it is straightforward to pinpoint the weights used by our construction (see Equation (11)) whereas in [5] the weights are computed via a recursive algorithm. As a result, it is difficult to argue about them. We believe that the simpler and more explicit structure of our construction would enable future works to combine our construction with other ideas for the purpose of obtaining improved constructions and derandomization results.

The common theme to both our construction and BCG is working with cancellations. We “read off” the Richardson iteration what cancellations to consider. As we discussed in the end of Section 1.6, we interpret Richardson iteration as comparing a PRG with the PRG obtained by replacing the first bit by a fresh truly random bit. The BCG construction, on the other hand, “plants” cancellations by considering two samplers–one more refined than the other–and encode their difference in their lists (this requires the introduction of bundles). So, in a sense, BCG’s cancellations are obtained by comparing one approximation to another where both approximations are obtained via samplers whereas we make use of one approximation coming from a PRG and another that is obtained by replacing the first bit by a fresh truly uniform bit. The way we combine these is dictated by Richardson iteration.

1.7.3 Common aspects with [13, 7]

For their derandomization result, Cheng and Hoza [7] introduce the notion of *local consistency*. Informally, the authors consider the difference between applying a generated sequence of instructions (via a hitting set) to that obtained by the generated sequence when replacing the last bit with a fresh truly random bit. This is somewhat reminisce to the way we read the cancellations of the Richardson iteration. However, while local consistency is used for making decisions once a ROBP is given, we combine the analog sequences using the Richardson iterator in a block-box matter.

The construction of Hoza and Zuckerman [13] also shares similar aspects with ours. There, they start with a modest-error PRG to get an ε-error hitting set by running the PRG for $k = \log_n(1/\varepsilon)$ times according to partitions of $[n]$ to k segments, resembling what we do. Instead of drawing the PRG's seeds uniformly at random, they derandomize the construction using a hitter. We note however, that their analysis is very different from ours, and uses a progress measure concerning the probability of reaching an accepting state.

2 Preliminaries

2.1 Matrices, branching programs, and space complexity

A matrix is Boolean if all its entries are in $\{0,1\}$, and stochastic if all its entries are nonnegative and the sum of each column is 1. Denote by $\mathrm{BSto}(w)$ the set of $w \times w$ boolean stochastic matrices. We will denote by $\|\cdot\|$ the induced ℓ_1 norm, i.e., $\|A\| = \max_j \sum_i |A_{i,j}|$.

We will often work with block matrices. For instance, we may interpret $A \in \mathbb{R}^{nm \times nm}$ as an $n \times n$ matrix with entries which are $m \times m$ matrices. Whenever this interpretation is clear, we let $A[i,j]$ be the (i,j)-th block. In this example, $A[i,j] \in \mathbb{R}^{m \times m}$.

▶ **Definition 4** (branching program). *Let Σ be some alphabet and let $n, w \in \mathbb{N}$. An (n, Σ, w) branching program (BP) is a sequence $\overline{B} = (B_1, \dots, B_n)$, where each $B_i \colon \Sigma \to \mathrm{BSto}(w)$.*

For $b \leq a$ we let $B_{[b,a]}$ be the $(a - b + 1, \Sigma, w)$ BP $(B_a, \dots, B_b)$.

▶ **Definition 5.** *The* value *of an (n, Σ, w) BP $\overline{B} = (B_1, \dots, B_n)$ on $x = (x_1, \dots, x_n) \in \Sigma^n$, denoted $\mathrm{val}(\overline{B}, x)$, is the realized $w \times w$ matrix of $\overline{B}$ when fed by x, i.e.*

$$\mathrm{val}\big(\overline{B}, x\big) = B_n(x_n) \cdot B_{n-1}(x_{n-1}) \cdots B_1(x_1).$$

If $\overline{B}$ is the empty sequence, we set $\mathrm{val}(\emptyset, x) = I_w$.

▶ **Definition 6** (weighted PRG). *We say W is an $(n, \Sigma, w, \varepsilon)$-WPRG against BPs with seed length s if:*

- *$W = (I, \mu)$ where $I \colon \{0,1\}^s \to \Sigma^n$ and $\mu \colon \{0,1\}^s \to \mathbb{R}$, and,*
- *For every (n, Σ, w) BP $\overline{B} = (B_1, \dots, B_n)$, it holds that*

$$\left\| \mathop{\mathbb{E}}_{x \in \{0,1\}^s} \left[\mu(x) \cdot \mathrm{val}\big(\overline{B}, I(x)\big)\right] - \mathop{\mathbb{E}}_{x \in \Sigma^n} \left[\mathrm{val}\big(\overline{B}, x\big)\right] \right\| \leq \varepsilon.$$

When $\mu \equiv 1$, we say that W is a PRG.

For $1 \leq \ell \leq n$ we let $G_\ell \colon \{0,1\}^{s_0} \to \Sigma^\ell$ be the first ℓ symbols of the output of G. Note that if $G \colon \{0,1\}^{s_0} \to \Sigma^n$ is an $(n, \Sigma, w, \varepsilon)$ PRG then G_ℓ is an $(\ell, \Sigma, w, \varepsilon)$ PRG.

We say $f : \Lambda_1 \to \Lambda_2$ is computable in space s, if given $x \in \Lambda_1$ and index j, $f(x)_j \in \Lambda_2$ can be computed in additional work space that consists of s bits. We will use the following well known theorem regarding the space complexity of compositions.

▶ **Theorem 7.** *Let $f_1, f_2 \colon \{0,1\}^\star \to \{0,1\}^\star$ be two functions that can be computed in $s_1, s_2 \colon \mathbb{N} \to \mathbb{N}$ space such that $s_1(n), s_2(n) = \Omega(\log n)$. Then, on input x, $f_2 \circ f_1 : \{0,1\}^\star \to \{0,1\}^\star$ can be computed using $O(s_1(|x|) + s_2(|f_1(x)|))$ space.*

2.2 Known PRG constructions

▶ **Theorem 8** ([17, 18]). *For any positive integers n, w, any error parameter $\varepsilon > 0$ and any alphabet Σ, there exists an $(n, \Sigma, w, \varepsilon)$ PRG with seed length*

$$s \;=\; O\left(\log n \cdot \log \frac{nw|\Sigma|}{\varepsilon}\right),$$

computable in space $\min\left\{O\left(\log \frac{nw|\Sigma|}{\varepsilon}\right), O\left(\log n \cdot \log\log \frac{nw|\Sigma|}{\varepsilon}\right)\right\}$.

▶ **Theorem 9** ([15]). *For any positive integers n, w, any error parameter $\varepsilon > 0$ and any alphabet Σ, there exists an $(n, \Sigma, w, \varepsilon)$ PRG with seed length*

$$s \;=\; \log|\Sigma| + O\left(\log n \cdot \log\left(\frac{nw}{\varepsilon}\right)\right),$$

computable in space $O\left(\log n \cdot \left(\log\log \frac{nw|\Sigma|}{\varepsilon}\right)^2\right)$.

Theorem 8 is derived almost directly from [17, 18], and Theorem 9 follows from [15], except for the space complexity which is implicit in those works and also depends on the specific implementation. For completeness, we give the proof of Theorem 8 in Appendix B.1, and of Theorem 9 in Appendix B.3.

3 Richardson iteration

Let A be an invertible $n \times n$ real matrix, and assume that B approximates A^{-1}, concretely, $\|B - A^{-1}\| \le \varepsilon_0$ for some sub-multiplicative norm. Richardson iteration is a method for obtaining a more refined approximation of A^{-1} given access to the crude B as well as to the original matrix A.

▶ **Lemma 10.** *Let $L \in \mathbb{R}^{m\times m}$ be an invertible matrix and $A \in \mathbb{R}^{m\times m}$ such that $\left\|L^{-1} - A\right\| \le \varepsilon_0$. For any nonnegative integer k, define*

$$\mathrm{R}(A, L, k) = \sum_{i=0}^{k} (I - AL)^i A.$$

Then, $\left\|L^{-1} - \mathrm{R}(A, L, k)\right\| \le \left\|L^{-1}\right\| \cdot \|L\|^{k+1} \cdot \varepsilon_0^{k+1}$.

The proof is deferred to Appendix A.

Following [1] we will be interested in the following instantiation of the Richardson iteration. Let $\overline{M} = (M_1, \dots, M_n)$ be a sequence of $w \times w$ matrices. We consider the $(n+1)w \times (n+1)w$ matrix

$$M = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ M_1 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & M_n & 0 \end{pmatrix}. \tag{4}$$

The Laplacian of M is $L = I_{(n+1)w} - M$, and we treat L as an $(n+1) \times (n+1)$ block matrix. The following claim follows by a simple calculation.

▷ Claim 11. For $i, j \in [n+1]$, the (i,j)-th block of L^{-1} is given by

$$L^{-1}[i,j] = \begin{cases} M_{i-1} \cdots M_j & i > j, \\ I_w & i = j, \\ 0 & i < j. \end{cases}$$

Richardson for branching programs

Let $\overline{B} = (B_1, \ldots, B_n)$ be an (n, Σ, w) BP and let $M_i = \mathbb{E}_{\sigma \in \Sigma}[B_i(\sigma)]$ be the corresponding transition matrices. Thus, approximating the transition probabilities of $\overline{B}$,

$$\mathop{\mathbb{E}}_{x \in \Sigma^n}\left[\mathrm{val}\left(\overline{B}, x\right)\right] = M_n \cdots M_1,$$

amounts to approximating the lowest leftmost entry $L^{-1}[n+1, 1]$.

▷ Claim 12. Let $\overline{B} = (B_1, \ldots, B_n)$ be an (n, Σ, w) BP. Set $M_i = \mathbb{E}_{\sigma \in \Sigma}[B_i(\sigma)]$ and L as in Equation (4). Also, let $G\colon \{0,1\}^s \to \Sigma^n$ be an $(n, \Sigma, w, \varepsilon_0)$ PRG and consider

$$A[a,b] = \begin{cases} \mathbb{E}_{x \in \{0,1\}^s}\left[\mathrm{val}\left(B_{[b,a-1]}, G_{a-b}(x)\right)\right], & a \geq b \\ 0 & a < b. \end{cases} \tag{5}$$

Then,

$$\left\|L^{-1} - \mathrm{R}(A, L, k)\right\| \leq (n+1) \cdot (2\varepsilon_0)^{k+1}.$$

Let A as in Equation (5) and write $\mathrm{R}(A, L, k) = \sum_{i=0}^{k} \Delta^i A$ where $\Delta = I - AL$. Denote $A' = A - I$, i.e., A' is the part of A below the main diagonal. Then,

$$\Delta = I - AL = I - A(I - M) = (I - A) + AM = AM - A'.$$

In block notation, for $a, b \in [n+1]$, following Equation (4),

$$AM[a,b] = \sum_{i=1}^{n+1} A[a,i]M[i,b] = A[a,b+1]M[b+1,b] = A[a,b+1] \cdot M_b.$$

Thus,

$$\Delta[a,b] = \begin{cases} A[a,b+1] \cdot M_b - A[a,b] & a > b, \\ 0 & a \leq b. \end{cases} \tag{6}$$

Going back to $\mathrm{R}(A, L, k)$, for $a > b$ we have that

$$\mathrm{R}(A,L,k)[a,b] = A[a,b] + \sum_{i=1}^{k} \sum_{a > r_i > \cdots > r_1 \geq b} \Delta[a, r_i] \cdot \Delta[r_i, r_{i-1}] \cdots \Delta[r_2, r_1] \cdot A[r_1, b]. \tag{7}$$

If we further let $C_0[a,b] = A[a,b+1] \cdot M_b$ and $C_1[a,b] = A[a,b]$, then

$$\mathrm{R}(A,L,k)[a,b] = A[a,b] + \sum_{\substack{\bar{t} \in \{0,1\}^i \\ a > r_i > \cdots > r_1 \geq b}} \sum_{t_1, \ldots, t_i \in \{0,1\}} (-1)^{t_1 + \cdots + t_i} \cdot C_{t_i}[a, r_i] \cdots C_{t_1}[r_2, r_1] \cdot A[r_1, b]. \tag{8}$$

4 The construction

4.1 Black-box error reduction

Let $G\colon \{0,1\}^{s_0} \to \Sigma^n$ be an $(n, \Sigma, w, \varepsilon_G)$ and $G_{\text{aux}}\colon \{0,1\}^{s_{\text{aux}}} \to (\{0,1\}^{s_0} \times \Sigma)^{k+1}$ be a $(k+1, \{0,1\}^{s_0} \times \Sigma, w, \varepsilon_{\text{aux}})$ PRG. Also, for $t \in \{0,1\}$ and $\sigma \in \Sigma$ we let

$$G_{t,\ell}(x,\sigma) = \begin{cases} \sigma \circ G_{\ell-1}(x) & t = 0, \\ G_\ell(x) & t = 1. \end{cases} \tag{9}$$

We now define the WPRG $(I, \mu)\colon \{0,1\}^s \to \Sigma \times \mathbb{R}$. The seed $x \in \{0,1\}^s$ to our WPRG is interpreted as follows.

- The first $\log(k+1)$ bits encode $i \in \{0, \dots, k\}$.
- The next $\log \binom{n}{i}$ bits encode a sequence $\overline{\ell} = (\ell_0, \ell_1, \dots, \ell_i)$ such that $\ell_0 + \cdots + \ell_i = n$, $\ell_i, \dots, \ell_1 > 0$, and $\ell_0 \geq 0$.
- The next i bits are denoted by $\bar{t} = \bar{t} = (t_1, \dots, t_i) \in \{0,1\}^i$.
- The next s_{aux} bits are denoted by $x_{\text{aux}} \in \{0,1\}^{s_{\text{aux}}}$.

Overall, we can write $x = (i, \overline{\ell}, \bar{t}, x_{\text{aux}})$, and the WPRG (I, μ) has seed length

$$s = s_{\text{aux}} + O(k \log n). \tag{10}$$

For brevity we sometimes omit the dependence of i, $(\ell_0, \dots, \ell_i)$, $(t_1, \dots, t_i)$, and x_{aux} on x. We define I and μ as follows.

$$I(x) = \begin{cases} G_n(G_{\text{aux}}(x_{\text{aux}})_0) & i = 0, \\ G_{t_i,\ell_i}(G_{\text{aux}}(x_{\text{aux}})_i) \circ \cdots \circ G_{t_1,\ell_1}(G_{\text{aux}}(x_{\text{aux}})_1) \circ G_{\ell_0}(G_{\text{aux}}(x_{\text{aux}})_0) & \text{otherwise.} \end{cases}$$

$$\mu(x) = \begin{cases} k+1 & i = 0, \\ (k+1) \cdot \binom{n}{i} \cdot 2^i \cdot (-1)^{t_1 + \cdots + t_i} & \text{otherwise.} \end{cases} \tag{11}$$

where $G_{\text{aux}}(x_{\text{aux}})_j$ denotes the j'th symbol in $G_{\text{aux}}(x_{\text{aux}}) \in (\{0,1\}^{s_0} \times \Sigma)^{k+1}$.

The weights are chosen so that the approximation yielded by the above WPRG is a derandomized version of Equation (8) for $(a,b) = (n+1, 1)$. Note that in Equation (8) we used $r_1, \dots, r_i$ which partitioned the interval $[n+1, 1]$, while in Equation (11) we used $\ell_0, \dots, \ell_i$ that sum to n. This is merely an alternative way of writing the sum – the ℓ_i-s are the sum of differences of the r_i-s.

4.2 Correctness

In this section we use the same notation as in Section 3.

▶ **Lemma 13.** *Let* $0 < \varepsilon < \varepsilon_0 = \frac{1}{4n}$ *and let* $k = \log_{1/\varepsilon_0}(1/\varepsilon)$*. Suppose*

- $G\colon \{0,1\}^{s_0} \to \Sigma^n$ *is an* $\left(n, \Sigma, w, \varepsilon_G = \frac{\varepsilon_0}{2(n+1)}\right)$ *PRG, and,*
- $G_{\text{aux}}\colon \{0,1\}^{s_{\text{aux}}} \to (\{0,1\}^{s_0} \times \Sigma)^{k+1}$ *is a* $(k+1, \{0,1\}^{s_0} \times \Sigma, w, \varepsilon_{\text{aux}} = \varepsilon^3)$ *PRG.*

Then, (I, μ) *is an* $(n, \Sigma, w, \varepsilon)$ *WPRG with seed length* $s = s_{\text{aux}} + O(\log(1/\varepsilon))$ *computable in space* $O(\text{space}(G_{\text{aux}}) + \text{space}(G) + \log s)$*.*

Proof. Assume k, G and G_{aux} are as in the hypothesis of the lemma. The space complexity follows from Theorem 7 and the seed length was analyzed in Equation (10). We are left to prove that (I,μ) is an (n,Σ,w,ε) WPRG. Fix any (n,Σ,w) BP $\overline{B}=(B_1,\dots,B_n)$. Let A be the $(n+1)w\times(n+1)w$ lower triangular block matrix in which

$$A[a,b]=\mathop{\mathbb{E}}_{x\in\{0,1\}^{s_0}}\left[\operatorname{val}\left(B_{[b,a-1]},G_{a-b}(x)\right)\right]$$

for $a>b$, and $A[a,a]=I_w$. Since G is $\left(n,\Sigma,w,\varepsilon_G=\frac{\varepsilon_0}{2(n+1)}\right)$ PRG we have that

$$\left\|L^{-1}[a,b]-A[a,b]\right\|\le\varepsilon_G$$

and $\left\|L^{-1}-A\right\|\le(n+1)\varepsilon_G$. By our choice of μ,

$$\mathop{\mathbb{E}}_{x\in\{0,1\}^{s}}\left[\mu(x)\cdot\operatorname{val}\left(\overline{B},I(x)\right)\right]=\sum_{i=0}^{k}\sum_{\overline{t},\overline{\ell}}(-1)^{t_1+\dots+t_i}\cdot\mathop{\mathbb{E}}_{x_{\text{aux}}}\left[\operatorname{val}\left(\overline{B},I(i,\overline{\ell},\overline{t},x_{\text{aux}})\right)\right],$$

and

$$\begin{aligned}\mathrm{R}(A,L,k)[n+1,1]=A[n+1,1]+&\\ &\sum_{i=1}^{k}\sum_{\overline{t},\overline{r}}(-1)^{t_1+\dots+t_i}\cdot C_{t_i}[n+1,r_i]\cdots C_{t_1}[r_2,r_1]\cdot A[r_1,1],\end{aligned}$$

where $\ell_0+\dots+\ell_i=n$ and $n+1>r_i>\dots>r_1\ge1$. We soon prove:

▷ Claim 14. For every fixed $i\in\{0,\dots,k\}$, $\overline{t}\in\{0,1\}^i$, and $\overline{\ell}$ such that $\ell_0+\dots+\ell_i=n$

$$\left\|\mathop{\mathbb{E}}_{x_{\text{aux}}}\left[\operatorname{val}\left(\overline{B},I(i,\overline{\ell},\overline{t},x_{\text{aux}})\right)\right]-C_{t_i}[n+1,r_i]\cdots C_{t_1}[r_2,r_1]\cdot A[r_1,1]\right\|\ \le\ \varepsilon_{\text{aux}},$$

where $r_j=1+\ell_0+\dots+\ell_{j-1}$.

As we have at most $(k+1)n^k2^k$ summands, we see that

$$\begin{aligned}\left\|\mathop{\mathbb{E}}_{x\in\{0,1\}^{s}}\left[\mu(x)\cdot\operatorname{val}\left(\overline{B},I(x)\right)\right]-\mathrm{R}(A,L,k)[n+1,1]\right\|&\le(k+1)n^k2^k\cdot\varepsilon_{\text{aux}}\\ &\le\frac{n^{2k}}{2}\cdot\varepsilon_{\text{aux}}\le\frac{\varepsilon}{2}.\end{aligned}$$

It therefore follows from Claim 12 that

$$\begin{aligned}\left\|\mathrm{R}(A,L,k)[n+1,1]-\mathop{\mathbb{E}}_{x\in\Sigma^n}\left[\operatorname{val}\left(\overline{B},x\right)\right]\right\|&\le(n+1)\left(2(n+1)\varepsilon_G\right)^{k+1}\\ &\le2n\cdot\varepsilon_0^{k+1}\le2n\varepsilon_0\varepsilon=\frac{\varepsilon}{2},\end{aligned}$$

which together completes the proof. ◀

Proof of Claim 14. Fix $i\in\{0,\dots,k\}$, $\ell_0+\dots+\ell_i=n$, and $\overline{t}\in\{0,1\}^i$ and recall that $r_j=1+\ell_0+\dots+\ell_{j-1}$. We define a $(k+1,\{0,1\}^{s_0}\times\Sigma,w)$ BP $\overline{B'}=(B'_0,\dots,B'_k)$ (that depends on i, $\overline{\ell}$, and $\overline{t}$) such that for all $j=0,...,k$,

$$B'_j(x,\sigma)\quad=\quad\begin{cases}\operatorname{val}\left(B_{[r_j,r_{j+1}-1]},\sigma\circ G_{\ell_j-1}(x)\right) & j>0,t=0,\\ \operatorname{val}\left(B_{[r_j,r_{j+1}-1]},G_{\ell_j}(x)\right) & j>0,t=1,\\ \operatorname{val}\left(B_{[1,r_1-1]},G_{\ell_0}(x)\right) & j=0.\end{cases}\tag{12}$$

We stress that B'_j is a BP because a product of Boolean stochastic matrices is Boolean stochastic. The claim now follows since G_{aux} is a $(k+1,\{0,1\}^{s_0}\times\Sigma,w,\varepsilon_{\text{aux}})$ PRG. ◁

4.3 The final construction

We now instantiate Lemma 13 with G_{aux} being the INW PRG from Theorem 9 and G being an arbitrary PRG. The reason for using the INW generator is its additive dependence on $\log|\Sigma|$.

▶ **Corollary 15.** *Let $G\colon \{0,1\}^{s_0} \to \Sigma^n$ be an $(n,\Sigma,w,\varepsilon_G)$. Then, for any error parameter $\frac{1}{4n} > \varepsilon > 0$ there exists an (n,Σ,w,ε) WPRG with seed length*

$$s_0 + O\left(\log\frac{w}{\varepsilon}\cdot\log\log_n\frac{1}{\varepsilon}\right)$$

computable in space $O\Big(\mathrm{space}(G) + \log\log_n(1/\varepsilon)\cdot\big(\log\log\frac{w}{\varepsilon}\big)^2\Big)$.

Had we used Nisan's PRG from Theorem 8 instead of INW then the seed length would deteriorate to

$$O\left(s_0\cdot\log\log_n\frac{1}{\varepsilon} + \log\frac{w}{\varepsilon}\cdot\log\log_n\frac{1}{\varepsilon}\right).$$

Corollary 15 can be interpreted as an error reduction procedure for PRGs with a slight overhead in the seed and space complexity. We proceed by applying this error reduction to Nisan's PRG from Theorem 8.

▶ **Corollary 16.** *For any positive integers n, w, any error parameter $\frac{1}{4n} > \varepsilon > 0$ and any alphabet Σ, there exists an (n,Σ,w,ε) WPRG with seed length*

$$O\left(\log n\log(nw|\Sigma|) + \log\frac{w}{\varepsilon}\cdot\log\log_n\frac{1}{\varepsilon}\right)$$

computable in space $O\Big(\log(nw|\Sigma|) + \log\log_n(1/\varepsilon)\cdot\big(\log\log\frac{w}{\varepsilon}\big)^2\Big)$.

Note that for ε which is not tiny the space complexity is dominated by the first term. Specifically, for $\varepsilon > 2^{-2^{\log^{1/3} n}}$, $w < 2^{2^{\log^{1/3} n}}$ the space complexity is indeed $O(\log(nw|\Sigma|))$. Had we used INW instead, the space complexity would deteriorate to

$$O\left(\log n\cdot\left(\log\log\frac{nw|\Sigma|}{\varepsilon}\right)^2 + \log\frac{w}{\varepsilon}\cdot\log\log_n\frac{1}{\varepsilon}\right).$$

References

1 AmirMahdi Ahmadinejad, Jonathan Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil Vadhan. High-precision estimation of random walks in small space. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2020)*, pages 1295–1306. IEEE, 2020.

2 Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992.

3 Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: `http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264`.

4 Allan Borodin, Stephen Cook, and Nicholas Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58(1-3):113–136, 1983.

5 Mark Braverman, Gil Cohen, and Sumegha Garg. Pseudorandom pseudo-distributions with near-optimal error for read-once branching programs. *SIAM Journal on Computing*, 49(5):STOC18–242–STOC18–299, 2020.

6 Eshan Chattopadhyay and Jyun-Jie Liao. Optimal error pseudodistributions for read-once branching programs. In *Proceedings of the 35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

7 Kuan Cheng and William M. Hoza. Hitting sets give two-sided derandomization of small space. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

8 Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Anup B. Rao, Aaron Sidford, and Adrian Vladu. Almost linear-time algorithms for Markov chains and new spectral primitives for directed graphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*. ACM, 2017.

9 Michael B. Cohen, Jonathan Kelner, John Peebles, Richard Peng, Aaron Sidford, and Adrian Vladu. Faster algorithms for computing the stationary distribution, simulating random walks, and more. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*. IEEE, 2016.

10 Oded Goldreich. *Computational complexity: a conceptual perspective*. Cambridge University Press, Cambridge, 2008.

11 Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures & Algorithms*, 11(4):315–343, 1997.

12 Alexander Healy and Emanuele Viola. Constant-depth circuits for arithmetic in finite fields of characteristic two. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS 2006)*. Springer, 2006.

13 William M. Hoza and David Zuckerman. Simple optimal hitting sets for small-success **RL**. *SIAM Journal on Computing*, 49(4):811–820, 2020.

14 Russel Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672–694, 2002.

15 Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the 26th Annual ACM SIGACT Symposium on Theory of Computing (STOC 1994)*. ACM, 1994.

16 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *computational complexity*, 13(1-2):1–46, 2004.

17 Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

18 Noam Nisan. $\mathbf{RL} \subseteq \mathbf{SC}$. *computational complexity*, 4(1):1–11, 1994.

19 Noam Nisan and Avi Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

20 Edward Pyne and Salil Vadhan. personal communication, February 2021.

21 Ran Raz and Omer Reingold. On recycling the randomness of states in space bounded computation. In *Proceedings of the 31st Annual ACM SIGACT Symposium on Theory of Computing (STOC 1999)*. ACM, 1999.

22 Michael E. Saks and Shiyu Zhou. $\mathsf{BP_H SPACE}(\mathrm{S}) \subseteq \mathsf{DSPACE}(\mathrm{S}^{2/3})$. *Journal of Computer and System Sceinces*, 58(2):376–403, 1999.

23 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, April 1970.

A Proof of Lemma 10

We restate Lemma 10.

▶ **Lemma 17.** *Let $L \in \mathbb{R}^{m\times m}$ be an invertible matrix and $A \in \mathbb{R}^{m\times m}$ such that $\left\|L^{-1} - A\right\| \leq \varepsilon_0$. For any nonnegative integer k, define*

$$\mathrm{R}(A, L, k) = \sum_{i=0}^{k} (I - AL)^i A.$$

Then, $\left\|L^{-1} - \mathrm{R}(A, L, k)\right\| \leq \left\|L^{-1}\right\| \cdot \|L\|^{k+1} \cdot \varepsilon_0^{k+1}$.

Proof. For any matrix Z, the matrices I and Z commute, and so by a straightforward induction,

$$I - \sum_{i=0}^{k} (I - Z)^i Z = (I - Z)^{k+1}.$$

In particular, for $Z = AL$,

$$I - \mathrm{R}(A, L, k) \cdot L = (I - AL)^{k+1}.$$

Thus,

$$\begin{aligned}
\left\|L^{-1} - \mathrm{R}(A, L, k)\right\| &= \left\|(I - \mathrm{R}(A, L, k) \cdot L) \cdot L^{-1}\right\| \\
&\leq \left\|L^{-1}\right\| \cdot \|I - \mathrm{R}(A, L, k) \cdot L\| \\
&\leq \left\|L^{-1}\right\| \cdot \|I - AL\|^{k+1} \\
&= \left\|L^{-1}\right\| \cdot \left\|(L^{-1} - A) \cdot L\right\|^{k+1} \\
&\leq \left\|L^{-1}\right\| \cdot \|L\|^{k+1} \cdot \varepsilon_0^{k+1}.
\end{aligned}$$ ◀

B The space complexity of some pseudorandom objects

In this section we show how to achieve the space complexity declared in Theorem 8 and Theorem 9. For the INW generator we choose a specific implementation with a small space complexity. The constructions are well known, and the variant of INW we use was explored by [12]. We give it here for completeness.

B.1 Nisan's generator

Proof sketch of Theorem 8. We are given parameters $n, \Sigma, w, \varepsilon$. We set $X = [A]$ for $A = O\left(\frac{nw\Sigma}{\varepsilon}\right)$. We let $\mathcal{H}$ be a 2-universal family of hash functions over X where $|\mathcal{H}| = A^2$ and $h(x)$, for $h \in \mathcal{H}$ and $x \in X$, can be computed in space $O(\log\log|X|)$ (see [17, 18]).

Nisan's generator interprets the seed as $y, h_1, \ldots, h_{\log n}$, where $y \in X$, and $h_1, \ldots, h_{\log n} \in \mathcal{H}$. For $j \in [n]$, the j-th symbol in the output of the generator is $h_1^{b_1}\left(h_2^{b_2}\left(\cdots h_{\log n}^{b_{\log n}}(y)\right)\right)$, where $(b_1, \ldots, b_{\log n}) \in \{0, 1\}^{\log n}$ is the binary representation of j, and h^b is either h, if $b = 1$, or the identity function, if $b = 0$. Given $y, h_1, \ldots, h_{\log n}$, $j = (b_1, \ldots, b_{\log n})$ we can compute the j-th output symbol in the following two alternative ways.

- We can successively compute $h_j^{b_j}\left(\cdots h_{\log n}^{b_{\log n}}(y)\right)$ for $j = \log n, \ldots, 1$, each time keeping the current X-symbol. This takes
$$O\left(\log \frac{nw|\Sigma|}{\varepsilon} + \log\log n + \log\log|X|\right) = O\left(\log \frac{nw|\Sigma|}{\varepsilon}\right)$$
space.
- Alternatively, we can do the above computation using composition of space bounded reductions, resulting in space complexity
$$O(\log n \cdot \log\log|X|) = O\left(\log n \cdot \log\log \frac{nw|\Sigma|}{\varepsilon}\right). \qquad ◀$$

B.2 A high min-entropy extractor

To apply INW, we need a space-efficient *seeded extractor* with a small entropy loss in the high min-entropy regime. Goldreich and Wigderson [11] gave such a construction utilizing a regular expander $G = (V, E)$ with a small normalized second eigenvalue. For our expander, we choose a Cayley graph over the commutative group $\mathbb{Z}_2^n$ with a generator set $S \subseteq \{0,1\}^n$ that is λ-biased. It is well known that $Cay(\mathbb{Z}_2^n, S)$ has normalized second largest eigenvalue at most λ. For the λ-biased set we choose a construction from [2]. Altogether, this unfolds for the following.

- For the λ-biased set S, first pick q to be the first power of two larger than $\frac{n}{\lambda}$. The set S is of cardinality q^2. For every $\alpha, \beta \in \mathbb{F}_q$ there is an elements $s_{\alpha,\beta} \in \mathbb{Z}_2^n$ where $(s_{\alpha,\beta})_i = \langle \alpha^i, \beta \rangle$, such that multiplication is in $\mathbb{F}_q$ and the inner product is over $\mathbb{Z}_2$. [2] showed the set is λ-biased.
- We let $G = (V, E)$ with $V = \mathbb{Z}_2^n$ and $(x, y) \in E$ iff $x + y \in S$. G is a λ-expander.

The extractor $\mathsf{GW}\colon \{0,1\}^n \times [D] \to \{0,1\}^n$ is defined by letting $G(x, i)$ be the i-th neighbour of x in the graph G.

▷ Claim 18. Let $0 < \Delta < n$ and set G and GW as above. Then, $\mathsf{GW}\colon \{0,1\}^n \times [D] \to \{0,1\}^n$ is a $(k = n - \Delta, \varepsilon)$ extractor with seed length $d = O(\Delta + \log \frac{n}{\varepsilon})$ and space complexity $O(\log n \cdot \log(\Delta + \log(n/\varepsilon)))$.

Proof. For correctness, note that the expander mixing lemma shows that GW is an $(n-\Delta, \varepsilon = O(2^{\Delta/2}\lambda))$ extractor.

Seed length. The seed length of this extractor is $\log|S| = O(\log \frac{n}{\lambda}) = O(\log \frac{n2^{\Delta}}{\varepsilon}) = O(\Delta + \log \frac{n}{\varepsilon})$.

Space complexity. The space complexity of computing $\mathsf{GW}(x, y)$ given x and y, is the space needed to compute $s_y \in S$ from $y = (\alpha, \beta) \in \mathbb{F}_q^2$, plus the space needed to compute $x + s_y$. The dominating step in computing s_y is computing α^i (for $i \leq n$) which can be done in $O(\log n \log\log q)$ with space composition. Altogether, the space needed is $O(\log n \cdot \log\log \frac{n}{\lambda}) = O\left(\log n \cdot \log\log \frac{n2^{\Delta}}{\varepsilon}\right)$.

We note that Healy and Viola [12] gave an extremely efficient implementation of the above AGHP generator, yielding a better space complexity of $O(\log(n + \log q))$ to compute $\langle \alpha^i, \beta \rangle$. However, in our overall setting of parameters it will make negligible difference. ◁

We remark that by using expanders with better dependence between D and λ, one can get $d = O(\Delta + \log \frac{1}{\varepsilon})$, but here we care more about the space complexity, and $\log n$ factors are negligible for us.

B.3 The INW generator

Proof sketch of Theorem 9. We consider the INW generator [15] instantiated with extractors (as, e.g., in [21]). We are given parameters n, Σ, w, and $\varepsilon = \varepsilon_{\mathsf{INW}}$. We set parameters $\Delta = \log w + O(\log \frac{n}{\varepsilon})$, and d as the seed length for the extractor of Claim 18 for length n, error $\varepsilon_{\mathsf{Ext}} = \frac{\varepsilon}{n}$ and Δ. We let $s = \log|\Sigma| + \log n \cdot 2d$ and we assume $s \le n$. We let $\ell_i = s - i \cdot \Delta$ for $0 \le i \le n$.

Given a seed $x \in \{0,1\}^s$ we view the computation of $\mathsf{INW}(x)$ as a full binary tree of depth $\log n$. Nodes in level i of the tree are labeled by strings of length ℓ_i. The root (at level 0) is labeled by x (of length $\ell_0 = s$). Given any internal node in level $i \in \{0, \dots, \log n\}$ labeled by some string $z \in \{0,1\}^{\ell_i}$, we write $z = z_1 \circ z_2$ with $z_i \in \{0,1\}^{\ell_{i+1}}$ and $z_2 \in \{0,1\}^d$. The left child of z is labeled with z_1, and the right child of z is labeled with $\mathsf{Ext}_i(z_1, z_2)$, where Ext_i is given by Claim 18 for Δ, length ℓ_{i+1} and error $\varepsilon_{\mathsf{Ext}}$ (notice that since $\ell_i < n$, d bits suffice for the seed). $\mathsf{INW}(x)$ is the concatenation of the leaf's labels, from left to right, truncating outputs to $\log|\Sigma|$ bits.

Given an index $j \in [n]$, computing $\mathsf{INW}(x)_j \in \Sigma$ can be done by walking down the computation tree, and each time either truncating a string or invoking an extractor. By composition of space bounded reductions the space complexity of the construction is $\log n$ times the space complexity of the worst extractor used. That is, $\log n \cdot \log \ell_0 \cdot \log(\Delta + \log \frac{\ell_0}{\varepsilon_{\mathsf{Ext}}})$. Plugging-in Δ and $\varepsilon_{\mathsf{Ext}}$, the space complexity is bounded by

$$O\Big(\log n \cdot \log \ell_0 \cdot \log\log \frac{nw}{\varepsilon}\Big) = O\Big(\log n \cdot \log\Big(\log|\Sigma| + \log n \log \frac{nw}{\varepsilon}\Big) \cdot \log\log \frac{nw}{\varepsilon}\Big)$$

$$= O\left(\log n \cdot \left(\log\log \frac{nw|\Sigma|}{\varepsilon}\right)^2\right). \qquad ◀$$

A Stress-Free Sum-Of-Squares Lower Bound for Coloring

Pravesh K. Kothari
Carnegie Mellon University, Pittsburgh, PA, USA

Peter Manohar
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We prove that with high probability over the choice of a random graph G from the Erdős-Rényi distribution $G(n, 1/2)$, a natural $n^{O(\varepsilon^2 \log n)}$-time, degree $O(\varepsilon^2 \log n)$ sum-of-squares semidefinite program cannot refute the existence of a valid k-coloring of G for $k = n^{1/2+\varepsilon}$. Our result implies that the refutation guarantee of the basic semidefinite program (a close variant of the Lovász theta function) cannot be appreciably improved by a natural $o(\log n)$-degree sum-of-squares strengthening, and this is tight up to a $n^{o(1)}$ slack in k. To the best of our knowledge, this is the first lower bound for coloring $G(n, 1/2)$ for even a single round strengthening of the basic SDP in any SDP hierarchy.

Our proof relies on a new variant of instance-preserving *non-pointwise complete reduction* within SoS from coloring a graph to finding large independent sets in it. Our proof is (perhaps surprisingly) short, simple and *does not* require complicated spectral norm bounds on random matrices with dependent entries that have been otherwise necessary in the proofs of many similar results [12, 33, 45, 28, 51].

Our result formally holds for a constraint system where vertices are allowed to belong to multiple color classes; we leave the extension to the formally stronger formulation of coloring, where vertices must belong to unique colors classes, as an outstanding open problem.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Sum-of-Squares, Graph Coloring, Independent Set, Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.23

Related Version *Full Version*: https://arxiv.org/abs/2105.07517

Funding *Pravesh K. Kothari*: Supported by NSF CAREER Award #2047933.
Peter Manohar: NSF Graduate Research Fellowship Program and the ARCS Foundation. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1745016. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Acknowledgements We thank Xinyu Wu for taking part in early stages of this research, and the anonymous reviewers for providing valuable feedback.

1 Introduction

Starting with the seminal work of Arora, Bollobás, Lovász and Tourlakis [2], understanding the power of systematic hierarchies of linear and semidefinite programs for solving combinatorial optimization problems has been a foundational goal in complexity theory. This project has achieved many successes including sharp lower bounds for basic problems [58, 20, 54, 17, 15, 16, 27] in various hierarchies of linear and semidefinite programs [47, 52, 59, 49] (see [22, 26] for expositions).

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 23; pp. 23:1–23:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

However, proving lower bounds for the sum-of-squares (SoS) semidefinite programming hierarchy – the strongest known hierarchy of efficiently solvable convex programs – has achieved only a limited amount of success. This is partially explained by the remarkable success of the SoS hierarchy in designing state-of-the-art algorithms for worst-case optimization problems such as max-cut [29], sparsest cut [4], unique games on general [1] and algebraic graphs [5, 32], quantum separability [13] and more recently, a string of successes in high-dimensional algorithmic statistics including robust estimation of moments [44], clustering spherical [34, 43] and non-spherical mixture models [8, 24], robust learning of all Gaussian mixtures [6, 48], list-decodable learning [39, 55, 7, 56], tensor decomposition [50], and sparse [25] and tensor principal component analysis [36], among others. Indeed, given the remarkable power of the SoS method in designing algorithms for such *average-case* settings, SoS lower bounds (and related restricted algorithmic techniques such as the low-degree polynomial method [33, 37, 46]) are increasingly used to ascertain average-case hardness and *algorithmic thresholds.*

In the last few years, there has been some progress in proving sum-of-squares lower bounds for average-case problems [30, 31, 57, 61, 11, 12, 42, 28, 51]. However, such progress has come about via fairly technical[1], problem-specific arguments and a host of natural questions, e.g. combinatorial optimization on sparse random graphs, remain out of reach of current techniques. In particular, a central challenge in this line of work has been to analyze the sum-of-squares semidefinite programs for refuting the existence of a k-coloring in Erdős-Rényi random graphs. Classical works [19] in probability showed that the chromatic number of $G \sim G(n, 1/2)$[2] is tightly concentrated around $n/2\log_2 n$. However, the best known polynomial time algorithm (corresponding to the degree 2 SoS relaxation, a close variant of the famous Lovász theta function) can only *refute* the existence of a $\sqrt{n}$-coloring in such random graphs[3]. While it is natural to guess that higher-degree relaxations yield no significant improvement, establishing this has proved to be an elusive goal. Indeed, even the easier goal of establishing sharp SoS lower bounds for the clique number of $G \sim G(n, 1/2)$ required [12] the introduction of *pseudo-calibration* – a technique that has found several further uses in establishing SoS lower bounds for average-case problems. However, analyzing lower bound constructions based on pseudo-calibration requires understanding the spectra of complicated random matrices with dependent random entries. While this has been accomplished for a few select examples [33, 28], the case of graph coloring seems to be particularly unwieldy and has thus resisted progress so far.

In this paper, we establish a tight SoS lower bound for a natural higher-degree SoS relaxation of the graph coloring problem in $G(n, 1/2)$. Our proof circumvents pseudo-calibration entirely. Instead, we exhibit a *non-pointwise complete reduction* – a notion of reductions that departs from the standard framework introduced by Tulsiani [61] (and used in [18]) – that obtains a lower bound for the coloring problem from a lower bound for the independent set problem (see Section Section 1.3 for a detailed discussion). Somewhat surprisingly, our analysis does *not* require spectral analysis of complicated random matrices and instead succeeds whenever the lower bound construction for the independent set problem satisfies some natural covering properties. Our main result then follows by verifying these properties for the construction of [12].

[1] Almost all recent analyses run into ~ 50 pages!

[2] Recall $G \sim G(n, 1/2)$ is a graph on n vertices where each edge $\{i, j\}$ is independently included with probability 1/2.

[3] We note that a close variant of Lovász-theta function is also a crucial component in the current state-of-the-art algorithms for worst-case coloring of k-colorable graphs with a small polynomial number of colors [38, 3, 21].

1.1 Results

Our results apply to the following polynomial constraint system in the real-valued variables $\{x_{i,c}\}_{i\in[n],c\in[k]}$ that is satisfiable if and only if the graph G is k-colorable.

Color Constraints

$$
\begin{aligned}
x_{i,c}^2 &= x_{i,c} \text{ for all } i\in[n], c\in[k] && \text{(Booleanity Constraints)}\\
x_{i,c}x_{j,c} &= 0 \text{ for all } c\in[k] \text{ and } \{i,j\}\in E(G) && \text{(Edge Constraints)}\\
\sum_c x_{i,c} &\geqslant 1 \text{ for all } i\in[n] && \text{(Sum Constraints)}
\end{aligned}
$$

In Color Constraints, the variable $x_{i,c}$ represents the 0-1 indicator of whether the i vertex is in the c-th color class. The booleanity constraints enforce that $x_{i,c}\in\{0,1\}$, the edge constraints enforce that if $\{i,j\}\in E(G)$, then the subset of colors assigned to i is disjoint from the subset of colors assigned to j, and the sum constraints enforce that each vertex is in at least one color class.

Color Constraints allow for a vertex to be in *more than one* color class. Our lower bound technique does not currently succeed for the related set of constraints where each vertex must belong to exactly one color class. See Section 1.5 for a discussion on the difference between the formulations.

Our main result shows that with high probability over the draw of $G\sim G(n,1/2)$, the degree $O(\varepsilon^2\log n)$ SoS proof system cannot refute Color Constraints for G when $k=n^{\frac{1}{2}+\varepsilon}$.

▶ **Theorem 1.** *Let n be sufficiently large positive integer and $\varepsilon\in(\Omega(\sqrt{\frac{1}{\log n}}),\frac{1}{2})$. Then, for $k=n^{\frac{1}{2}+\varepsilon}$ and $d=O(\varepsilon^2\log n)$, with probability $1-1/\operatorname{poly}(n)$ over the draw of $G\sim G(n,1/2)$, the $n^{O(d)}$-time, degree d sum-of-squares relaxation of Color Constraints cannot refute the existence of a k-coloring of G.*

Equivalently, Theorem 1 says that with high probability over $G\sim G(n,1/2)$, Color Constraints do not admit an $O(\varepsilon^2\log n)$-degree positivstellensatz refutation when $k=n^{\frac{1}{2}+\varepsilon}$. As was formally verified in [10][4], a degree 2 coloring pseudo-expectation is equivalent to a vector solution with value at least k to the semidefinite program that computes the Lovász theta function. To the best of our knowledge, this result gives the first lower bound for $\omega(1)$ rounds (or even a single round of strengthening of the basic SDP) in a natural SDP hierarchy.

▶ Remark 1 (Tightness of Theorem 1). It is well-known [23, 9] that the degree 2 sum-of-squares relaxation of Color Constraints can refute the existence of k-coloring in $G\sim G(n,1/2)$ for $k=O(\sqrt{n})$. Thus, our lower bound in Theorem 1 is tight up to a n^{ε} factor in k. On the other hand, we give a simple proof in Appendix B that shows that the degree $8(1+o(1))\log_2 n$ SoS relaxation of Color Constraints succeeds in refuting the existence of a k-coloring in $G(n,1/2)$ (w.h.p.) for the nearly optimal [19] bound of $k\leqslant\frac{n}{e\cdot 2(1+o(1))\log_2 n}$. Hence, the upper bound on d in Theorem 1 is tight up to constants.

[4] [10] proved this equivalence for a slightly different formulation of Color Constraints, which we will discuss in Section 1.5. However, the same proof works even for Color Constraints.

1.2 A non-pointwise complete SoS reduction from coloring to independent set

Using standard SDP duality, proving Theorem 1 is equivalent to proving the existence of a dual witness called a pseudo-expectation defined below (see lecture notes [14] and the monograph [26] for background).

▶ **Definition 2** (Pseudo-expectation for Coloring). *A degree d coloring pseudo-expectation $\tilde{\mathbb{E}}$ for G using k colors is a linear operator that maps polynomials of degree $\leqslant d$ in variables $\{x_{i,c}\}_{i\in[n],c\in[k]}$ to $\mathbb{R}$, satisfying the following three properties:*

1. ***Normalization:*** $\tilde{\mathbb{E}}[1] = 1$*,*
2. ***Positivity:*** $\tilde{\mathbb{E}}[f^2] \geqslant 0$ *for every polynomial f of degree at most $d/2$,*
3. ***Coloring Constraints**: $\tilde{\mathbb{E}}$ satisfies Color Constraints.*
   (a) *for every polynomial f of degree at most $d-2$, $\tilde{\mathbb{E}}[f \cdot (x_{i,c}^2 - x_{i,c})] = 0$,*
   (b) *for every polynomial f of degree at most $d-2$ and any edge $\{i,j\} \in E(G)$, $\tilde{\mathbb{E}}[f \cdot x_{i,c}x_{j,c}] = 0$,*
   (c) *for every polynomial f of degree at most $\frac{d-1}{2}$, $\tilde{\mathbb{E}}[f^2 \cdot (\sum_{c\leqslant k} x_{i,c} - 1)] \geqslant 0$.*

In order to prove Theorem 1, it suffices to show that with high probability over the draw of $G \sim G(n,1/2)$, there is a degree $O(\varepsilon^2 \log n)$ coloring pseudo-expectation for the graph G that uses $k = n^{\frac{1}{2}+\varepsilon}$ colors. Somewhat surprisingly, we prove the existence of such a pseudo-expectation essentially without any random matrix analysis. Instead, we construct a coloring pseudo-expectation $\tilde{\mathbb{E}}'$ for G from a pseudo-expectation $\tilde{\mathbb{E}}$ satisfying the related independent set constraints for *the same graph G* whenever $\tilde{\mathbb{E}}$ satisfies two additional natural "covering" properties. We recall the definition of an independent set pseudo-expectation below.

▶ **Definition 3** (Pseudo-expectation for Independent Set). *A degree d independent set pseudo-expectation $\tilde{\mathbb{E}}$ is a linear operator that maps polynomials of degree $\leqslant d$ in variables $\{x_i\}_{i\in[n]}$ to $\mathbb{R}$, satisfying the following three properties:*

1. ***Normalization:*** $\tilde{\mathbb{E}}[1] = 1$*,*
2. ***Positivity:*** $\tilde{\mathbb{E}}[f^2] \geqslant 0$ *for every polynomial f of degree at most $d/2$,*
3. ***Independent Set Constraints:*** *For every polynomial f of degree at most $d-2$, $\tilde{\mathbb{E}}[f \cdot (x_i^2 - x_i)] = 0$ and $\tilde{\mathbb{E}}[f \cdot x_i x_j] = 0$ for any edge $\{i,j\} \in E(G)$.*

Our main result that constructs a reduction from coloring to independent set is described below.

▶ **Theorem 2.** *Let G be a graph on n vertices, and let $\tilde{\mathbb{E}}$ be a degree d independent set pseudo-expectation. Suppose further that $\tilde{\mathbb{E}}$ satisfies the two "covering" properties: (1) $\tilde{\mathbb{E}}[x_i] \geqslant \frac{1}{k_0}$ for some integer k_0, and (2) there exists $\lambda \in \mathbb{R}_{>0}$ such that for all multilinear f with $\deg(f) \leqslant d/2$, $\tilde{\mathbb{E}}[f^2] \geqslant \lambda \|\Pi_G f\|_2^2$, where Π_G is the projection of f onto the linear subspace orthogonal to $\{g x_i x_j : \{i,j\} \in E(G), \deg(g) \leqslant d-2\}$ (viewed as a subset of coefficient vectors of polynomials with the Euclidean inner product), and $\|f\|_2$ denotes the ℓ_2 norm of the polynomial of f, viewed as a coefficient vector. Then, there is a degree $d' := 1 + d/2$ coloring pseudo-expectation $\tilde{\mathbb{E}}'$ using $k = O(k_0 d \log(n^d/\lambda))$ colors.*

Theorem 1 follows by verifying (see Section 3) that the independent set pseudo-expectation constructed in [12] satisfies the hypotheses of Theorem 2 with $k_0 = n^{\frac{1}{2}+\varepsilon}$ and $\lambda = n^{-O(d)}$.

Theorem 2 holds for *every* graph G that admits an independent set pseudo-expectation satisfying the two additional covering properties. Hence, Theorem 2 gives a reduction "within SoS" from the problem of coloring G to the problem of finding a large independent set in G. As a consequence of the modularity of Theorem 2, we have also reduced the task of proving

SoS lower bounds for coloring for $G(n, p)$ with $p \ll 1/2$ to the task of "merely" proving a similar lower bound for independent set for $G(n, p)$. The latter task, though challenging, appears significantly less daunting than attacking coloring directly.

To understand the two covering properties intuitively, note that even in "real-life" the existence of a single large independent set (say of size $\sim n/k$) does not imply the existence of a k-coloring of G. However, the existence of a k-coloring follows if we can prove that there is a collection of k independent sets that *cover* all vertices of G. The conditions appearing in Theorem 2 can be thought of as forcing two "low-degree" consequences of such a uniform covering property on the pseudo-expectation for independent sets. Informally, the first constraint says that each vertex i appears in the independent set with reasonable probability, and the second constraint says that the minimum eigenvalue of $\tilde{\mathbb{E}}$ is not too small, once we ignore polynomials that are required to have pseudo-expectation 0 due to the independent set constraints.

1.3 Comparison with Tulsiani's framework

Our proof of Theorem 2 requires a notion of reduction that departs from the standard framework introduced in [60]. Tulsiani's method[5] uses a *pointwise complete* reduction from problem B to problem A to construct a pseudo-expectation consistent with a polynomial formulation for B from a pseudo-expectation consistent with a polynomial formulation for A. Specifically, a *pointwise complete* SoS reduction from problem B to problem A is a map from instances I_A of problem A to instances I_B of problem B, along with a "solution map" $x \mapsto y$ that takes any solution x of instance I_A into a solution y of instance I_B that, in addition, satisfies: (1) each entry of the solution map $x \to y$ is computable by low-degree polynomials, and (2) there is a "low-degree sum-of-squares proof" that if x satisfies the constraint system A for instance I_A then y satisfies the constraint system B for instance I_B. In particular, if $y_i = p_i(x)$ for each i for polynomials $p_1, p_2, \ldots$ of degree most d_1, then the framework allows us to transform a degree d pseudo-expectation consistent with A into a degree $\approx d/d_1$ pseudo-expectation consistent with B. Tulsiani used this machinery to prove several SoS lower bounds for *worst-case* combinatorial optimization problems such as constraint satisfaction, vertex cover, independent set and coloring.

In average-case settings, however, we need tight control over the map between instances I_A and I_B in order to obtain a lower bound that applies to the target distribution over the instances of problem B. This makes Tulsiani's method not directly applicable to our setting since (if we insist on instance-preserving reductions) there is provably no pointwise complete, instance-preserving reduction from k-coloring to independent set. This is because the existence of a large independent set in G does not, in general, imply the existence of a valid coloring of G with a small number of colors. Instead, as we discuss next, our reduction directly maps a pseudo-expectation for independent set into a pseudo-expectation for coloring as long as the pseudo-expectation for independent set satisfies the additional uniform covering conditions.

1.4 Proof overview: coloring by repeated sampling

We describe our construction and a couple of main insights that go into the proof of Theorem 2 here. These ideas make the proof of Theorem 1 "stress-free": they allow us to completely sidestep the technical complexity of analyzing constructions based on pseudo-calibration that involve computing the spectra of certain random matrices (called graphical matrices) for proving SoS lower bounds.

[5] What follows is an equivalent description of Tulsiani's work in the language of pseudo-expectations.

We begin by describing the conceptual heart of the idea. In order to do this, it is helpful to consider the thought experiment (and very special case!) where the independent set pseudo-expectation $\tilde{\mathbb{E}}$ is in fact the expectation operator $\mathbb{E}_\mu$ associated with some *distribution* μ on independent sets of G. Further, suppose that $\mathbb{E}_\mu[x_i] = \Pr[i \in S] \geqslant \frac{1}{k_0}$. Then, observe that we can immediately derive that G must be k-colorable with $O(k_0 \log n)$ colors. In fact, we can produce a simple, explicit probability distribution on k-colorings of G: independently sample k independent sets $S_1, \ldots, S_k$ from μ and set each of them to be a new color class. Observe that the chance that a certain vertex is not included in any of the S_i's is $\Pr[i \notin \cup_{j=1}^k S_j] = (1 - \frac{1}{k_0})^k \leqslant e^{-k/k_0} \ll \frac{1}{n}$ for $k = O(k_0 \log n)$, and hence by a union bound, using the S_i's as color classes gives a valid k-coloring of G with high probability.[6] To get a distribution μ' entirely supported over k-colorings of G, one can simply sample $S_1, \ldots, S_k$ from μ *conditioned* on the high probability event that $\cup_{j=1}^k S_j = V(G)$.

Our key idea is to replicate this "independent sampling" step within the sum-of-squares framework. For pseudo-expectations, independent sampling produces a pseudo-expectation on a tuple of k independent sets given by the k-th tensor $\tilde{\mathbb{E}}^{\otimes k}$. However, there is no natural way to perform the final "conditioning" step for low-degree pseudo-expectations[7], which for distributions is the simplest way to ensure the "covering property", that is, $i \in \cup_{j=1}^k S_j$ for every i, or equivalently to make $\mathbb{E}_{\mu'}$ satisfy the sum constraints $\sum_c x_{i,c} \geqslant 1$ for every i.

The sampling analogy suggests a way out, however: observe that when one draws $S_1, \ldots, S_k$ from an actual probability distribution μ on independent sets that satisfies $\mathbb{E}_\mu x_i \geqslant 1/k_0$, we expect each i to be in not just one but in fact in $\frac{k}{k_0} = O(\log n)$ of the subsets. Equivalently, we expect that $\mathbb{E}_\mu \sum_c x_{i,c} = \Omega(\log n)$. Because this expectation is large, if low-degree polynomials of μ are sufficiently well-concentrated around their expectations, we may expect that the influence of the points x in the support of μ where $\sum_c x_{i,c} \leqslant 1 \ll \mathbb{E}_\mu \sum_c x_{i,c}$ to be small. Thus, one may hope that expectations of low-degree ($\deg \leqslant d$) polynomials cannot "distinguish" between distributions μ where every point in the support of μ satisfies $\sum_c x_{i,c} > 1$ versus those where the probability of $\sum_c x_{i,c} = 0$ is non-zero for some i. In that case, one might expect $\tilde{\mathbb{E}}^{\otimes k}$ to satisfy the sum constraints.

Our actual proof establishes precisely such a statement even for pseudo-distributions whenever the smallest nontrivial eigenvalue of the pseudo-moment matrix of the independent set pseudo-expectation $\tilde{\mathbb{E}}$ is not too small. We show that this condition implies a non-trivial eigenvalue lower bound for the k-fold tensor power of $\tilde{\mathbb{E}}$ on polynomials of total degree[8] d. A direct argument relying on spectra of the tensor product of matrices yields an estimate that decays exponentially in k, which is too weak for us. Instead, we show that the smallest eigenvalue of $\tilde{\mathbb{E}}^{\otimes k}$ *when restricted to the subspace of polynomials of total degree* $\leqslant d$ decays only as an exponential in $d \log n$. While eventually elementary, this argument is both crucial and somewhat technical and is presented in full in Section 2.4.1.

Intuitively, a good enough lower bound on the smallest non-zero eigenvalue of $\tilde{\mathbb{E}}^{\otimes k}$ on the relevant subspace of polynomials is our "surrogate" for the concentration of low-degree polynomials that we needed in the case of actual probability distributions above. Concretely,

[6] Note that a vertex i will, with high probability, belong to multiple color classes. In order to obtain a valid k-coloring, we simply remove each vertex from all but one of its assigned color classes.

[7] There is a natural and standard way to import "conditioning" of probability distributions into the SoS framework via "polynomial reweightings" (see [13] for a formal treatment of such reweightings). However, the relevant polynomial $\prod_i (1 - \prod_c (1 - x_{i,c}))$ in our case has degree nk, and so we would need the independent set pseudo-expectation to have degree $\gg n$ in order for the reweighting to be well-defined!

[8] Notice that $\tilde{\mathbb{E}}^{\otimes k}$ is defined and even positive semidefinite on a *larger* subspace of polynomials that includes some of total degree $\sim kd$!

we use this non-trivial eigenvalue lower bound on $\tilde{\mathbb{E}}^{\otimes k}$ as follows: let h_i be the indicator polynomial of the "bad event" $\sum_c x_{i,c} \leqslant 1$. Then, we prove that for a polynomial f to be able to "detect" this event, we must have $\tilde{\mathbb{E}}^{\otimes k}[f^2 h_i] = \Omega(\tilde{\mathbb{E}}^{\otimes k}[f^2])$. However, applying Cauchy-Schwarz, we have that $\tilde{\mathbb{E}}^{\otimes k}[f^2 h_i] \leqslant \sqrt{\tilde{\mathbb{E}}^{\otimes k}[f^4]\tilde{\mathbb{E}}^{\otimes k}[h_i^2]}$. We show that the smallest eigenvalue condition implies a $2 \to 4$ *hypercontractive inequality* on the pseudo-expectation operator on polynomials of total degree $\leqslant d$, i.e., $\tilde{\mathbb{E}}^{\otimes k}[f^4] \leqslant (\frac{n^{O(d)}}{\lambda})^d \tilde{\mathbb{E}}^{\otimes k}[f^2]^2$. Combined with the estimate (that one roughly expects to hold from the independent sampling based argument) $\tilde{\mathbb{E}}^{\otimes k}[h_i] \approx e^{-k/k_0}$, this yields that $\tilde{\mathbb{E}}^{\otimes k}$ indeed satisfies the constraints $\sum_c x_{i,c} \geqslant 1$ for every i, when $k = O(k_0 d \log(n^d/\lambda))$.

1.5 Weak vs. strong formulation for coloring

The coloring axioms are often stated with an equality (we call this the *strong form*) in the sum constraints along with the additional constraints $\{x_{i,c}x_{i,c'} = 0 : c \neq c'\}$, instead of an inequality (the *weak form*) as done in Color Constraints. Namely, the strong coloring constraints are the following.

(Strong) Color Constraints

$$x_{i,c}^2 = x_{i,c} \text{ for all } i \in [n], c \in [k] \qquad \text{(Booleanity Constraints)}$$

$$x_{i,c}x_{j,c} = 0 \text{ for all } c \in [k] \text{ and } \{i,j\} \in E(G) \qquad \text{(Edge Constraints)}$$

$$\sum_c x_{i,c} = 1 \text{ for all } i \in [n] \qquad \text{(Sum Equality Constraints)}$$

$$x_{i,c}x_{i,c'} = 0 \text{ for all } c \neq c' \in [k] \qquad \text{(Same Color Constraints)}$$

When viewed as a polynomial optimization problem, there is no difference between the weak and strong formulations: one is satisfiable if and only if the other is. Further, SoS relaxations of both formulations "converge" (i.e., refute k-coloring in $G(n, 1/2)$ for the right value of k) at $O(\log n)$ degree, and both imply corresponding lower bounds for independent set: a degree d coloring (weak or strong) pseudo-expectation with k colors can easily be transformed into a degree d independent set pseudo-expectation with independent set size $\geqslant \frac{n}{k}$. Thus, while the SoS relaxation of the strong form is *formally* stronger (for degrees > 2), Color Constraints do not appear to meaningfully weaken the strong formulation.

However, at the moment our technique does not succeed in constructing a pseudo-expectation that satisfies the constraints in the strong formulation. This is an important technical issue encountered in proving several prior SoS lower bounds where it turns out to be unwieldy to handle "hard" constraints such as those formulated by an exact polynomial equality. For example, in the planted clique problem, one may naturally wish for the pseudo-expectation to satisfy the clique-size constraint "$\sum_i x_i = \omega$" *exactly*. While this is achieved for the degree 4 pseudo-expectation of [35, 53], the degree $\sim \log n$ pseudo-expectation constructed in [12] does *not* satisfy this as a constraint. This technical deficiency can sometimes even be crucial in downstream applications. For example, the construction of the hardness result for finding Nash equilibria in two player games in [41] (see also the discussion in [40]) needs elaborate work-arounds in order to work without satisfying such exact constraints.

Informally speaking, this is because the proofs of positivity of candidate pseudo-expectations rely on "collecting terms" together in the *graphical matrix* (a class of structured random matrices) decomposition in order to form PSD matrices. This aggregation step needs coefficients on various graphical matrices appearing in the decomposition to satisfy certain exact relationships. Modifying such coefficients to satisfy hard constraints while maintaining positivity appears challenging.

Such technical difficulty has been dealt with in some special cases (where the analyses did not need pseudo-calibration in the first place). For example, for the (much simpler) case of constraint satisfaction problems with a single global equality constraint, this problem was addressed via certain *ad hoc* methods in a recent work [40]. That work also includes a longer discussion on the issues arising in constructing pseudo-expectations satisfying hard constraints. Finding general techniques to design and analyze pseudo-expectations that exactly satisfy multiple hard constraints *simultaneously* – such as those arising in the strong formulation of graph coloring – is an important and challenging open problem.

2 Reduction to SoS Lower Bounds for Independent Set

In this section, we prove Theorem 2 (restated below).

▶ **Theorem** (Theorem 2, restated). *Let G be a graph on n vertices, and let $\tilde{\mathbb{E}}$ be a degree d independent set pseudo-expectation. Suppose further that $\tilde{\mathbb{E}}$ satisfies the two "covering" properties: (1) $\tilde{\mathbb{E}}[x_i] \geqslant \frac{1}{k_0}$ for some integer k_0, and (2) there exists $\lambda \in \mathbb{R}_{>0}$ such that for all multilinear f with $\deg(f) \leqslant d/2$, $\tilde{\mathbb{E}}[f^2] \geqslant \lambda \left\|\Pi_G f\right\|_2^2$, where Π_G is the projection of f onto the linear subspace orthogonal to $\{g x_i x_j : \{i,j\} \in E(G), \deg(g) \leqslant d-2\}$ (viewed as a subset of coefficient vectors of polynomials with the Euclidean inner product), and $\left\|f\right\|_2$ denotes the ℓ_2 norm of the polynomial of f, viewed as a coefficient vector. Then, there is a degree $d' := 1 + d/2$ coloring pseudo-expectation $\tilde{\mathbb{E}}'$ using $k = O(k_0 d \log(n^d/\lambda))$ colors.*

2.1 *Coloring degree* of polynomials

Before proceeding with the proof, we first introduce some notation. Let f be a polynomial in the variables $\{x_{i,c}\}_{i\in[n],c\in[k]}$. We define the *coloring degree* of f, denoted by $\mathrm{cdeg}(f)$, to be the maximum, taken over all monomials $\prod_{c=1}^{k}\prod_{i=1}^{n} x_{i,c}^{\alpha_{i,c}}$ for which f has a nonzero coefficient, of $\max_{c\in[k]} \deg(\prod_{i=1}^{n} x_{i,c}^{\alpha_{i,c}})$. As an example, the polynomial $x_{1,1}x_{1,2}$ has degree 2 and coloring degree 1, while the polynomial $x_{1,1}x_{2,1}$ has degree 2 and coloring degree 2. Informally, the coloring degree only "charges" a polynomial for degrees in variables of a single color.

Let $\mathcal{P}_d$ denote the set of polynomials in the variables $\{x_i\}_{i\in[n]}$ of degree at most d. Then, the set of coloring degree $\leqslant d$ polynomials is precisely $\mathcal{P}_d^{\otimes k}$. Recall that the operator Π_G is the projection of $f \in \mathcal{P}_d$ to the subspace orthogonal to $\{g x_i x_j : \{i,j\} \in E(G), \deg(g) \leqslant d-2\}$. We let $\Pi_G^{\otimes k}$ denote the k-th tensor of Π_G. Namely, $\Pi_G^{\otimes k}$ is the projection of $f \in \mathcal{P}_d^{\otimes k}$ to the subspace orthogonal to $\{g x_{i,c} x_{j,c} : \{i,j\} \in E(G), c \in [k], \mathrm{cdeg}(g x_{i,c} x_{j,c}) \leqslant d\}$.

Recall that a degree d pseudo-expectation is a linear operator $\tilde{\mathbb{E}}\colon \mathcal{P}_d \to \mathbb{R}$ such that $\tilde{\mathbb{E}}[1] = 1$, and $\tilde{\mathbb{E}}[f^2] \geqslant 0$ for all f with $\deg(f) \leqslant d/2$. For a pseudo-expectation $\tilde{\mathbb{E}}\colon \mathcal{P}_d \to \mathbb{R}$, we define $\tilde{\mathbb{E}}^{\otimes k}\colon \mathcal{P}_d^{\otimes k} \to \mathbb{R}$ to be the k-th tensor of $\tilde{\mathbb{E}}$. Concretely, $\tilde{\mathbb{E}}^{\otimes k}$ is a pseudo-expectation in the variables $\{x_{i,c}\}_{i\in[n],c\in[k]}$, defined as follows. For polynomials $f_1, \ldots, f_k$ where (1) f_c is a polynomial in the variables $\{x_{i,c}\}_{i\in[n]}$ for each c, and (2) $\deg(f_c) \leqslant d$ for all c, we first define $\tilde{\mathbb{E}}^{\otimes k}[\prod_{c=1}^{k} f_c] \overset{\mathrm{def}}{=} \prod_{c=1}^{k} \tilde{\mathbb{E}}[f_c]$, and then extend $\tilde{\mathbb{E}}^{\otimes k}$ to be defined on all $f \in \mathcal{P}_d^{\otimes k}$ via linearity. We define a *coloring degree d pseudo-expectation* to be a linear operator $\tilde{\mathbb{E}}\colon \mathcal{P}_d^{\otimes k} \to \mathbb{R}$ such that $\tilde{\mathbb{E}}[1] = 1$ and $\tilde{\mathbb{E}}[f^2] \geqslant 0$ for all f with $\mathrm{cdeg}(f) \leqslant d/2$. If $\tilde{\mathbb{E}}$ is a degree d pseudo-expectation, then $\tilde{\mathbb{E}}^{\otimes k}$ is a coloring degree d pseudo-expectation.

It is well-known that degree d pseudo-expectations satisfy the Cauchy-Schwarz inequality:

▶ **Fact 4** (See [14]). *Let f, g be polynomials with $\deg(f), \deg(g) \leqslant d/2$, and let $\tilde{\mathbb{E}}$ be a degree d pseudo-expectation. Then $\tilde{\mathbb{E}}[fg] \leqslant \sqrt{\tilde{\mathbb{E}}[f^2]\tilde{\mathbb{E}}[g^2]}$.*

We observe that a similar fact also holds for coloring degree d pseudo-expectations.

▶ **Fact 5.** *Let f, g be polynomials with $\mathrm{cdeg}(f), \mathrm{cdeg}(g) \leqslant d/2$, and let $\tilde{\mathbb{E}}$ be a coloring degree d pseudo-expectation. Then $\tilde{\mathbb{E}}[fg] \leqslant \sqrt{\tilde{\mathbb{E}}[f^2]\tilde{\mathbb{E}}[g^2]}$.*

The proof of Fact 5 is nearly identical to the proof of Fact 4, as the proof of Fact 4 merely requires that $\tilde{\mathbb{E}}$ is a pseudo-expectation where $\tilde{\mathbb{E}}[f^2]$, $\tilde{\mathbb{E}}[g^2]$, $\tilde{\mathbb{E}}[fg]$ and $\tilde{\mathbb{E}}[(f-g)^2]$ are all well-defined.

2.2 Proof of Theorem 2

Construction of the pseudo-expectation. Fix a graph G and degree bound d, and let $\tilde{\mathbb{E}} \colon \mathcal{P}_d \to \mathbb{R}$ be a degree d independent set pseudo-expectation for G such that (1) $\tilde{\mathbb{E}}[x_i] \geqslant \frac{1}{k_0}$, and (2) $\tilde{\mathbb{E}}[f^2] \geqslant \lambda \left\|\Pi_G f\right\|_2^2$. Let $k \in \mathbb{N}$ to be chosen later, and assume without loss of generality that $k < n$.

Let $\tilde{\mathbb{E}}^{\otimes k} \colon \mathcal{P}_d^{\otimes k} \to \mathbb{R}$ be the k-fold tensor power of $\tilde{\mathbb{E}}$, and let $\tilde{\mathbb{E}}'$ be the pseudo-expectation defined over all polynomials f that have *degree* at most $1 + d/2$ obtained by restricting $\tilde{\mathbb{E}}^{\otimes k}$ to this subspace.

Analysis of the constraints. We first observe that both $\tilde{\mathbb{E}}'$ and $\tilde{\mathbb{E}}^{\otimes k}$ trivially satisfy the booleanity constraints, edge constraints, and positivity constraint (over their respective domains), since $\tilde{\mathbb{E}}$ satisfies these constraints. We verify these simple facts in Appendix A. As a consequence, if f has $\Pi_G^{\otimes k} f = 0$, then $\tilde{\mathbb{E}}^{\otimes k}$ satisfies $f = 0$ as a constraint; namely, for any g with $\mathrm{cdeg}(fg) \leqslant d$, it holds that $\tilde{\mathbb{E}}^{\otimes k}[fg] = 0$.

It remains to show that $\tilde{\mathbb{E}}'$ satisfies the sum constraints, i.e., for all f with $\deg(f) \leqslant \frac{d}{4}$ and for every i, $\tilde{\mathbb{E}}'[f^2(\sum_c x_{i,c} - 1)] \geqslant 0$. Fix i, and let $h_i := \sum_c x_{i,c} \prod_{c' \neq c}(1 - x_{i,c'}) + \prod_c (1 - x_{i,c})$. Note that h_i is the indicator of the event that $\sum_{c=1}^{k} x_{i,c} \leqslant 1$, and when written as a polynomial, has *coloring* degree 1.

We will rely on the following two technical lemmas in our proof. The first informally shows that $\tilde{\mathbb{E}}^{\otimes k}$ "thinks" that $\sum_c x_{i,c} \geqslant 2$ when the event indicated by h_i, namely "$\sum_c x_{i,c} \leqslant 1$", does not occur. Intuitively, this should clearly hold.

▶ **Lemma 6.** *$\tilde{\mathbb{E}}^{\otimes k}$ satisfies the constraint $(1 - h_i)(\sum_c x_{i,c} - 2) \geqslant 0$. Namely, for every polynomial f with $\mathrm{cdeg}(f) \leqslant \frac{d-2}{2}$, it holds that $\tilde{\mathbb{E}}^{\otimes k}[f^2(1 - h_i)(\sum_c x_{i,c} - 2)] \geqslant 0$.*

The second lemma shows that the linear operator $\tilde{\mathbb{E}}$ satisfies a hypercontractive inequality – that is, the expectations of 4th powers of low-degree polynomials can be upper-bounded in terms of the expectations of their 2nd powers. Readers familiar with Fourier analysis over the hypercube may observe that the "scaling" in our estimate grows as $\exp(O(d \log n))$ in contrast to the $\exp(O(d))$ scaling in the usual hypercontractive inequality over the uniform measure on the Boolean hypercube. However, this worse bound will be sufficient for our purposes.

▶ **Lemma 7** (Hypercontractivity). *For any multilinear f with $\mathrm{cdeg}(f) \leqslant d/4$ satisfying $f = \Pi_G^{\otimes k} f$, we have $\tilde{\mathbb{E}}^{\otimes k}[f^4] \leqslant n^{O(\deg(f))} \cdot \tilde{\mathbb{E}}^{\otimes k}[f^2]^2/(\lambda n^{-O(d)})^{2\deg(f)}$.*

We postpone the proofs of Lemmas 6 and 7 to Sections 2.3 and 2.4, respectively, and finish the proof assuming these two claims. Let f be any polynomial with $\mathrm{cdeg}(f) \leqslant \frac{d}{4}$. We lower bound $\tilde{\mathbb{E}}^{\otimes k}[f^2 \sum_c x_{i,c}]$. Without loss of generality, we can assume that f is multilinear, as if f is not multilinear, then we can reduce it modulo the booleanity constraints. We can also assume that $f = \Pi_G^{\otimes k} f$, as if this does not hold then we write $f = f_1 + f_2$ where $f_1 = \Pi_G^{\otimes k} f_1$ and $\Pi_G^{\otimes k} f_2 = 0$, and then we observe that $\tilde{\mathbb{E}}^{\otimes k}[(f_1+f_2)^2 \sum_c x_{i,c}] = \tilde{\mathbb{E}}^{\otimes k}[f_1^2 \sum_c x_{i,c}]$ (because $f_2 = 0$ is satisfied by $\tilde{\mathbb{E}}^{\otimes k}$ as a constraint) and $\mathrm{cdeg}(f_1) \leqslant \mathrm{cdeg}(f) \leqslant \frac{d}{4}$, as the projection operation can only decrease coloring degree. We note that $\tilde{\mathbb{E}}^{\otimes k}[h_i^2] = \tilde{\mathbb{E}}^{\otimes k}[h_i]$, since $h_i^2 \equiv h_i$ modulo the booleanity constraints $\{x_{i,c}^2 = x_{i,c}\}$. We have that

$$\begin{aligned}
&\tilde{\mathbb{E}}^{\otimes k}[f^2 \sum_c x_{i,c}] = \tilde{\mathbb{E}}^{\otimes k}[f^2 h_i \sum_c x_{i,c}] + \tilde{\mathbb{E}}^{\otimes k}[f^2(1-h_i)\sum_c x_{i,c}] \\
&= \tilde{\mathbb{E}}^{\otimes k}[f^2 h_i^2 \sum_c x_{i,c}^2] + \tilde{\mathbb{E}}^{\otimes k}[f^2(1-h_i)\sum_c x_{i,c}] \text{ (as } h_i^2 \equiv h_i \text{ and } x_{i,c}^2 \equiv x_{i,c}) \\
&\geqslant 0 + \tilde{\mathbb{E}}^{\otimes k}[f^2(1-h_i)\sum_c x_{i,c}] \text{ (by positivity of } \tilde{\mathbb{E}}^{\otimes k}) \\
&\geqslant \tilde{\mathbb{E}}^{\otimes k}[f^2 \cdot 2(1-h_i)] \text{ (by Lemma 6)} \\
&= 2(\tilde{\mathbb{E}}^{\otimes k}[f^2] - \tilde{\mathbb{E}}^{\otimes k}[f^2 h_i]) \ .
\end{aligned}$$

Note that this is well-defined because $\tilde{\mathbb{E}}^{\otimes k}$ is defined on each of the terms in the above inequalities since $\mathrm{cdeg}(f) \leqslant d/4 \leqslant (d-4)/2$ and $\mathrm{cdeg}(h_i) = \mathrm{cdeg}(\sum_c x_{i,c}) = 1$.

Now, we observe that

$$\begin{aligned}
&\tilde{\mathbb{E}}^{\otimes k}[f^2 h_i] \leqslant \sqrt{\tilde{\mathbb{E}}^{\otimes k}[f^4]}\sqrt{\tilde{\mathbb{E}}^{\otimes k}[h_i^2]} \text{ (by Fact 5)} \\
&\leqslant \left(\frac{n^{O(d)}}{\lambda}\right)^{2\deg(f)} \tilde{\mathbb{E}}^{\otimes k}[f^2] \cdot \sqrt{\tilde{\mathbb{E}}^{\otimes k}[h_i^2]} \text{ (by Lemma 7)} \\
&\leqslant \left(\frac{n^{O(d)}}{\lambda}\right)^{2\deg(f)} \tilde{\mathbb{E}}^{\otimes k}[f^2] \cdot \sqrt{\tilde{\mathbb{E}}^{\otimes k}[h_i]} \text{ (since } h_i^2 \equiv h_i) \ .
\end{aligned}$$

Next, we observe that $\tilde{\mathbb{E}}^{\otimes k}[h_i] = \tilde{\mathbb{E}}^{\otimes k}[\prod_c (1-x_{i,c})] + \sum_c \tilde{\mathbb{E}}^{\otimes k}[x_{i,c}\prod_{c' \neq c}(1-x_{i,c'})] = (1 - \tilde{\mathbb{E}}[x_i])^k + (k-1)(\tilde{\mathbb{E}}[x_i](1-\tilde{\mathbb{E}}[x_i])^{k-1}) \leqslant k \cdot e^{-k/k_0}$ using the tensor structure of $\tilde{\mathbb{E}}^{\otimes k}$ and that $\tilde{\mathbb{E}}[x_i] \geqslant \frac{1}{k_0}$. Hence,

$$\begin{aligned}
&\tilde{\mathbb{E}}^{\otimes k}[f^2 h_i] \leqslant \left(\frac{n^{O(d)}}{\lambda}\right)^{2\deg(f)} \tilde{\mathbb{E}}^{\otimes k}[f^2] \cdot k \cdot e^{-k/k_0} \\
&\implies \tilde{\mathbb{E}}^{\otimes k}[f^2 \sum_c x_{i,c}] \geqslant 2\tilde{\mathbb{E}}^{\otimes k}[f^2]\left(1 - k \cdot e^{-k/k_0}\left(\frac{n^{O(d)}}{\lambda}\right)^{2\deg(f)}\right) \ .
\end{aligned}$$

Now, suppose that f is any polynomial with $\deg(f) \leqslant \frac{d}{4}$. This implies that (1) $\tilde{\mathbb{E}}'[f^2 \sum_c x_{i,c}]$ is defined, and (2) $\mathrm{cdeg}(f) \leqslant \frac{d}{4}$, and so we have that (using $\lambda \leqslant 1$)

$$\begin{aligned}
&\tilde{\mathbb{E}}'[f^2 \sum_c x_{i,c}] = \tilde{\mathbb{E}}^{\otimes k}[f^2 \sum_c x_{i,c}] \geqslant 2\tilde{\mathbb{E}}^{\otimes k}[f^2]\left(1 - k \cdot e^{-k/k_0}\left(\frac{n^{O(d)}}{\lambda}\right)^{d/2}\right) \\
&= \tilde{\mathbb{E}}'[f^2] \cdot 2\left(1 - k \cdot e^{-k/k_0}\left(\frac{n^{O(d)}}{\lambda}\right)^{d/2}\right) \ .
\end{aligned}$$

Choosing $k = O(k_0 d \log(n^d/\lambda))$, it follows that $1 - k \cdot e^{-k/k_0} \left(\frac{n^{O(d)}}{\lambda}\right)^{d/2} \geqslant \frac{1}{2}$ and so $\tilde{\mathbb{E}}'[f^2(\sum_c x_{i,c} - 1)] \geqslant 0$ for all f with $\deg(f) \leqslant \frac{d}{4}$. Since $\tilde{\mathbb{E}}'$ is a degree $1 + \frac{d}{2}$ pseudo-expectation, this means that $\tilde{\mathbb{E}}'$ satisfies the constraint $\sum_c x_{i,c} - 1 \geqslant 0$, which finishes the proof.

2.3 Proof of Lemma 6

Let f be any polynomial with $\mathrm{cdeg}(f) \leqslant (d-2)/2$. It suffices to show that $\tilde{\mathbb{E}}^{\otimes k}[f^2(1 - h_i)(\sum_c x_{i,c} - 2)] \geqslant 0$. For $2 \leqslant t \leqslant k$, let $g_i^{(t)} = \prod_{c \leqslant t}(1 - x_{i,c})$, $g_{i,c}^{(t)} = \prod_{c' \neq c, c' \leqslant t}(1 - x_{i,c'})$, and $h_i^{(t)} := \sum_{c \leqslant t} x_{i,c} g_{i,c}^{(t)} + g_i^{(t)}$. We show by induction on t that for each $t \geqslant 0$, it holds that $\tilde{\mathbb{E}}^{\otimes k}[f^2(1 - h_i^{(t)})(\sum_{c \leqslant t} x_{i,c} - 2)] \geqslant 0$ for every f where the coloring degree of f on the first t colors is at most $(d-2)/2$, and $\mathrm{cdeg}(f) \leqslant d/2$.

The base case is when $t = 2$. In this case, we have $1 - h_i^{(t)} = 1 - x_{i,1}(1 - x_{i,2}) - x_{i,2}(1 - x_{i,1}) - (1 - x_{i,1})(1 - x_{i,2}) = x_{i,1}x_{i,2}$, so $\tilde{\mathbb{E}}^{\otimes k}[f^2(1 - h_i^{(t)})(\sum_{c \leqslant t} x_{i,c} - 2)] = \tilde{\mathbb{E}}^{\otimes k}[f^2(x_{i,1}x_{i,2})(x_{i,1} + x_{i,2} - 2)] = \tilde{\mathbb{E}}^{\otimes k}[f^2(2x_{i,1}x_{i,2} - 2x_{i,1}x_{i,2})] = 0$. Note that since f has coloring degree at most $(d-2)/2$ on the first colors, $\tilde{\mathbb{E}}^{\otimes k}$ is always defined on each of these polynomials.

We now show the induction step. We observe that $h_i^{(t+1)} = (1 - x_{i,t+1})h_i^{(t)} + x_{i,t+1} g_i^{(t)}$. Let f be a polynomial that has coloring degree $\leqslant (d-2)/2$ on the first $t+1$ colors, and $\mathrm{cdeg}(f) \leqslant d/2$. We have

$$
\begin{aligned}
&\tilde{\mathbb{E}}^{\otimes k}[f^2(1 - h_i^{(t+1)})(x_{i,t+1} + \sum_{c \leqslant t+1} x_{i,c} - 2)] \\
&= \tilde{\mathbb{E}}^{\otimes k}[f^2\Big((1 - x_{i,t+1})(1 - h_i^{(t)}) + x_{i,t+1}(1 - g_i^{(t)})\Big) \cdot (x_{i,t+1} + \sum_{c \leqslant t} x_{i,c} - 2)] \\
&= \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - x_{i,t+1})(1 - h_i^{(t)})(\sum_{c \leqslant t} x_{i,c} - 2)] + \tilde{\mathbb{E}}^{\otimes k}[f^2 x_{i,t+1}(1 - g_i^{(t)})(1 + \sum_{c \leqslant t} x_{i,c} - 2)] \\
&= \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - x_{i,t+1})^2(1 - h_i^{(t)})(\sum_{c \leqslant t} x_{i,c} - 2)] + \tilde{\mathbb{E}}^{\otimes k}[f^2 x_{i,t+1}^2(1 - g_i^{(t)})(\sum_{c \leqslant t} x_{i,c} - 1)] \ .
\end{aligned}
$$

Since f has coloring degree at most $(d-2)/2$ on the first $t+1$ colors, $f \cdot (1 - x_{i,t+1})$ has coloring degree at most $(d-2)/2$ on the first t colors, and also $\mathrm{cdeg}(f \cdot (1 - x_{i,t+1}))$ is at most $d/2$, as on the $(t+1)$-th color it has degree at most $(d-2)/2 + 1$, and on every other color it is either at most $(d-2)/2$ or $d/2$. So, $\tilde{\mathbb{E}}^{\otimes k}[f^2(1 - x_{i,t+1})^2(1 - h_i^{(t)})(\sum_{c \leqslant t} x_{i,c} - 2)] \geqslant 0$ by the induction hypothesis. We also observe that $f \cdot x_{i,t+1}$ has coloring degree at most $(d-2)/2$ on the first t colors, and has coloring degree at most $d/2$.

It remains to show that $\tilde{\mathbb{E}}^{\otimes k}[f^2(1 - g_i^{(t)})(\sum_{c \leqslant t} x_{i,c} - 1)] \geqslant 0$ for all t and for all f with $\mathrm{cdeg}(f) \leqslant d/2$ and coloring degree at most $(d-2)/2$ in the first t colors. We observe that $g_i^{(t)} x_{i,c} \equiv 0$ for all $c \leqslant t$, and so it suffices to show that $\tilde{\mathbb{E}}^{\otimes k}[f^2 \sum_{c \leqslant t} x_{i,c}] \geqslant \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - g_i^{(t)})]$. We do this by induction on t. In the base case, we have $\tilde{\mathbb{E}}^{\otimes k}[f^2 x_{i,1}] = \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - (1 - x_{i,1}))]$. For the induction step, we have

$$
\begin{aligned}
&\tilde{\mathbb{E}}^{\otimes k}[f^2(x_{i,t+1} + \sum_{c \leqslant t} x_{i,c})] \geqslant \tilde{\mathbb{E}}^{\otimes k}[f^2 x_{i,t+1}] + \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - g_i^{(t)})] \\
&\geqslant \tilde{\mathbb{E}}^{\otimes k}[f^2 x_{i,t+1} g_i^{(t)}] + \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - g_i^{(t)})] \\
&= \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - (1 - x_{i,t+1})g_i^{(t)})] = \tilde{\mathbb{E}}^{\otimes k}[f^2(1 - g_i^{(t+1)})] \ ,
\end{aligned}
$$

where we use the fact that $f x_{i,t+1} g_i^{(t)}$ has coloring degree $\leqslant d/2$ since f has coloring degree at most $(d-2)/2$ in the first $t+1$ colors, and that $(g_i^{(t)})^2 \equiv g_i^{(t)}$ modulo the hypercube constraints. This finishes the proof.

2.4 Proof of Lemma 7: hypercontractivity

Let f be a multilinear polynomial with $\mathrm{cdeg}(f) \leqslant d/4$ with $\Pi_G^{\otimes k} f = f$. Suppose that:

$$\tilde{\mathbb{E}}^{\otimes k}[f^2] \geqslant \lambda_1 \|f\|_2^2 \ , \tag{2.1}$$

$$\tilde{\mathbb{E}}^{\otimes k}[f^4] \leqslant \lambda_2 \left\|f^2\right\|_2^2 \ , \tag{2.2}$$

$$\left\|f^2\right\|_2^2 \leqslant C \|f\|_2^4 \ . \tag{2.3}$$

Then it follows that $\tilde{\mathbb{E}}^{\otimes k}[f^4] \leqslant \lambda_2 \left\|f^2\right\|_2^2 \leqslant \lambda_2 C \|f\|_2^4 \leqslant \frac{\lambda_2 C}{\lambda_1^2} \tilde{\mathbb{E}}^{\otimes k}[f^2]^2$.

Equation (2.1) follows from the following lemma with $\lambda_1 := (\lambda n^{-O(d)})^{\deg(f)}$.

▶ **Lemma 8** (Eigenvalue lower bound for $\tilde{\mathbb{E}}^{\otimes k}$). *Suppose that for any multilinear g with $\deg(g) \leqslant d/2$, $\tilde{\mathbb{E}}[g^2] \geqslant \lambda \|\Pi_G g\|_2^2$. Then for any multilinear f of coloring degree $\leqslant d/2$, it holds that $\tilde{\mathbb{E}}^{\otimes k}[f^2] \geqslant (\lambda n^{-O(d)})^{\deg(f)} \cdot \left\|\Pi_G^{\otimes k} f\right\|_2^2$.*

We postpone the proof of Lemma 8 for now, and finish the proof of Lemma 7.

Let $g = \sum_m g_m \cdot m$ be a polynomial of degree $\deg(g)$ with $\mathrm{cdeg}(g) \leqslant d/2$, where m is a monomial and g_m is the coefficient of g for the monomial m. Since $\tilde{\mathbb{E}}^{\otimes k}$ satisfies the booleanity constraints, it follows that $\tilde{\mathbb{E}}[m] \leqslant 1$ for all monomials m. Hence, $\tilde{\mathbb{E}}[g^2] \leqslant \sum_{m_1,m_2} |g_{m_1} g_{m_2}| = (\sum_m |g_m|)^2 \leqslant (nk)^{O(\deg(g))} \|g\|_2^2$, by Cauchy-Schwarz, as g is supported on at most $(nk)^{O(\deg(g))}$ distinct monomials. Since $(nk)^{O(\deg(g))} \leqslant n^{O(\deg(g))}$ as $k < n$, it follows that $\tilde{\mathbb{E}}^{\otimes k}[g^2] \leqslant n^{O(\deg(g))} \|g\|_2^2$, and so (setting $g = f^2$) Equation (2.2) holds with $\lambda_2 := n^{O(\deg(f))}$.

Finally, for a polynomial g and monomial m let g_m be the coefficient of g on m. For any m of degree $\leqslant 2\deg(f)$, we have that $f_m^2 = \sum_{m_1,m_2 : m_1 \cdot m_2 = m} f_{m_1} f_{m_2}$. We observe that this is equal to $\langle v^{(m)}, f \rangle$, where $v^{(m)}$ is the vector defined as $v_{m_2}^{(m)} \stackrel{\text{def}}{=} f_{m_1}$ where $m_1 \cdot m_2 = m$ (and is 0 if no such m_1 exists). It follows that $\left\|v^{(m)}\right\|_2 \leqslant \|f\|_2$, and hence that $\left|\langle v^{(m)}, f \rangle\right| \leqslant \max_{v : \|v\|_2 \leqslant \|f\|_2} |\langle v, f \rangle| = \|f\|_2^2$. Hence, $\left|f_m^2\right|^2 \leqslant \|f\|_2^4$, and so $\left\|f^2\right\|_2^2 = \sum_{m : \deg(m) \leqslant 2\deg(f)} \left|f_m^2\right| \leqslant (nk)^{O(\deg(f))} \|f\|_2^4 = n^{O(\deg(f))} \|f\|_4^2$, and so Equation (2.3) holds with $C := n^{O(\deg(f))}$.

Combining, we conclude that $\tilde{\mathbb{E}}^{\otimes k}[f^4] \leqslant \tilde{\mathbb{E}}^{\otimes k}[f^2]^2 / (\lambda n^{-O(d)})^{2\deg(f)}$, which finishes the proof.

2.4.1 Proof of Lemma 8: eigenvalue lower bound for $\tilde{\mathbb{E}}^{\otimes k}$

Proof outline. The proof proceeds in three steps. First, we show that the moment matrix of the independent set pseudo-expectation $\tilde{\mathbb{E}}$, when written in a basis so that the constant polynomial 1 is an eigenvector, has an eigenvalue lower bound of $\lambda n^{-O(d)}$. To show that this property implies the desired eigenvalue lower bound, we observe that any f of total degree $\leqslant d$ is a linear combination of monomials that use at most $\deg(f)$ colors. Further, (the coefficient vector of) each such monomial is a linear combination of tensor products of eigenvectors of the $\tilde{\mathbb{E}}$ that use a "non-1" eigenvector in at most $\deg(f)$ modes of the tensor and thus is in the span of eigenvectors of $\tilde{\mathbb{E}}^{\otimes k}$ (in the new basis) with eigenvalue at least $(\lambda n^{-O(d)})^{\deg(f)}$. This reasoning immediately implies that f, when written in the chosen basis, has the desired eigenvalue lower bound. To finish the proof, we argue that the change of basis does not modify $\|f\|_2$ by too much.

We now proceed with implementing the above proof plan. For every $S \subseteq [n]$ with $|S| \leqslant d$, recall that we can express any multilinear polynomial g with degree $\leqslant d$ as a linear combination of the monomials $x_S \stackrel{\text{def}}{=} \prod_{i \in S} x_i$. Let g_S be the coefficient of g on the monomial S, so that $g = \sum_{|S| \leqslant d} g_S x_S$. Let e_S be the S-th standard basis vector in $\mathbb{R}^{\binom{n}{\leqslant d}}$. Then g (as a vector of coefficients) is $\sum_S g_S e_S$. For $S \neq \emptyset$, define e'_S as $e_S - \tilde{\mathbb{E}}[x_S] \cdot e_\emptyset$. We can write g uniquely in the e'_S basis as $g = \sum_S g'_S e'_S$, where $g'_S = g_S$ for $S \neq \emptyset$, and $g'_\emptyset = g_\emptyset + \sum_{S \neq \emptyset} g_S \tilde{\mathbb{E}}[x_S] = \tilde{\mathbb{E}}[g]$. Note that, if we let $x'_S := x_S - \tilde{\mathbb{E}}[x_S]$ for $S \neq \emptyset$ and $x'_\emptyset := x_\emptyset = 1$, then $g = \sum_S g'_S x'_S$ as a polynomial.

Let $\mathcal{M}$ be the moment matrix of $\tilde{\mathbb{E}}$ in the x' basis. This matrix is indexed by sets $S, S' \subseteq [n]$ with $|S|, |S'| \leqslant d/2$, and $\mathcal{M}(S, S') = \tilde{\mathbb{E}}[x'_S x'_{S'}]$, which is equal to $\tilde{\mathbb{E}}[(x_S - \tilde{\mathbb{E}}[x_S])(x_{S'} - \tilde{\mathbb{E}}[x_{S'}])] = \tilde{\mathbb{E}}[x_S x_{S'}] - \tilde{\mathbb{E}}[x_S]\tilde{\mathbb{E}}[x_{S'}]$ if $S, S' \neq \emptyset$, equal to 0 if exactly one of S, S' is $\emptyset$, and equal to 1 if $S = S' = \emptyset$. This implies that $e'_\emptyset$ is an eigenvector of $\mathcal{M}$ with eigenvalue 1. We also observe that if g has degree $\leqslant d/2$ and g' is the coefficient vector of g in the e' basis, then $\tilde{\mathbb{E}}[g^2] = g'^\top \mathcal{M} g'$.

We now prove the following eigenvalue lower bound on $\mathcal{M}$.

▷ Claim 9. $\mathcal{M} \succeq \lambda n^{-O(d)} \Pi_G$.

Proof. Let S with $|S| \leqslant d/2$ be a set that is not an independent set in G, i.e. that $\Pi_G e_S = 0$. We observe that $\mathcal{M} e'_S = 0$. Indeed, the T-th entry of $\mathcal{M} e'_S$ is $\mathcal{M}(T, S) = \tilde{\mathbb{E}}[x'_T x'_S] = \tilde{\mathbb{E}}[x_T x_S] - \tilde{\mathbb{E}}[x_T]\tilde{\mathbb{E}}[x_S] = 0 - 0 = 0$ for $T \neq \emptyset$, and is 0 if $T = \emptyset$ because $M(\emptyset, S) = 0$ for $S \neq \emptyset$.

Now, let $g' = \sum_{S : |S| \leqslant d} g'_S e'_S$ be arbitrary. By the above, without loss of generality we may assume that $g'_S = 0$ for all S that is not an independent set in G. Let g be the corresponding polynomial in the x basis, so that $g = \sum_S g_S x_S$, where $g_\emptyset = g'_\emptyset - \sum_{S \neq \emptyset} g'_S \tilde{\mathbb{E}}[x_S]$ and $g_S = g'_S$ for all $S \neq \emptyset$. Notice that $\tilde{\mathbb{E}}[g] = g'_\emptyset$. We observe that $\Pi_G g = g$, as $g_S = g'_S = 0$ for all S that is not an independent set in G. Now, we have that $g'^\top M g' = \tilde{\mathbb{E}}[g^2] \geqslant \lambda \left\|\Pi_G g\right\|_2^2 = \lambda \left\|g\right\|_2^2$, by our eigenvalue lower bound assumption on $\tilde{\mathbb{E}}$.

It remains to relate $\left\|g\right\|_2^2$ and $\left\|g'\right\|_2^2$. We have that $\left\|g'\right\|_2^2 = \sum_{|S| \leqslant d/2} g'^2_S = \tilde{\mathbb{E}}[g]^2 + \sum_{S \neq \emptyset} g'^2_S \leqslant \tilde{\mathbb{E}}[g]^2 + \left\|g\right\|_2^2 \leqslant \tilde{\mathbb{E}}[g^2] + \left\|g\right\|_2^2 \leqslant (n^{O(d)} + 1)\left\|g\right\|_2^2$, as $\tilde{\mathbb{E}}[g^2] \leqslant n^{O(d)} \left\|g\right\|_2^2$ since $0 \leqslant \tilde{\mathbb{E}}[x_S x_T] \leqslant 1$ for all S, T, and there are at most $n^{O(d)}$ such pairs. Hence, $g'^\top M g' \geqslant \lambda n^{-O(d)} \left\|g'\right\|_2^2$ when $g' = \Pi_G g'$, and so $M \succeq \lambda n^{-O(d)} \Pi_G$. ◁

We have already shown that $e'_\emptyset$ is an eigenvector of $\mathcal{M}$ with eigenvalue 1, and that the zero eigenvectors of $\mathcal{M}$ are the vectors e'_S where S is not an independent set in G. Let $f_0 = e'_\emptyset, f_1, \ldots, f_r$ be the eigenvectors of $\mathcal{M}$ with nonzero eigenvalues $\lambda_0 = 1, \lambda_1, \ldots, \lambda_t$, where $\lambda_i \geqslant \lambda n^{-O(d)}$ for $1 \leqslant i \leqslant t$. Let $\mathcal{M}^{\otimes k}$ be the k-th tensor of $\mathcal{M}$. Let $f_i^{(c)}$ denote the i-th eigenvector in the c-th component of the tensor. The eigenvectors of $\mathcal{M}^{\otimes k}$ are the vectors $\bigotimes_{c=1}^k f_{i_c}^{(c)}$. We additionally observe that $\mathcal{V}^{(c)} \overset{\text{def}}{=} \textbf{Span}\left(f_i^{(c)} : i > 0\right) = \textbf{Span}\left(e'^{(c)}_S : |S| > 0, \Pi_G e_S = e_S\right)$, as $f_0^{(c)} = e'^{(c)}_\emptyset$.

Let f be a multilinear polynomial with $\text{cdeg}(f) \leqslant d/2$ in the variables $\{x_{i,c}\}_{i \in [n], c \in [k]}$. That is, f is a vector in $\textbf{Span}\left(\bigotimes_{c=1}^k e_{S_c}^{(c)} : |S_c| \leqslant d/2 \ \forall c \in [k]\right)$, where $e_S^{(c)}$ denotes the S-th standard basis vector in the c-th component of the tensor. As before, we can write f as a vector f' in the e' basis, so $f' = \sum_{(S_1, \ldots, S_k) : |S_c| \leqslant d/2 \ \forall c \in [k]} f'_{(S_1, \ldots, S_k)} \bigotimes_{c=1}^k e'^{(c)}_{S_c}$. We again observe that $\tilde{\mathbb{E}}^{\otimes k}[f^2] = f'^\top \mathcal{M}^{\otimes k} f'$, because the $((S_1, \ldots, S_k), (T_1, \ldots, T_k))$-th entry of $\mathcal{M}^{\otimes k}$ is exactly $\prod_{c=1}^k \tilde{\mathbb{E}}[x'_{S_c} x'_{T_c}] = \tilde{\mathbb{E}}^{\otimes k}[\prod_{c=1}^k x'_{S_c} x'_{T_c}]$. Note that by the structure of the zero eigenvectors of $\mathcal{M}$, if f satisfies $\Pi_G^{\otimes k} f = 0$, then f is an eigenvector of $\mathcal{M}^{\otimes k}$ with eigenvalue 0. In particular, without loss of generality we can assume that $f = \Pi_G^{\otimes k} f$, as by the above we can discard the component of f in the kernel of $\Pi_G^{\otimes k}$.

Let $(S_1,\dots,S_k)$ be such that $f'_{(S_1,\dots,S_k)} \neq 0$. We must have $S_c = \emptyset$ for all but at most $\deg(f)$ of the c's. This is because f has degree $\deg(f)$, and so in particular every monomial in f can only use at most $\deg(f)$ distinct colors. This shows that $f' \in \mathcal{V} := \textbf{Span}\left(\bigotimes_{c\in C}\mathcal{V}^{(c)}\bigotimes_{c\notin C} e'^{(c)}_{\emptyset} : C \subseteq [k], |C| \leqslant \deg(f)\right)$. We observe that $\mathcal{V}$ is the span of eigenvectors of $\mathcal{M}$ of the form $\bigotimes_{c\in C} f^{(c)}_{i_c}\bigotimes_{c\notin C} e'^{(c)}_{\emptyset}$ for $|C|\leqslant\deg(f)$. Since each of these vectors is an eigenvector with eigenvalue at least $(\lambda n^{-O(d)})^{|C|}\cdot 1^{k-|C|} \geqslant (\lambda n^{-O(d)})^{\deg(f)}$, it follows that $f'^{\top}\mathcal{M}^{\otimes k} f' \geqslant (\lambda n^{-O(d)})^{\deg(f)}\|f'\|_2^2$. Thus, $\tilde{\mathbb{E}}^{\otimes k}[f^2] \geqslant (\lambda n^{-O(d)})^{\deg(f)}\|f'\|_2^2$.

It remains to relate $\|f'\|_2^2$ and $\|f\|_2^2$. Fix $(S_1,\dots,S_k)$ with $|S_c| \leqslant d/2$. Let $(T_1,\dots,T_k)$ with $|T_c| \leqslant d/2$. We say that $(T_1,\dots,T_k)$ *extends* $(S_1,\dots,S_k)$ if for every c, either $T_c = S_c$ or $T_c \neq \emptyset$ and $S_c = \emptyset$. The *parity* of the extension is the parity of the number of c where $T_c \neq \emptyset$ and $S_c = \emptyset$. We observe that $f_{(S_1,\dots,S_k)} = \sum_{(T_1,\dots,T_k) \text{ extending } (S_1,\dots,S_k)} (\text{parity of extension}) \cdot f'_{(T_1,\dots,T_k)}$. This is because $e'_{(T_1,\dots,T_k)} = \bigotimes_{c=1}^{k} e'_{T_c} = \bigotimes_{c=1}^{k}(e_{T_c} - e_{\emptyset})$. We thus see that $\|f\|_1 \leqslant \sum_{(T_1,\dots,T_k)}\left|f'_{(T_1,\dots,T_k)}\right| \cdot n_{(T_1,\dots,T_k)}$, where $n_{(T_1,\dots,T_k)}$ is the number of $(S_1,\dots,S_k)$ that $(T_1,\dots,T_k)$ extends. We have shown that if $f'_{(T_1,\dots,T_k)} \neq 0$ then it must be the case that $T_c \neq \emptyset$ for at most $\deg(f)$ of the c's. Hence, such $(T_1,\dots,T_k)$ can only extend at most $2^{\deg(f)}$ of the $(S_1,\dots,S_k)$'s, as each of the $(S_1,\dots,S_k)$'s is obtained by changing a subset of the T_c's to be empty. Hence, $\|f\|_1 \leqslant 2^{\deg(f)}\|f'\|_1$. Since f' has at most $(nk)^{\deg(f)} \leqslant n^{2\deg(f)}$ nonzero coefficients, we get that $\|f\|_2 \leqslant n^{\deg(f)}\cdot 2^{\deg(f)}\|f'\|_2$, and so we conclude that $\|f\|_2^2 \leqslant n^{O(\deg(f))}\|f'\|_2^2$, which finishes the proof.

3 Proof of Theorem 1: coloring lower bound

We now prove Theorem 1 (restated below in the language of pseudo-expectations) from Theorem 2. In this section, we assume familiarity with the planted clique pseudo-expectation of [12].

▶ **Theorem** (Theorem 1, restated). *For sufficiently large n, for any $\varepsilon \in (\Omega(\sqrt{\frac{1}{\log n}}), \frac{1}{2})$, with probability $1 - 1/\operatorname{poly}(n)$ over the draw of $G \sim G(n, 1/2)$, there is a degree $d = O(\varepsilon^2 \log n)$ coloring pseudo-expectation $\tilde{\mathbb{E}}$ using $k = n^{\frac{1}{2}+\varepsilon}$ colors.*

We begin by recalling the main theorem of [12].

▶ **Theorem 10** ([12]). *There is an absolute constant C such that for n sufficiently large, $C/\sqrt{\log n} \leqslant \varepsilon < \frac{1}{2}$, $\omega = n^{\frac{1}{2}-\varepsilon}$, and $d = (\varepsilon/C)^2 \log n$, with probability $1 - 1/\operatorname{poly}(n)$ over $G \sim G(n,1/2)$, the operator $\tilde{\mathbb{E}}_G$ defined in [12] satisfies:*

1. $\tilde{\mathbb{E}}_G[1] = 1 \pm n^{-\Omega(\varepsilon)}$,
2. $\tilde{\mathbb{E}}_G[\sum_i x_i] = \omega(1 \pm n^{-\Omega(\varepsilon)})$,
3. $\tilde{\mathbb{E}}_G[x_S] = 0$ *for all* $|S| \leqslant d$ *that is not a clique in* G,
4. $\tilde{\mathbb{E}}_G[f^2] \geqslant \lambda \|\Pi'_G f\|_2^2$ *where* $\lambda = \Omega\left(\left(\frac{\omega}{n}\right)^{d+1}\right)$ *and* Π'_G *is the projection onto* x_S *for* S *a clique in* G.

We first observe that if $G \sim G(n,1/2)$, then the complement graph $\bar{G} \sim G(n,1/2)$ also, and moreover $\tilde{\mathbb{E}}_G$ will satisfy the independent set constraints as $\tilde{\mathbb{E}}_G[x_S] = 0$ for S that is not a clique in G, which is equivalent to S not being an independent set in $\bar{G}$. We also note that $\Pi'_G = \Pi_{\bar{G}}$, and that the final pseudo-expectation is obtained by setting $\tilde{\mathbb{E}}[x_S] := \tilde{\mathbb{E}}_G[x_S]/\tilde{\mathbb{E}}_G[1]$; this is done so that the normalization condition $\tilde{\mathbb{E}}[1] = 1$ is satisfied.

We thus see that $\tilde{\mathbb{E}}$ satisfies the second additional condition of Theorem 2. Hence, in order to apply Theorem 2 to conclude Theorem 1, it suffices to argue that with high probability over G, it holds that $\tilde{\mathbb{E}}_G[x_i] \geqslant \frac{\omega}{n}(1 - n^{-\Omega(\varepsilon)})$ for all i. Indeed, if this holds then we have $\tilde{\mathbb{E}}[x_i] \geqslant \frac{\omega}{n}(1 - n^{-\Omega(\varepsilon)})$ also, and then we can apply Theorem 2 with $k = \frac{n}{\omega}(1 + n^{-\Omega(\varepsilon)})$ which finishes the proof. Thus, it remains to prove the following claim.

▷ Claim 11. For each i, $\tilde{\mathbb{E}}_G[x_i] \geqslant \frac{\omega}{n} \cdot (1 - n^{-\Omega(\varepsilon)})$ with probability $1 - n^{-\log n}$.

Proof. We have that $\tilde{\mathbb{E}}_G[x_i] := \sum_{T \subseteq \binom{[n]}{2} : |V(T)| \leqslant \tau} \left(\frac{\omega}{n}\right)^{|V(T) \cup \{i\}|} \chi_T(G)$, where $\tau \leqslant (\varepsilon/C)\log n$. The $T = \emptyset$ term always contributes $\frac{\omega}{n}$. The other terms all have $|V(T)| \geqslant 2$. Let H_1 be the set of T such that $i \in V(T)$, and let H_2 be the set of T such that $i \notin V(T)$. Let $H_1^{(t)}$ be the set of $T \in H_1$ with $|V(T)| = t$, and similarly for $H_2^{(t)}$. Each set $H_1^{(t)}$ can be partitioned into families $\{\mathcal{T}_{1,r}^{(t)}\}_{r=1}^{p_{1,t}}$ where T and T' are in the same family if there is a permutation $\sigma \colon [n] \to [n]$ such that $T = \sigma(T')$, or equivalently if T and T' are isomorphic. Similarly, each set $H_2^{(t)}$ can be partitioned into families $\{\mathcal{T}_{2,r}^{(t)}\}_{r=1}^{p_{2,t}}$.

We thus have

$$|\tilde{\mathbb{E}}_G[x_i] - \frac{\omega}{n}| \leqslant \sum_{t=2}^{\tau} \left[\left(\frac{\omega}{n}\right)^t \sum_{r=1}^{p_{1,t}} \Big| \sum_{T \in \mathcal{T}_{1,r}^{(t)}} \chi_T(G) \Big| + \left(\frac{\omega}{n}\right)^{t+1} \sum_{r=1}^{p_{2,t}} \Big| \sum_{T \in \mathcal{T}_{2,r}^{(t)}} \chi_T(G) \Big| \right] .$$

▶ **Lemma 12.** *Let $\mathcal{T}$ be a family of subsets of $\binom{[n]}{2}$ such that $|V(T)| = t$ for every $T \in \mathcal{T}$, and for every $T, T' \in \mathcal{T}$, there exists $\sigma \colon [n] \to [n]$ such that $T = \sigma(T')$. Let $S = \cap_{T \in \mathcal{T}} V(T)$. Then for every $s \geqslant 0$ and even ℓ,*

$$\Pr_{G \sim G(n,1/2)} \left[\Big| \sum_{T \in \mathcal{T}} \chi_T(G) \Big| \leqslant s \right] \geqslant 1 - \frac{n^{(t-|S|)\ell/2} \cdot (t\ell)^{t\ell}}{s^\ell} .$$

We postpone the proof of Lemma 12 to the end of the section, and now use it to finish the proof of Claim 11. Applying Lemma 12 with $\ell = (\log n)^2$, we get

$$\Big| \sum_{T \in \mathcal{T}_{1,r}^{(t)}} \chi_T(G) \Big| \leqslant n^{(t-1)/2} (\log n)^{3t} \text{ with probability } \geqslant 1 - 2^{-t\log^2 n(\log\log n - \log t)}$$
$$\Big| \sum_{T \in \mathcal{T}_{2,r}^{(t)}} \chi_T(G) \Big| \leqslant n^{t/2} (\log n)^{3t} \text{ with probability } \geqslant 1 - 2^{-t\log^2 n(\log\log n - \log t)} .$$

We observe that $p_{1,t}$ and $p_{2,t}$ are both at most 2^{t^2}, as an equivalence class with t vertices is uniquely determined by a graph on t vertices. By union bound, we see that the above holds for all equivalence classes $\mathcal{T}_{1,r}^{(t)}$ and $\mathcal{T}_{2,r}^{(t)}$ with probability at least $1 - 2\sum_{t=2}^{\tau} 2^{t^2 - t\log^2 n(\log\log n - \log t)}$. Since $t \leqslant \tau \leqslant (\varepsilon/C)\log n$, it follows that

$$\begin{aligned}
&\sum_{t=2}^{\tau} 2^{t^2 - t\log^2 n(\log\log n - \log t)} = \sum_{t=2}^{\tau} 2^{t(t - \log^2 n(\log\log n - \log t))} \\
&\leqslant \sum_{t=2}^{\tau} 2^{t(\frac{\varepsilon}{C}\log n - (\log^2 n)(\log\log n - \log\frac{\varepsilon}{C} - \log\log n))} \\
&\leqslant \tau \cdot 2^{2\log n \cdot (\frac{\varepsilon}{C} - \log\frac{C}{\varepsilon} \cdot \log n)} \leqslant n^{-\log n} ,
\end{aligned}$$

as $\frac{C}{\varepsilon} \geqslant C \geqslant 16$. Thus, with probability at least $1 - n^{-\log n}$, we have

$$\begin{aligned}
&|\tilde{\mathbb{E}}_G[x_i] - \frac{\omega}{n}| \leqslant \sum_{t=2}^{\tau} \left[\left(\frac{\omega}{n}\right)^t 2^{t^2} n^{(t-1)/2} (\log n)^{3t} + \left(\frac{\omega}{n}\right)^{t+1} 2^{t^2} n^{t/2} (\log n)^{3t} \right] \\
&\leqslant \frac{2}{\sqrt{n}} \sum_{t=2}^{\tau} \left(\frac{\omega}{n}\right)^t 2^{t^2} (\log n)^{3t} n^{t/2} = 2\left(\frac{\omega}{n}\right) \sum_{t=2}^{\tau} n^{(t-1)/2} n^{-(t-1)\varepsilon} 2^{t^2} (\log n)^{3t} n^{t/2} n^{-1/2}
\end{aligned}$$

$$\begin{aligned}
&\leqslant 2\left(\frac{\omega}{n}\right)\sum_{t=2}^{\tau} n^{-(t-1)\varepsilon}2^{t^2}(\log n)^{3t} \leqslant \left(\frac{\omega}{n}\right)\cdot \max_{2\leqslant t\leqslant \tau} 2n^{-(t-1)\varepsilon}2^{t^2}(\log n)^{3t+1}\\
&\leqslant \left(\frac{\omega}{n}\right)\cdot 2n^{\varepsilon}(\log n)\max_{2\leqslant t\leqslant \tau}(n^{-\varepsilon}2^{\tau}(\log n)^3)^t \leqslant \left(\frac{\omega}{n}\right)\cdot n^{\varepsilon}n^{2\varepsilon/K}\max_{2\leqslant t\leqslant \tau}(n^{-\varepsilon}n^{\varepsilon/C}\cdot n^{3\varepsilon/K})^t\\
&\leqslant \left(\frac{\omega}{n}\right)\cdot n^{\varepsilon}n^{2\varepsilon/K}n^{-2\varepsilon(1-1/C-3/K)} = \left(\frac{\omega}{n}\right)\cdot n^{-\varepsilon(1-2/C-8/K)} \leqslant \left(\frac{\omega}{n}\right)\cdot n^{-\varepsilon/2}\ ,
\end{aligned}$$

as $\varepsilon \geqslant C/\sqrt{\log n} \geqslant K\log\log n/\log n$ for $K \geqslant 32$ and $\tau \leqslant (\varepsilon/C)\log n$. Hence, with probability $1 - 1/n^{\log n}$, we have that $\tilde{\mathbb{E}}_G[x_i] = \frac{\omega}{n}(1 \pm n^{-\varepsilon/2})$, which completes the proof. ◁

Proof of Lemma 12. Let $\ell \in \mathbb{N}$ be even. We have that

$$\mathop{\mathbb{E}}_{G\sim G(n,1/2)} \big|\sum_{T\in\mathcal{T}}\chi_T(G)\big|^{\ell} = \mathop{\mathbb{E}}_{G\sim G(n,1/2)}\big(\sum_{T\in\mathcal{T}}\chi_T(G)\big)^{\ell} = \sum_{T_1,\ldots,T_\ell\in\mathcal{T}}\ \mathop{\mathbb{E}}_{G\sim G(n,1/2)}\prod_{i=1}^{\ell}\chi_{T_i}(G)\ .$$

We have that $\mathbb{E}_{G\sim G(n,1/2)}\prod_{i=1}^{\ell}\chi_{T_i}(G) = 1$ iff $\bigoplus_{i=1}^{\ell} T_i = \emptyset$, that is, every edge in the multiset $\cup_{i=1}^{\ell} T_i$ appears an even number of times, and otherwise the term is 0. Since every edge in the multiset appears an even number of times, every vertex also appears an even number of times in $\cup_{i=1}^{\ell} V(T_i)$, and hence every vertex appears at least twice. Since $S \subseteq V(T_i)$ for all i, every vertex in S appears exactly ℓ times. So, the number of distinct vertices in $\cup_{i=1}^{\ell}(V(T_i)\setminus S)$ is at most $(t-|S|)\cdot\ell/2$. Each tuple $(T_1,\ldots,T_\ell)$ with this property can thus be chosen by (1) selecting $(t-|S|)\cdot\ell/2$ distinct vertices S' (at most $n^{(t-|S|)\ell/2}$ choices), and then (2) choosing injections $\sigma_i\colon V(T)\to S'$ and setting $T_i = \sigma_i(T)$, where $T\in\mathcal{T}$ is an arbitrary fixed element (at most $(|S'|^t)^{\ell} \leqslant (t\ell)^{t\ell}$ choices). Thus, we get $\mathbb{E}_{G\sim G(n,1/2)}\big|\sum_{T\in\mathcal{T}}\chi_T(G)\big|^{\ell} \leqslant n^{(t-|S|)\ell/2}(t\ell)^{t\ell}$. By Markov's inequality, it follows that $\Pr_{G\sim G(n,1/2)}\left[\big|\sum_{T\in\mathcal{T}}\chi_T(G)\big| > s\right] = \Pr_{G\sim G(n,1/2)}\left[\big|\sum_{T\in\mathcal{T}}\chi_T(G)\big|^{\ell} > s^{\ell}\right] \leqslant \frac{n^{(t-|S|)\ell/2}(t\ell)^{t\ell}}{s^{\ell}}$, which completes the proof. ◀

References

1 Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *J. ACM*, 62(5):Art. 42, 25, 2015. doi:10.1145/2775105.

2 Sanjeev Arora, Béla Bollobás, László Lovász, and Iannis Tourlakis. Proving integrality gaps without knowing the linear program. *Theory Comput.*, 2:19–51, 2006. doi:10.4086/toc.2006.v002a002.

3 Sanjeev Arora, Eden Chlamtac, and Moses Charikar. New approximation guarantee for chromatic number. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 215–224. ACM, New York, 2006. doi:10.1145/1132516.1132548.

4 Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):Art. 5, 37, 2009. doi:10.1145/1502793.1502794.

5 Mitali Bafna, Boaz Barak, Pravesh Kothari, Tselil Schramm, and David Steurer. Playing unique games on certified small-set expanders. *CoRR*, abs/2006.09969, 2020. arXiv:2006.09969.

6 Ainesh Bakshi, Ilias Diakonikolas, He Jia, Daniel M. Kane, Pravesh K. Kothari, and Santosh S. Vempala. Robustly learning mixtures of k arbitrary gaussians. *CoRR*, abs/2012.02119, 2020. arXiv:2012.02119.

7 Ainesh Bakshi and Pravesh Kothari. List-decodable subspace recovery via sum-of-squares. *CoRR*, abs/2002.05139, 2020. arXiv:2002.05139.

8 Ainesh Bakshi and Pravesh Kothari. Outlier-robust clustering of non-spherical mixtures, 2020. arXiv:2005.02970.

9 Jess Banks, Robert Kleinberg, and Cristopher Moore. The lovász theta function for random regular graphs and community detection in the hard regime. In *APPROX-RANDOM*, volume 81 of *LIPIcs*, pages 28:1–28:22. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

10 Jess Banks, Robert Kleinberg, and Cristopher Moore. The lovász theta function for random regular graphs and community detection in the hard regime. *SIAM J. Comput.*, 48(3):1098–1119, 2019. `doi:10.1137/18M1180396`.

11 Boaz Barak, Siu On Chan, and Pravesh K. Kothari. Sum of squares lower bounds from pairwise independence [extended abstract]. In *STOC'15—Proceedings of the 2015 ACM Symposium on Theory of Computing*, pages 97–106. ACM, New York, 2015.

12 Boaz Barak, Samuel B. Hopkins, Jonathan A. Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *FOCS*, pages 428–437. IEEE Computer Society, 2016.

13 Boaz Barak, Pravesh K. Kothari, and David Steurer. Quantum entanglement, sum of squares, and the log rank conjecture. In *STOC*, pages 975–988. ACM, 2017.

14 Boaz Barak and David Steurer. Proofs, beliefs, and algorithms through the lens of sum-of-squares, 2016. Lecture notes in preparation, available on `http://sumofsquares.org`.

15 Siavosh Benabbas, Siu On Chan, Konstantinos Georgiou, and Avner Magen. Tight gaps for vertex cover in the sherali-adams SDP hierarchy. In *FSTTCS*, volume 13 of *LIPIcs*, pages 41–54. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

16 Siavosh Benabbas, Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. SDP gaps from pairwise independence. *Theory of Computing*, 8(1):269–289, 2012.

17 Siavosh Benabbas and Avner Magen. Extending SDP integrality gaps to sherali-adams with applications to quadratic programming and maxcutgain. In *IPCO*, volume 6080 of *Lecture Notes in Computer Science*, pages 299–312. Springer, 2010.

18 Aditya Bhaskara, Moses Charikar, Aravindan Vijayaraghavan, Venkatesan Guruswami, and Yuan Zhou. Polynomial integrality gaps for strong SDP relaxations of densest k-subgraph. In *SODA*, pages 388–405. SIAM, 2012.

19 B. Bollobás. The chromatic number of random graphs. *Combinatorica*, 8(1):49–55, 1988. `doi:10.1007/BF02122551`.

20 Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Integrality gaps for Sherali-Adams relaxations. In *STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing*, pages 283–292. ACM, New York, 2009.

21 Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *FOCS*, pages 691–701. IEEE Computer Society, 2007.

22 Eden Chlamtac and Madhur Tulsiani. Convex relaxations and integrality gaps. In *Handbook on semidefinite, conic and polynomial optimization*, volume 166 of *Internat. Ser. Oper. Res. Management Sci.*, pages 139–169. Springer, New York, 2012. `doi:10.1007/978-1-4614-0769-0_6`.

23 Amin Coja-Oghlan. The lovász number of random graphs. *Comb. Probab. Comput.*, 14(4):439–465, 2005. `doi:10.1017/S0963548305006826`.

24 Ilias Diakonikolas, Samuel Hopkins, Daniel Kane, and Sushrut Karmalkar. Robustly learning any clusterable mixture of gaussians. *Personal Communication*, 2020.

25 Tommaso d'Orsi, Pravesh K. Kothari, Gleb Novikov, and David Steurer. Sparse PCA: algorithms, adversarial perturbations and certificates. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 553–564. IEEE, 2020. `doi:10.1109/FOCS46700.2020.00058`.

26 Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019. `doi:10.1561/0400000086`.

27 Mrinal Kanti Ghosh and Madhur Tulsiani. From weak to strong LP gaps for all csps. In *Computational Complexity Conference*, volume 79 of *LIPIcs*, pages 11:1–11:27. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

28 Mrinalkanti Ghosh, Fernando Granha Jeronimo, Chris Jones, Aaron Potechin, and Goutham Rajendran. Sum-of-squares lower bounds for sherrington-kirkpatrick via planted affine planes, 2020. `arXiv:2009.01874`.

29 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.

30 D. Grigoriev. Complexity of Positivstellensatz proofs for the knapsack. *Comput. Complexity*, 10(2):139–154, 2001. doi:10.1007/s00037-001-8192-0.

31 Dima Grigoriev and Nicolai Vorobjov. Complexity of Null- and Positivstellensatz proofs. *Ann. Pure Appl. Logic*, 113(1-3):153–160, 2002. First St. Petersburg Conference on Days of Logic and Computability (1999). doi:10.1016/S0168-0072(01)00055-0.

32 Max Hopkins, Tali Kaufman, and Shachar Lovett. High dimensional expanders: Random walks, pseudorandomness, and unique games. *CoRR*, abs/2011.04658, 2020. arXiv:2011.04658.

33 S. B. Hopkins, P. K. Kothari, A. Potechin, P. Raghavendra, T. Schramm, and D. Steurer. The power of sum-of-squares for detecting hidden structures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 720–731, 2017. doi:10.1109/FOCS.2017.72.

34 Sam B. Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs, 2017.

35 Samuel B. Hopkins, Pravesh K. Kothari, and Aaron Potechin. Sos and planted clique: Tight analysis of MPW moments at all degrees and an optimal lower bound at degree four. *CoRR*, abs/1507.05230, 2015. arXiv:1507.05230.

36 Samuel B. Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In *COLT*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 956–1006. JMLR.org, 2015.

37 Samuel B. Hopkins and David Steurer. Efficient bayesian estimation from few samples: Community detection and related problems. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 379–390. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.42.

38 David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998.

39 Sushrut Karmalkar, Adam R. Klivans, and Pravesh K. Kothari. List-decodable linear regression. *CoRR*, abs/1905.05679, 2019. arXiv:1905.05679.

40 Pravesh Kothari, Ryan O'Donnell, and Tselil Schramm. SOS lower bounds with hard constraints: think global, act local. *CoRR*, abs/1809.01207, 2018. arXiv:1809.01207.

41 Pravesh K. Kothari and Ruta Mehta. Sum-of-squares meets nash: lower bounds for finding any equilibrium. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1241–1248. ACM, 2018. doi:10.1145/3188745.3188892.

42 Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *STOC*, pages 132–145. ACM, 2017.

43 Pravesh K. Kothari and Jacob Steinhardt. Better agnostic clustering via relaxed tensor norms, 2017.

44 Pravesh K. Kothari and David Steurer. Outlier-robust moment-estimation via sum-of-squares. *CoRR*, abs/1711.11581, 2017. arXiv:1711.11581.

45 Dmitriy Kunisky and Afonso S. Bandeira. A tight degree 4 sum-of-squares lower bound for the sherrington-kirkpatrick hamiltonian. *CoRR*, abs/1907.11686, 2019. arXiv:1907.11686.

46 Dmitriy Kunisky, Alexander S. Wein, and Afonso S. Bandeira. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. *CoRR*, abs/1907.11636, 2019. arXiv:1907.11636.

47 Jean Bernard Lasserre. Optimisation globale et théorie des moments. *C. R. Acad. Sci. Paris Sér. I Math.*, 331(11):929–934, 2000. doi:10.1016/S0764-4442(00)01750-X.

48 Allen Liu and Ankur Moitra. Settling the robust learnability of mixtures of gaussians. *CoRR*, abs/2011.03622, 2020. arXiv:2011.03622.

49 László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991.

50 Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *FOCS*, pages 438–446. IEEE Computer Society, 2016.

51 Sidhanth Mohanty, Prasad Raghavendra, and Jeff Xu. Lifting sum-of-squares lower bounds: Degree-2 to degree-4. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 840–853, New York, NY, USA, 2020. Association for Computing Machinery. `doi:10.1145/3357713.3384319`.

52 Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

53 Prasad Raghavendra and Tselil Schramm. Tight lower bounds for planted clique in the degree-4 SOS program. *CoRR*, abs/1507.05136, 2015. `arXiv:1507.05136`.

54 Prasad Raghavendra and David Steurer. Integrality gaps for strong SDP relaxations of Unique Games. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2009*, pages 575–585. IEEE Computer Soc., Los Alamitos, CA, 2009. `doi:10.1109/FOCS.2009.73`.

55 Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. *CoRR*, abs/1905.04660, 2019. `arXiv:1905.04660`.

56 Prasad Raghavendra and Morris Yau. List decodable subspace recovery, 2020. `arXiv:2002.03004`.

57 Grant Schoenebeck. Linear level lasserre lower bounds for certain k-csps. In *FOCS*, pages 593–602. IEEE Computer Society, 2008.

58 Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. Tight integrality gaps for Lovasz-Schrijver LP relaxations of vertex cover and max cut. In *STOC'07—Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 302–310. ACM, New York, 2007. `doi:10.1145/1250790.1250836`.

59 Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990. `doi:10.1137/0403036`.

60 Madhur Tulsiani. CSP gaps and reductions in the lasserre hierarchy. In *STOC*, pages 303–312. ACM, 2009.

61 Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy [extended abstract]. In *STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing*, pages 303–312. ACM, New York, 2009.

A Satisfying the booleanity, edge and positivity constraints

We prove the following three simple claims.

▷ Claim 13. $\tilde{\mathbb{E}}^{\otimes k}$ and $\tilde{\mathbb{E}}'$ satisfy the booleanity constraints $\{x_{i,c}^2 = x_{i,c} : i \in [n], c \in [k]\}$.

▷ Claim 14. $\tilde{\mathbb{E}}^{\otimes k}$ and $\tilde{\mathbb{E}}'$ satisfy the edge constraints $\{x_{i,c}x_{j,c} = 0 : (i,j) \in E(G), c \in [k]\}$.

▷ Claim 15. $\tilde{\mathbb{E}}^{\otimes k}$ and $\tilde{\mathbb{E}}'$ satisfy the positivity constraint.

Proof of Claim 13. Since $\tilde{\mathbb{E}}'$ is obtained by restricting $\tilde{\mathbb{E}}^{\otimes k}$ to a smaller domain, it suffices to show that $\tilde{\mathbb{E}}^{\otimes k}$ satisfies the constraints. We observe that $\tilde{\mathbb{E}}^{\otimes k}$ satisfies the above constraints if and only if for all monomials $\prod_{c=1}^k \prod_{i \in S_c} x_{i,c}^{\alpha_{i,c}}$ (where each $\alpha_{i,c} \geqslant 1$), it holds that $\tilde{\mathbb{E}}^{\otimes k}[\prod_{c=1}^k \prod_{i \in S_c} x_{i,c}^{\alpha_{i,c}}] = \tilde{\mathbb{E}}^{\otimes k}[\prod_{c=1}^k \prod_{i \in S_c} x_{i,c}]$. We have that $\tilde{\mathbb{E}}^{\otimes k}[\prod_{c=1}^k \prod_{i \in S_c} x_{i,c}^{\alpha_{i,c}}] = \prod_{c=1}^k \tilde{\mathbb{E}}[\prod_{i \in S_c} x_i^{\alpha_{i,c}}] = \prod_{c=1}^k \tilde{\mathbb{E}}[\prod_{i \in S_c} x_i] = \tilde{\mathbb{E}}^{\otimes k}[\prod_{c=1}^k \prod_{i \in S_c} x_{i,c}]$, as $\tilde{\mathbb{E}}$ satisfies the constraints $x_i^2 = x_i$, and so we are done. ◁

Proof of Claim 14. Since $\tilde{\mathbb{E}}'$ is obtained by restricting $\tilde{\mathbb{E}}^{\otimes k}$ to a smaller domain, it suffices to show that $\tilde{\mathbb{E}}^{\otimes k}$ satisfies the constraints. We observe that $\tilde{\mathbb{E}}^{\otimes k}$ satisfies the above constraints if and only if for all multilinear monomials $\prod_{c=1}^k x_{S_c,c}$ of coloring degree at most $d-2$, it holds that $\tilde{\mathbb{E}}^{\otimes k}[x_{i,c}x_{j,c}\prod_{c'=1}^{k} x_{S'_c,c'}] = 0$. This is because by Claim 13, we can reduce any polynomial modulo the booleanity constraints to make it multilinear. Using the tensor product structure, we have $\tilde{\mathbb{E}}^{\otimes k}[x_{i,c}x_{j,c}\prod_{c'=1}^{k} x_{S'_c,c'}] = \prod_{c'\neq c}\tilde{\mathbb{E}}[x_{S_{c'}}]\cdot\tilde{\mathbb{E}}[x_{S_c}x_ix_j] = \prod_{c'\neq c}\tilde{\mathbb{E}}[x_{S_{c'}}]\cdot 0 = 0$, since $\tilde{\mathbb{E}}$ satisfies the edge constraints. This completes the proof. ◁

Proof of Claim 15. Since $\tilde{\mathbb{E}}'$ is obtained by restricting $\tilde{\mathbb{E}}^{\otimes k}$ to a smaller domain, it suffices to prove the claim only for $\tilde{\mathbb{E}}^{\otimes k}$. Let $\mathcal{M}$ be the moment matrix of $\tilde{\mathbb{E}}$. That is, $\mathcal{M}$ is the matrix indexed by sets (S,T) with $|S|,|T| \leqslant d/2$ and $\mathcal{M}(S,T) := \tilde{\mathbb{E}}[x_Sx_T]$. We note that for any $f \in \mathcal{P}^n_{d/2}$, $\tilde{\mathbb{E}}[f^2] = f^\top Mf$, where we interpret f as a vector of coefficients in the second expression. The moment matrix of $\tilde{\mathbb{E}}^{\otimes k}$ is indexed by tuples of sets $((S_1,\dots,S_k),(T_1,\dots,T_k))$ where $|S_c|,|T_c| \leqslant d/2$ for all $c \in [k]$. We observe that the moment matrix of $\tilde{\mathbb{E}}^{\otimes k}$ is $\mathcal{M}^{\otimes k}$, as the $((S_1,\dots,S_k),(T_1,\dots,T_k))$-th entry is $\tilde{\mathbb{E}}^{\otimes k}[\prod_{c=1}^{k} x_{S_c,c}x_{T_c,c}] = \prod_{c=1}^{k}\tilde{\mathbb{E}}[x_{S_c}x_{T_c}] = \prod_{c=1}^{k}\mathcal{M}(S_c,T_c)$. We also note that for any f with $\mathrm{cdeg}(f) \leqslant d/2$, it holds that $\tilde{\mathbb{E}}^{\otimes k}[f^2] = f^\top\mathcal{M}^{\otimes k}f \geqslant 0$, as the tensor product of a positive semidefinite matrix is also positive semidefinite. This shows that $\tilde{\mathbb{E}}^{\otimes k}[f^2] \geqslant 0$ for all f with $\mathrm{cdeg}(f) \leqslant d/2$, which finishes the proof. ◁

B Tightness of degree in Theorem 1

In this section, we prove the following lemma, showing that the upper bound on d in Theorem 1 is tight up to constant factors.

▶ **Lemma 16.** *With high probability over $G \sim G(n,1/2)$, there is no degree $8(1+o(1))\log_2 n$ coloring pseudo-expectation for G using $k \leqslant \frac{n}{e\cdot 2(1+o(1))\log_2 n}$ colors.*

Let $t = 2(1+o(1))\log_2 n$. We show that with high probability over $G \sim G(n,1/2)$, there is no degree $4t$ coloring pseudo-expectation for G using $k \leqslant \frac{n}{et}$ colors. We first observe that with high probability, the maximum independent set in G has size at most t. Suppose that we draw $G \sim G(n,1/2)$ such that this holds, and suppose that such a pseudo-expectation $\tilde{\mathbb{E}}'$ exists. We observe that there is a natural action of permutations $\sigma\colon [k] \to [k]$ on $\tilde{\mathbb{E}}'$, given by $\tilde{\mathbb{E}}'^{(\sigma)}[\prod_{c=1}^{k}\prod_{i\in S_c} x_{i,c}] := \tilde{\mathbb{E}}'[\prod_{c=1}^{k}\prod_{i\in S_c} x_{i,\sigma(c)}]$. Let $\tilde{\mathbb{E}}'' := \mathbb{E}_\sigma\tilde{\mathbb{E}}'^{(\sigma)}$ be the pseudo-expectation obtained by averaging over all σ. We then have that $\tilde{\mathbb{E}}''$ satisfies the coloring constraints and is symmetric with respect to the color classes, e.g. that $\tilde{\mathbb{E}}''[x_{i,c}] = \tilde{\mathbb{E}}''[x_{i,c'}]$ for all $c,c' \in [k]$. This implies that $\tilde{\mathbb{E}}''[x_{i,1}] = \frac{1}{k}\sum_{c=1}^{k}\tilde{\mathbb{E}}''[x_{i,c}] \geqslant \frac{1}{k}\cdot 1$. Let $\tilde{\mathbb{E}}$ be the projection of $\tilde{\mathbb{E}}''$ onto the first color, so that $\tilde{\mathbb{E}}[\prod_{i\in S} x_i] := \tilde{\mathbb{E}}''[\prod_{i\in S} x_{i,1}]$. We then see that $\tilde{\mathbb{E}}$ is a degree $4t$ independent set pseudo-expectation with $\tilde{\mathbb{E}}[\sum_i x_i] \geqslant \omega$, where $\omega := \frac{n}{k} \geqslant et$.

To complete the proof, we show the following lemma.

▶ **Lemma 17.** *Suppose that the maximum independent set in G has size $\leqslant t$. Then there is no degree $4t$ independent set pseudo-expectation $\tilde{\mathbb{E}}$ for G with $\tilde{\mathbb{E}}[\sum_i x_i] = \omega \geqslant et$.*

Proof. Suppose that such a pseudo-expectation $\tilde{\mathbb{E}}$ exists. Let $f = \sum_i x_i$, and let $\ell \in \mathbb{N}$ be the smallest integer so that $2^\ell \geqslant 2t$. Note that $2^\ell \leqslant 4t$ must hold also. By Cauchy-Schwarz, we have

$$\begin{aligned} &\tilde{\mathbb{E}}[f^{2^\ell}] \geqslant (\tilde{\mathbb{E}}[f^{2^{\ell-1}}])^2 \ , \\ &\tilde{\mathbb{E}}[f^{2^{\ell-1}}] \geqslant \tilde{\mathbb{E}}[f^{2^{\ell-2}}]^2 \geqslant \ldots \geqslant \tilde{\mathbb{E}}[f]^{2^{\ell-1}} \ , \\ &\implies \tilde{\mathbb{E}}[f^{2^\ell}] \geqslant (\tilde{\mathbb{E}}[f^{2^{\ell-1}}])\cdot(\tilde{\mathbb{E}}[f])^{2^{\ell-1}} = \tilde{\mathbb{E}}[f^{2^{\ell-1}}]\cdot\omega^{2^{\ell-1}} \ . \end{aligned}$$

Note that each polynomial above has degree at most $2^\ell \leqslant 4t$, so the above pseudo-expectations are all well-defined. Now, we observe that

$$\tilde{\mathbb{E}}[f^{2^{\ell-1}}] = \tilde{\mathbb{E}}[\sum_{S\subseteq[n]:|S|\leqslant 2^{\ell-1}} c_Sx_S] = \sum_{S:|S|\leqslant t,\ S \text{ indep set in } G} c_S\tilde{\mathbb{E}}[x_S] \ ,$$

$$\tilde{\mathbb{E}}[f^{2^\ell}] = \sum_{S:|S|\leqslant t,\ S \text{ indep set in } G} c'_S \tilde{\mathbb{E}}[x_S]$$

where the coefficients c_S and c'_S are each nonnegative integers. Notice that $c'_S \leqslant |S|^{2^\ell}$, as every contribution to x_S is made by choosing an $i \in S$ from each of the $\sum_i x_i$ factors. We also observe that $c_S \geqslant |S|^{2^{\ell-1}-|S|} \cdot (|S|!)$, as we can choose each $i \in S$ exactly once from the first $|S|$ factors, and then select an arbitrary $i \in S$ from the remaining $2^{\ell-1} - |S|$ factors. Note that here we use the fact that $|S| \leqslant t \leqslant 2^{\ell-1}$ always holds. Fix S, and let $s = |S|$. We observe that

$$\frac{c'_S}{c_S} \leqslant \frac{s^{2^\ell}}{s^{2^{\ell-1}-s} \cdot s!} \leqslant s^{2^{\ell-1}} \cdot s^s \cdot \frac{1}{\sqrt{2\pi} \cdot s^{s+\frac{1}{2}} e^{-s}} < s^{2^{\ell-1}} \cdot e^s \leqslant (e \cdot s)^{2^{\ell-1}} \leqslant \omega^{2^{\ell-1}} \ ,$$

using Stirling's approximation and the fact that $\omega \geqslant et \geqslant es$. It therefore follows that $(c'_S - c_S\omega^{2^{\ell-1}})\tilde{\mathbb{E}}[x_S] < 0$. Hence,

$$\tilde{\mathbb{E}}[f^{2^\ell}] - \tilde{\mathbb{E}}[f^{2^{\ell-1}}] \cdot \omega^{2^{\ell-1}} = \sum_{S:|S|\leqslant t,\ S \text{ indep set in } G} (c'_S - c_S\omega^{2^{\ell-1}})\tilde{\mathbb{E}}[x_S] < 0 \ ,$$

which is a contradiction. ◀

Junta Distance Approximation with Sub-Exponential Queries

Vishnu Iyer
University of California at Berkeley, CA, USA

Avishay Tal
University of California at Berkeley, CA, USA

Michael Whitmeyer
University of California at Berkeley, CA, USA

Abstract

Leveraging tools of De, Mossel, and Neeman [FOCS, 2019], we show two different results pertaining to the *tolerant testing* of juntas. Given black-box access to a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$:

1. We give a $\mathsf{poly}(k, \frac{1}{\varepsilon})$ query algorithm that distinguishes between functions that are γ-close to k-juntas and $(\gamma + \varepsilon)$-far from k'-juntas, where $k' = O(\frac{k}{\varepsilon^2})$.
2. In the non-relaxed setting, we extend our ideas to give a $2^{\widetilde{O}(\sqrt{k/\varepsilon})}$ (adaptive) query algorithm that distinguishes between functions that are γ-close to k-juntas and $(\gamma+\varepsilon)$-far from k-juntas. To the best of our knowledge, this is the first subexponential-in-k query algorithm for approximating the distance of f to being a k-junta (previous results of Blais, Canonne, Eden, Levi, and Ron [SODA, 2018] and De, Mossel, and Neeman [FOCS, 2019] required exponentially many queries in k).

Our techniques are Fourier analytical and make use of the notion of "normalized influences" that was introduced by Talagrand [32].

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Algorithms, Complexity Theory, Fourier Analysis, Juntas, Normalized Influence, Property Testing, Tolerant Property Testing

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.24

Acknowledgements We thank Anindya De, Shafi Goldwasser, Amit Levi, and Orr Paradise for very helpful discussions.

1 Introduction

The study of property testing, initiated by Blum, Luby, and Rubinfeld in their seminal work on linearity testing [8], is concerned with making fast decisions about a global object having some global property, while only accessing (or "querying") parts of it. This notion was further explored by Goldreich, Goldwasser, and Ron [17], who drew connections to the areas of learning theory and approximation algorithms in the context of graph properties. We focus on properties of Boolean functions, i.e., $f : \{\pm 1\}^n \to \{\pm 1\}$. First, we state the definition of a property testing algorithm $\mathcal{A}$. Given $\varepsilon > 0$ and a class of functions $\mathcal{C}$, we say that $\mathcal{A}$ is a property tester for $\mathcal{C}$ if it satisfies the following two conditions:

1. if $f \in \mathcal{C}$, then $\mathcal{A}$ accepts f with probability at least 2/3;
2. if $\mathsf{dist}(f, g) \geq \varepsilon$ for all $g \in \mathcal{C}$, then $\mathcal{A}$ rejects with probability at least 2/3.

In the above definition, $\mathsf{dist}(f, g) = \mathbf{Pr}[f(x) \neq g(x)]$ is the fraction of inputs on which f and g disagree under the uniform distribution. The primary measure of efficiency for such property testing algorithms is the algorithms *query complexity*, or the number of times it must use its black box access to f. Such query algorithms can be *adaptive* in that the coordinates on which they query f depend on previous answers, or they can be *nonadaptive* in that the algorithm always queries f in a predetermined manner.

In this writeup, our algorithms will be adaptive, and we will focus on testing the particular class of functions known as k-juntas. Juntas comprise a simple and natural class of functions: those that depend only on a smaller subset of their input variables. More precisely, a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$ is said to be a k-junta if there exists k coordinates $i_1, \ldots, i_k \in [n]$ such that $f(x)$ only depends on $x_{i_1}, \ldots, x_{i_k}$. In essence, juntas capture the existence of many irrelevant variables, and arise naturally in the context of feature selection in machine learning and many computational biology problems. A canonical example is the problem of determining the relationship between genes and phenotypes; for example, one might wish to test whether a particular physical trait is a function of many genes or only a small number.

The fundamental problem of learning and/or testing juntas has been given much attention in recent years. We refer the reader to the works of Mossel, O'Donnell, and Servedio [24] and Valiant [33] for the most recent work on learning k-juntas. In this paper, we focus on the problem of testing juntas. Testing 1-juntas (aka dictators) and related functions had initial theoretical interest in the context of long-code testing in PCPs [18, 3], and was first formally explored in [29], which gave algorithms for testing dictators, monomials, and monotone DNFs. The more general problem of testing k-juntas was first studied by Fischer et. al. [16], where they exhibited a k-junta tester with query complexity $\widetilde{O}(k^2)$ queries to f. Crucially, their upper bound lacked any dependence on the ambient dimension n. More recently, it was shown in [5] that $O(k \log k + k/\varepsilon)$ adaptive queries suffice to test k-juntas, and this is tight for constant ε [30, 12]. There has also been recent interest in the distribution free setting for junta testing (wherein the distribution on inputs is not assumed to be uniform). Liu et al. [23] initially gave a $\widetilde{O}(k^2/\varepsilon)$-query algorithm with one-sided error, which was quickly followed up by the works of Bshouty [9] and Zhang [34] who gave $\widetilde{O}(k/\varepsilon)$-query algorithms with two-sided and one-sided error, respectively. The methods utilized by Bshouty extend those of Diakonikolas et al. [14] and result in algorithms not only for junta testing but also several subclasses of juntas. We note that while we solve a similar problem in a different setting, some of our techniques resemble those of [9]: notably, an idea introduced in [9] is to find a witness such that, if all coordinates outside a subset of the coordinates are fixed to this witness' values, then f becomes a dictator on a single coordinate within that subset. This can be thought of as obtaining oracle access to a relevant coordinate, an idea pervasive throughout the work of [13] and ours. The techniques in [14, 15, 9] can all be categorized in the "testing via implicit learning" paradigm, as surveyed in [31].

1.1 Tolerant Junta Testing

One of the first relaxations of the standard property testing model considered (sometimes referred to as the "parameterized" regime) were testers that distinguished between $f \in H$ and f being ε-far from $H' \supseteq H$. This notion was introduced by Kearns and Ron [20] in the context of testing decision trees and certain classes of neural networks. We note that if H' is a strict superset of H, then the job of the tester becomes easier, and smaller query or sample complexity is often achievable than in the regular testing model. Indeed, our Theorem 3 is an example of a (tolerant) parameterized tester. *Tolerant testing* is another generalization of the standard property testing model. The notion was first introduced by Parnas, Ron, and Rubinfeld [28]. Normal property testing entails distinguishing between functions that *exactly* satisfy a certain property, and functions that are ε-far from satisfying said property. This is somewhat restrictive, and the tolerant testing problem seeks to more generally distinguish

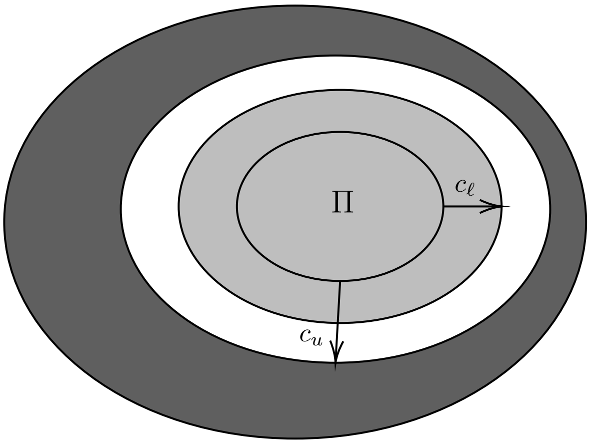


Figure 1 A visualization of the tolerant property testing paradigm. Assuming the outermost oval represents all functions $f : \{\pm 1\}^n \to \{\pm 1\}$ and the property at hand is represented by a class of functions Π, the goal is to distinguish between the light grey (at most c_ℓ close to a function in Π) and the dark grey (at least c_u far from all functions in Π) regions.

functions that are c_ℓ close to having the desired property, and those that are at least c_u far from having the property, for some $0 < c_\ell < c_u < 1$. We also note that the notion of tolerant testing is closely related to the notion of distance approximation – indeed, if one can estimate $\mathsf{dist}(f, \mathcal{C})$ up to additive error $(c_u - c_\ell)/2$ with probability at least $2/3$, then one has solved the tolerant testing problem for that class.[1] In general, tolerant testing (and therefore distance approximation), is much more challenging than traditional property testing. Figure 1 provides a visualization of the tolerant testing problem. Tolerant testing has received a lot of attention recently, see for example [7] for work on tolerant testing of decision trees and [1, 27] for work on tolerant testing of monotonicity. For the case of k-juntas, we have the following (relaxed) definition of a tolerant tester. In the following we denote by $\mathcal{J}_{n,k}$ the class of k-juntas, and for a class of functions $\mathcal{C}$, we denote $\mathsf{dist}(f, \mathcal{C}) := \min_{g\in\mathcal{C}} \mathsf{dist}(f, g)$.

▶ **Definition 1.** *For constants $0 < c_\ell < c_u < 1/2$ and a given $k', k \in \mathbb{N}$ with $k' \geq k$, a* (k, k', c_ℓ, c_u) tolerant junta tester *is an algorithm that, given oracle access to* $f : \{\pm 1\}^n \to \{\pm 1\}$*,*

1. *if* $\mathsf{dist}(f, \mathcal{J}_{n,k}) \leq c_\ell$ *accepts with probability* $2/3$*;*
2. *if* $\mathsf{dist}(f, \mathcal{J}_{n,k'}) \geq c_u$ *rejects with probability* $2/3$*.*

Our definition incorporates both tolerant and parameterized testers; when $c_\ell = 0$ the tester is non-tolerant and when $k' = k$ the tester is non-parameterized. We note that in the above definition we upper bound $c_u < 1/2$ since k-juntas are closed under complements, meaning if $g \in \mathcal{J}_{n,k}$, then $-g \in \mathcal{J}_{n,k}$. Parnas, Ron, and Rubinfeld in their seminal work [28] showed that while standard property testers, when querying uniformly, are weakly tolerant, entirely new algorithms are usually needed to tolerant test with better parameters. Tolerant junta testing was first considered by Diakonikolas et al. [14] which used the aforementioned observation from [28] to show that a standard tester from [16] actually gave a $(k, k, \mathsf{poly}(\frac{\gamma}{k}), \gamma)$ tolerant tester. Chakraborty et al. [10] subsequently showed that a similar analysis to that of Blais [5] gave a $(k, k, \gamma/C, \gamma)$ tolerant junta tester (for some constant C) using $\exp(k/\gamma)$ queries.

[1] The reverse direction is also true – given a tolerant tester it is possible to estimate the distance to that property. See for example section 3 in [1].

More recently, Blais et al. [6, Theorem 1.2] showed a tradeoff between query complexity and the amount of tolerance. In particular, they gave an algorithm which, given k, γ, and $\rho \in (0,1)$, is a $(k, k, \rho\gamma/16, \gamma)$ tolerant junta tester. The query complexity of the algorithm is $O\left(\frac{k \log k}{\gamma\rho(1-\rho)^k}\right)$. In particular, note that when ρ is a constant bounded away from zero, this yields an $\exp(k)$ query algorithm, but when $\rho = 1/k$ this yields a $\mathsf{poly}(k)$ query algorithm. We also note that there is an undesirable multiplicative "gap" between c_u and c_ℓ that precludes one from tolerantly testing for arbitrary close values of c_u and c_ℓ (i.e., in [6], $c_u \geq 16c_\ell$ for all choices of ρ). The recent work of [13] addressed this, giving an algorithm for any arbitrary $\gamma, \varepsilon > 0$ that required $2^k \mathsf{poly}(k, \frac{1}{\varepsilon})$ queries and was a $(k, k, \gamma, \gamma + \varepsilon)$ tolerant junta tester.

In the relaxed setting (when $k' \neq k$), [6, Theorem 1.1] also gave an algorithm which used $\mathsf{poly}(k, \frac{1}{\gamma})$ queries to f and was a $(k, 4k, \gamma/16, \gamma)$ tolerant junta tester. This once again posed the issue of not allowing for arbitrary c_u and c_ℓ values, which was resolved by [13, Corollary 1.6], which gave a $(k, O(k^2/\varepsilon^2), \gamma, \gamma+\varepsilon)$ tolerant junta tester with query complexity $\mathsf{poly}(k, \frac{1}{\varepsilon})$.

It is interesting to note that the techniques used to obtain the results from [6] and [13] are actually quite different, and yield results that are qualitatively similar but quantitatively incomparable. The results from [6] extend the techniques of [5], which partition the n input coordinates into $\mathsf{poly}(k)$ disjoint sets or "parts". It is immediate that any k-junta is a k-part junta, but in [5] it was shown that with high probability a function that is far from being a k-junta is also far from being a "k-part junta" (for a definition of this and more details we refer the reader to [5]). The results of [6] extend the idea of considering the relationship between k-juntas and k-part juntas in the context of tolerant testing.

The techniques in [13] suggest a new way of attacking the problem of tolerant k-junta testing. The core idea in [13] was to get access to "oracles" to coordinates of f which have large low-degree influence. These *coordinate oracles* are obtained with high probability via a combination of random restrictions and noise operators to the original function, and once obtained, can be used to search, in a brute force manner, for the nearest k-junta.

In terms of lower bounds for tolerant testing of juntas, two recent works addressed the non-adaptive case. Levi and Waingarten [22] demonstrated that there exists $0 < \varepsilon_1 < \varepsilon_2 < 1/2$ such that any $(k, k, \varepsilon_1, \varepsilon_2)$ tolerant junta tester requires $\widetilde{\Omega}(k^2)$ non-adaptive queries to f. In particular, this result demonstrated that the tolerant testing regime is quantitatively harder than the standard testing regime, in which a $\widetilde{O}(k^{3/2})$-query non-adaptive query algorithm is known [4] (and indeed optimal due to [11]). Subsequently, Pallavoor, Raskhodnikova, and Waingarten [27] demonstrated that for any $k \leq n/2$ there exists $0 < \varepsilon_1 < \varepsilon_2 < 1/2$ (with $\varepsilon_1 = O(1/k^{1-\eta})$ and $\varepsilon_2 = \Omega(1/\sqrt{k})$) such that every nonadaptive $(k, k, \varepsilon_1, \varepsilon_2)$-tolerant junta tester requires at least 2^{k^η} queries to f, for any $0 < \eta < 1/2$.[2]

1.2 Our Results

Our first result is a subexponential-in-k query tolerant junta tester in the standard (non-relaxed) setting. In fact, we obtain an ε-accurate estimate of the distance of f to the class of k-juntas.

▶ **Theorem 2.** *Given a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$, it is possible to estimate the distance of f from the class of k-juntas to within additive error ε with probability $2/3$ using $2^{\widetilde{O}(\sqrt{k/\varepsilon})}$ adaptive queries to f. In particular, when ε is constant, this yields a $2^{\widetilde{O}(\sqrt{k})}$-query algorithm. However, the algorithm still requires $\exp(k/\varepsilon)$ time.*

[2] We note that this lower bound does not necessarily rule out $\mathsf{poly}(k)\exp(1/\varepsilon)$ nonadaptive query $(k, k, \varepsilon_1, \varepsilon_2)$ (where $\varepsilon = \varepsilon_2 - \varepsilon_1$) tolerant junta testers due to the setting of ε_1 and ε_2 in their hard instance.

A simple corollary of the above theorem is that for any $0 < c_\ell < c_u < 1/2$, we have a (c_u, c_ℓ, k, k) tolerant junta tester with the same query complexity as in Theorem 2, where $\varepsilon = (c_u - c_\ell)/2$. This is an improvement of the results of [13, 6], whose tolerant junta testers when $k' = k$ required exponential query complexity in k in the worst case. We note that although we obtain this improvement, our algorithm still requires $\exp(k)$ time. In the appendix, we show a result solving a similar problem[3] with an improved dependence on ε, giving an algorithm requiring only $2^{\widetilde{O}(\sqrt{k}\log(1/\varepsilon))}$-queries and $\exp(k\log(1/\varepsilon))$ time (see Theorem 49).

In the relaxed/parameterized setting when $k' \neq k$, we give a polynomial-in-k query tolerant junta tester that is valid for any setting of c_u and c_ℓ, and reduces k' dependence on k to be linear instead of quadratic due to the result of [13, Corollary 1.6].

▶ **Theorem 3.** *For any $\gamma, \varepsilon > 0$ and $k \in \mathbb{N}$, there is an algorithm with query complexity* $\mathsf{poly}(k, 1/\varepsilon)$ *that is a $(k, O(k/\varepsilon^2), \gamma, \gamma + \varepsilon)$-tolerant junta tester.*

Theorem 3 is a simple corollary of the following theorem we prove.

▶ **Theorem 4.** *Let $\varepsilon > 0$, $k \in \mathbb{N}$, and $k' = O(k/\varepsilon^2)$. Then, there exists an algorithm that given parameters k, ε and oracle access to f makes at most $poly(k, 1/\varepsilon)$ queries to f and returns a number α such that with high probability (at least 0.99)*

1. $\alpha \leq \mathsf{dist}(f, \mathcal{J}_{n,k}) + \varepsilon$
2. $\alpha \geq \mathsf{dist}(f, \mathcal{J}_{n,k'}) - \varepsilon$

Indeed, to solve the problem in Theorem 3 we can apply the algorithm from Theorem 4 with $\varepsilon = (c_u - c_\ell)/3$ and accept if and only if $\alpha < \frac{1}{2}(c_u + c_\ell)$. If $\mathsf{dist}(f, \mathcal{J}_{n,k}) \leq c_\ell$ we have that with high probability $\alpha \leq c_\ell + \varepsilon < \frac{1}{2}(c_u + c_\ell)$ and we will accept. On the other hand, if $\mathsf{dist}(f, \mathcal{J}_{n,k'}) \geq c_u$ we have that with high probability $\alpha \geq c_u - \varepsilon > \frac{1}{2}(c_u + c_\ell)$ and we will reject.

Both of the algorithms used to prove Theorem 2 and Theorem 4 rely on the fact that we can get approximate oracle access to influential coordinates of f using techniques from [13]. From there, we analyze the Fourier coefficients of f after a series of random restrictions in order gain more information about the relevant coordinates of f at different Fourier levels. Along the way, we give an algorithm which provides us with oracle access to a junta in the following sense:[4]

▶ **Theorem 5** (Informal). *Let $f : \{\pm 1\}^n \to \{\pm 1\}$, $\mathcal{D} = \{g_1, \ldots, g_{k'}\}$ be a set of functions giving oracle access to a certain set of coordinates. Let g be a function from $\{\pm 1\}^{k'} \to [-1, 1]$ defined by $g(x) = \mathbf{E}[f(y) | g_1(y) = x_1, \ldots, g_{k'}(y) = x_{k'}]$. Then g can be computed by a randomized algorithm that runs in expected time* $\mathsf{poly}(k')$*.*

We note that one can view this as an oracle access to the junta, without even figuring out the coordinates on which the junta depends. More details on the ideas behind both algorithms can be found in Section 3.

[3] In particular, this problem is the problem of finding the subset of k inputs that "contain" the most Fourier mass – see Section 2 and Theorem 49 for more details.

[4] A similar technique appeared in [13, Section 5.1] to sample two inputs on which the coordinate oracles agree. We note that our algorithm allows to specify the values the coordinate oracles attain.

1.3 Structure of this Paper

Section 2 surveys some necessary preliminaries. Section 3 gives high level overviews of the techniques and ideas that go into the proofs of Theorem 4 and Theorem 2. Section 4 first describes how to get obtain “oracle access” to a junta (see Theorem 5) using only oracles for relevant coordinates of the junta, and then provides all the details of the algorithm and proof for Theorem 4. Finally, Section 5 provides all the details of the algorithm and proof for Theorem 2.

2 Preliminaries

Throughout the paper we adopt certain notation conventions. For a positive integer n, we denote by $[n]$ the set $\{1, \dots, n\}$. For a distribution $\mathcal{D}$, we denote that a random variable x is sampled according to $\mathcal{D}$ by $x \sim \mathcal{D}$. In the case that x is sampled uniformly at random from a set S, we will abuse notation slightly and write $x \sim S$. The binomial distribution with n trials and probability p per trial will be denoted $\mathrm{Bin}(n, p)$. We denote the set $\{-1, 1\}$ with the shorthand $\{\pm 1\}$. For functions f, g from $\{\pm 1\}^n$ to $\{\pm 1\}$ we define $\mathsf{dist}(f, g) = \mathbf{Pr}_{x \sim \{\pm 1\}^n}[f(x) \neq g(x)]$: that is, the fraction of inputs on which f and g differ. For a set $S \subseteq [n]$ we will denote by $\{\pm 1\}^S$ the set of possible assignments to the variables $\{x_i\}_{i \in S}$.

2.1 Probability

We recall the following Chernoff/Hoeffding bounds.

▶ **Fact 6.** *If $X_1, \dots, X_N$ are independent random variables bounded in $[0, 1]$ and $\bar{X} := \frac{1}{N}\sum_{i=1}^{N} X_i$, then we have*

$$\mathbf{Pr}[|\bar{X} - \mathbf{E}[\bar{X}]| \geq \eta] \leq 2\exp(-2N\eta^2),$$

Furthermore, denoting by $p = \mathbf{E}[\bar{X}]$, we have

$$\mathbf{Pr}[\bar{X} \leq p - \eta] \leq \exp(-2N\eta^2),$$

$$\mathbf{Pr}[\bar{X} \leq (1-\eta)p] \leq \left(\frac{e^{-\eta}}{(1-\eta)^{1-\eta}}\right)^{pN} \leq \exp\left(-\frac{\eta^2 pN}{2}\right).$$

2.2 Boolean Functions

In this section we recall some tools in the analysis of Boolean functions. For a more thorough introduction to the field, we refer the reader to [25]. For every subset $S \subseteq [n]$, we define the parity function on the bits in S, denoted by $\chi_S : \{\pm 1\}^n \to \{\pm 1\}$ as $\chi_S(x) = \prod_{i \in S} x_i$. It is a well-known fact that we can express uniquely any $f : \{\pm 1\}^n \to \mathbb{R}$ as a linear combination of $\{\chi_S\}_{S \subseteq [n]}$:

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x).$$

The coefficients $\{\widehat{f}(S)\}_{S \subseteq [n]}$ are referred to as the Fourier coefficients of f, and can be calculated by $\widehat{f}(S) = \mathbf{E}[f(x)\chi_S(x)]$. We say Fourier coefficients are on *level* s if they correspond to subsets of size s.

Given a function $f : \{\pm1\}^n \to \{\pm1\}$ and a coordinate $i \in [n]$, we define the *influence* of the i-th coordinate on f to be

$$\mathbf{Inf}_i[f] = \mathop{\mathbf{Pr}}_{x\sim\{\pm1\}^n}[f(x) \neq f(x^i)].$$

It is a well-known fact (see, e.g., [25, Theorem 2.20]) that $\mathbf{Inf}_i[f] = \sum_{S\ni i} \widehat{f}(S)^2$. The latter definition naturally extends to functions $f : \{\pm1\}^n \to \mathbb{R}$. We naturally extend this notion and define the *low-degree influence* (up to level k) of coordinate i on f as

$$\mathbf{Inf}_i^{\leq k}[f] = \sum_{S\ni i, |S|\leq k} \widehat{f}(S)^2.$$

For a set $T \subseteq [n]$ we define the *projection* of the function f to T, denoted $f^{\subseteq T}$, as the partial Fourier expansion restricted to sets contained in T, i.e., $f^{\subseteq T}(x) = \sum_{S:S\subseteq T} \widehat{f}(S)\chi_S(x)$. We observe that $f^{\subseteq T}$ depends only on coordinates in T and that it can be alternatively defined as $f^{\subseteq T}(x) = \mathbf{E}_{y\sim\{\pm1\}^n}[f(y)|y_T = x_T]$. As suggested by the last identity, we also denote $f^{\subseteq T}$ by $f_{\mathsf{avg},T}$.

In the regime of property testing, we will need a notion of "closeness" of functions.

▶ **Definition 7.** *For functions $f, g : \{\pm1\}^n \to \{\pm1\}$ and a set of functions G, all from $\{\pm1\}^n \to \{\pm1\}$ we say that*

1. *f is ν-*close *to g if $\mathsf{dist}(f, g) \leq \nu$;*
2. *f is ν-close to G if $\min_{g\in G} \mathsf{dist}(f, g) \leq \nu$;*
3. *f and g are c-correlated if $\mathbf{E}_{x\in\{\pm1\}^n}[f(x)g(x)] = c$;*
4. *f and G are c-correlated (denoted $\mathsf{corr}(f, G) = c$) if $\max_{g\in G} \mathbf{E}_{x\in\{\pm1\}^n}[f(x)g(x)] = c$.*

In the paper, we will occasionally abbreviate the correlation between f and g as $\mathbf{E}[fg]$ when the domain is implied. Observe that when f and g are Boolean-valued (in ±1) we have $\mathbf{E}[fg] = 1 - 2\mathsf{dist}(f, g)$.

▶ **Fact 8.** *For functions $f, g : \{\pm1\}^n \to \mathbb{R}$, we have Plancheral's identity:*

$$\mathop{\mathbf{E}}_{x\sim\{\pm1\}^n}[f(x)g(x)] = \sum_{S\subseteq[n]} \widehat{f}(S)\widehat{g}(S) .$$

When $f = g$, this fact is known as Parseval's identity.

▶ **Definition 9.** *For a function $f : \{\pm1\}^n \to \mathbb{R}$ we define:*

$$\mathbf{W}^{\leq k}[f] = \sum_{|S|\leq k} \widehat{f}(S)^2 .$$

The definitions of $\mathbf{W}^{\geq k}[f]$, $\mathbf{W}^{=k}[f]$, and similar follow from a natural extension. Now, we define some classes of Boolean functions with properties that will be useful to us.

▶ **Definition 10** (Junta). *Let $T \subseteq [n]$. A function $f : \{\pm1\}^n \to \mathbb{R}$ is called a* junta *on T if f depends only on coordinates in T. I.e., there exists a function $g : \{\pm1\}^T \to \mathbb{R}$ such that $f(x) = g(x_T)$. A function is called a k-*junta *if it is a junta on T for some $T \subseteq [n]$ of size k. Following the notation of [13], we denote the class of k-juntas on n inputs as $\mathcal{J}_{n,k}$. We also denote $\mathcal{J}_{U,k}$ as the set of k-juntas with inputs inside of U, and when $|U| = k$ then we often denote $\mathcal{J}_U := \mathcal{J}_{U,k}$ for brevity.*

▶ **Definition 11** (Dictator, Anti-Dictator). *The i-th dictator function is given by $\mathsf{Dict}_i(x) = x_i$, for $x \in \{\pm1\}^n$. The i-th antidictator function is simply the negation $-\mathsf{Dict}_i(x)$.*

▷ Claim 12 (Nearest k-junta on a Subset). For a function $f : \{\pm 1\}^n \to [-1, 1]$ and a subset $T \subseteq [n]$, the Boolean-valued junta-on-T most correlated with f is given by

$$\mathsf{sgn}(f_{\mathsf{avg},T}(x)) = \mathsf{sgn}\left(\mathop{\mathbf{E}}_{y\in\{\pm 1\}^n}[f(y)|y_T = x_T]\right).$$

Furthermore, the correlation between f and $\mathsf{sgn}(f_{\mathsf{avg},T}(x))$ is simply $\mathbf{E}_{x\sim\{\pm 1\}^n}[|f_{\mathsf{avg},T}(x)|]$.

We keep the proof for this well-known claim for completeness.

Proof. Let $g : \{\pm 1\}^n \to [-1, 1]$ be any junta-on-T. It suffices to show that $\mathbf{E}_x[f(x)g(x)] \leq \mathbf{E}[f(x)\mathsf{sgn}(f_{\mathsf{avg},T}(x))]$, as we do next. Indeed, for any $g(x)$ that is a junta-on-T we have $g(x) = g'(x_T)$ for some $g' : \{\pm 1\}^T \to [-1, 1]$. Thus, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\{\pm 1\}^n}[f(x)g(x)] &= \mathop{\mathbf{E}}_{x\sim\{\pm 1\}^n}[f(x)g'(x_T)] \\
&= \mathop{\mathbf{E}}_{x\sim\{\pm 1\}^n}\left[g'(x_T)\cdot \mathop{\mathbf{E}}_{y\sim\{\pm 1\}^n}[f(y)|x_T = y_T]\right] \\
&= \mathop{\mathbf{E}}_{x\sim\{\pm 1\}^n}[g'(x_T)f_{\mathsf{avg},T}(x)] \\
&\leq \mathop{\mathbf{E}}_{x\sim\{\pm 1\}^n}[|f_{\mathsf{avg},T}(x)|] \\
&= \mathop{\mathbf{E}}_{x\sim\{\pm 1\}^n}[\mathsf{sgn}(f_{\mathsf{avg},T}(x))\cdot f_{\mathsf{avg},T}(x)] \\
&= \mathop{\mathbf{E}}_{x\sim\{\pm 1\}^n}[f(x)\mathsf{sgn}(f_{\mathsf{avg},T}(x))]. \qquad \lhd
\end{aligned}$$

A useful tool in Boolean Function Analysis is the noise operator T_ρ. For a vector $x \in \{\pm 1\}^n$ we denote by $N_\rho(x)$ the distribution over vectors $y \in \{\pm 1\}^n$ such that for each coordinate $i \in [n]$ independently $y_i = x_i$ with probability $(1 + \rho)/2$ and $y_i = -x_i$ otherwise (alternatively, $\mathbf{E}[x_i y_i] = \rho$). For a function $f : \{\pm 1\}^n \to \mathbb{R}$ we denote by $T_\rho f : \{\pm 1\}^n \to \mathbb{R}$ the function defined by

$$T_\rho f(x) = \mathop{\mathbf{E}}_{y\sim N_\rho(x)}[f(y)]$$

There's also a nice Fourier expression for the function $T_\rho f$ given by $T_\rho f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\rho^{|S|}$. We will need a simple fact about the noise operator.

▶ **Fact 13** ([25, Exercise 2.33]). *For any function $f : \{\pm 1\}^n \to \mathbb{R}$ and any $\rho \in [-1, 1]$ we have that $\mathbf{E}[|T_\rho f|] \leq \mathbf{E}[|f|]$.*

2.3 Estimating Fourier Coefficients

The following claim is a standard tool in many learning algorithms. It establishes that estimating Fourier coefficients of a Boolean function f can be done with a few queries to f.

▷ Claim 14 ([25, Proposition 3.39]). Suppose $f : \{\pm 1\}^n \to \{\pm 1\}$ and $S \subseteq [n]$ then there exists an algorithm that estimates $\widehat{f}(S)$ up to additive error ε with probability at least $1 - \delta$ that makes $O((1/\varepsilon^2) \cdot \log(1/\delta))$ samples.

The next claim generalizes the claim to a bounded function $f : \{\pm 1\}^n \to [-1, 1]$. For that generalization, we need the definition of a randomized algorithm computing a bounded function f.

▶ **Definition 15** (Randomized Algorithm for a Bounded Function)**.** *Let $f : \{\pm 1\}^n \to [-1, 1]$ be a bounded function. We say that algorithm A is a randomized algorithm for f if on any fixed input x algorithm A outputs a random bit $\mathbf{y} \in \{\pm 1\}$ with $\mathbf{E}[\mathbf{y}] = f(x)$.*

▷ Claim 16. Let $f : \{\pm 1\}^n \to [-1, 1]$, and let A be a randomized algorithm for f. Then, there exists an algorithm making $O((1/\varepsilon^2) \cdot \log(1/\delta))$ calls to A that estimates $\widehat{f}(S)$ up to additive error ε with probability at least $1 - \delta$.

Proof Sketch. We estimate $\widehat{f}(S)$ by sampling $m = O((1/\varepsilon^2) \cdot \log(1/\delta))$ uniformly random inputs $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, applying A to each of them to get random bits $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$, and taking the empirical mean of $\frac{1}{m}\sum_{i=1}^m \mathbf{y}_i \cdot \chi_S(\mathbf{x}^{(i)})$. Note that for each $i \in [m]$ we have that $\mathbf{y}_i \cdot \chi_S(\mathbf{x}^{(i)})$ is a $\{\pm 1\}$ random variable with expectation

$$\mathop{\mathbf{E}}_{\mathbf{x}^{(i)}, \mathbf{y}_i}[\mathbf{y}_i \cdot \chi_S(\mathbf{x}^{(i)})] = \mathop{\mathbf{E}}_{\mathbf{x}^{(i)}}\left[\mathop{\mathbf{E}}_{\mathbf{y}_i}[\mathbf{y}_i | \mathbf{x}^{(i)}] \cdot \chi_S(\mathbf{x}^{(i)})\right] = \mathop{\mathbf{E}}_{\mathbf{x}^{(i)}}[f(\mathbf{x}^{(i)}) \cdot \chi_S(\mathbf{x}^{(i)})] = \widehat{f}(S).$$

The claim follows from Fact 6. ◁

2.4 Random Restrictions

▶ **Definition 17** (Restriction)**.** *Consider the class of functions on $\{\pm 1\}^n$. A restriction is a pair (J, z) where $J \subseteq [n]$, and $z \in \{\pm 1\}^{\overline{J}}$. Given a function $f : \{\pm 1\}^n \to \mathbb{R}$, and a restriction (J, z), the* restricted function *$f_{\overline{T}\to z} : \{\pm 1\}^T \to \mathbb{R}$ is defined by $f_{\overline{T}\to z}(x) = f(y)$ where $y_T = x$ and $y_{\overline{T}} = z$.*

▶ **Definition 18** (δ-Random Restriction)**.** *For $\delta \in [0, 1]$ we say that J is a δ-*random subset *of S if it is formed by including each element independently with probability δ, which we denote as $J \subseteq_\delta S$. A δ-random restriction, denoted $(J, z) \sim \mathcal{R}_\delta$, is sampled by taking J to be a δ-random subset J on $[n]$, and taking z to be a uniformly random string in $\{\pm 1\}^{\overline{J}}$.*

Occasionally, we will abuse notation and think of $f_{\overline{T}\to z}$ as a function from $\{\pm 1\}^n$ to $\{\pm 1\}$ that ignores bits outside T. For example, $f_{\overline{T}\to z} : \{\pm 1\}^n \to \{\pm 1\}$ is given by $f_{\overline{T}\to z}(x) = f(x_T, z_{\overline{T}})$. Finally, we will use the following fact on random restrictions:

▶ **Fact 19** ([25, Corollary 3.22])**.** *For a function $f : \{\pm 1\}^n \to \mathbb{R}$ and sets $S \subseteq J \subseteq [n]$ we have*

$$\mathop{\mathbf{E}}_{z \in \{\pm 1\}^{\overline{J}}}[\widehat{f_{\overline{J}\to z}}(S)^2] = \sum_{R \subseteq [n], R \cap J = S} \widehat{f}(R)^2.$$

3 Overview of Techniques

Both of our algorithms rely on only having to consider a subset of influential coordinates, rather than all n input variables. This is obtained using results from [13], and is discussed further in Section 4. For now, we simply assume that we are only dealing with $\mathsf{poly}(k, 1/\varepsilon)$ coordinates $\mathcal{S}$. For simplicity of presentation, we ignore dependence on ε, and focus only the dependence on k. Thus, in this section, assume that ε is a small universal constant, e.g., $\varepsilon = 0.01$.

3.1 Techniques for Establishing Theorem 4

Our first result shows how to further reduce the number of coordinates we need to consider down to $O(k/\varepsilon^2)$, while only losing at most ε amount of correlation with the maximally correlated k-junta. In establishing Theorem 4, we first develop intuition behind a notion of normalized influence that we introduce next:

▶ **Definition 20** (Normalized Influence). *Let $f : \{\pm 1\}^n \to \mathbb{R}$. We define the normalized influence of coordinate i on f as*

$$\mathbf{NInf}_i[f] = \sum_{S \ni i} \frac{\widehat{f}(S)^2}{|S|}.$$

We also naturally define the normalized influence below level k:

$$\mathbf{NInf}_i^{\leq k}[f] := \sum_{\substack{|S| \leq k \\ S \ni i}} \frac{\widehat{f}(S)^2}{|S|}.$$

We note that while the term "normalized influence" is new, the quantity itself is not. It first appeared in a work of Talagrand [32] (expressed as $M(\Delta_i f)^2$) which generalized the famous KKL theorem [19, 21], and subsequently appeared in followup works extending Talagrand's theorem to Schreier graphs [26]. As far as we know, this is the first use of this quantity in a learning or testing setting.

The next claim states that the sum of normalized influences of f equals its variance.

▷ Claim 21. For any function $f : \{\pm 1\}^n \to \mathbb{R}$, we have that $\sum_i \mathbf{NInf}_i[f] = \mathbf{Var}[f]$.

Proof. We have that

$$\sum_{i \in [n]} \mathbf{NInf}_i[f] = \sum_{i \in [n]} \sum_{S \ni i} \frac{\widehat{f}(S)^2}{|S|} = \sum_{\substack{S \subseteq [n] \\ S \neq \emptyset}} \sum_{i \in S} \frac{\widehat{f}(S)^2}{|S|} = \sum_{\substack{S \subseteq [n] \\ S \neq \emptyset}} |S| \frac{\widehat{f}(S)^2}{|S|} = \sum_{\substack{S \subseteq [n] \\ S \neq \emptyset}} \widehat{f}(S)^2 = \mathbf{Var}[f],$$

where the last equality follows from Parseval's identity. ◁

▶ Remark 22. We note that for a balanced Boolean function f (that is, one where $\mathbf{E}_x[f(x)] = 0$) the normalized influences form a probability distribution on the coordinates i.

The idea behind establishing Theorem 4 begins with the observation the these normalized influences can be thought of as defining a sub-probability distribution over the input coordinates of f, since these are non-negative numbers whose sum is at most 1. The weight assigned to coordinate i, similar to the regular influence, captures how important i is to f, but assigns a higher relative weight to the coordinates with Fourier mass coming from the lower levels of the Fourier decomposition.

The second important observation for us is that for any set T of size at most k we can write

$$\sum_{i \in T} \mathbf{NInf}_i^{\leq k}[f] = \sum_{i \in T} \sum_{\substack{|S| \leq k \\ \emptyset \neq S \ni i}} \frac{\widehat{f}(S)^2}{|S|} \geq \sum_{i \in T} \sum_{\substack{S \subseteq T \\ S \ni i}} \frac{\widehat{f}(S)^2}{|S|} = \sum_{\emptyset \neq S \subseteq T} \widehat{f}(S)^2. \tag{1}$$

Intuitively, this shows that if some set of coordinates captures large amount of Fourier mass, then this same subset of coordinates also is very likely to be sampled by our sub-probability distribution defined by the normalized influences. Our idea follows this line of thought – we get decent estimates for all of the normalized influences, and sample coordinates from this estimated distribution. Let T be the "target set" of size k, i.e., the one for which the closest k-junta to f is a junta on T. Without loss of generality we can assume that T captures constant fraction of the Fourier mass, meaning $\sum_{\emptyset \neq S \subseteq T} \widehat{f}(S)^2 \geq \Omega(1)$. Otherwise, the best correlation of f with a k-junta is $o(1) < \varepsilon$ and the task of ε-accurately estimating the distance to the set of k-juntas becomes trivial. Assuming T captures constant fraction of the Fourier mass, Equation (1) tells us that we will sample $i \in T$ with constant probability mass. Thus, sampling from this distribution $O(k)$ times means we will have seen most of T up to a small loss in correlation.

To actually estimate these normalized influences, we apply a series of $\log 10k$ random restrictions to our function f (first take 1-random restrictions, then 1/2-random restrictions, then 1/4-random restrictions, and so on), and then show that summing $\widehat{f_{\bar{J}\to z}}(\{i\})^2$ for each of these restrictions is sandwiched between $\mathbf{NInf}_i^{\leq k}[f]$ and $\mathbf{NInf}_i[f]$:

$$\frac{1}{2}\mathbf{NInf}_i^{\leq k}[f] \leq \sum_{i=0}^{\log 10k} \mathop{\mathbf{E}}_{(J,z)\sim\mathcal{R}_{2^{-i}}}\left[\widehat{f_{\bar{J}\to z}}(\{i\})^2\right] \leq 2\mathbf{NInf}_i[f].$$

This would allow us to effectively sample from a proxy distribution that still samples $i \in T$ with constant probability.

We repeat the process iteratively, sampling coordinates one at a time, until we either sampled all of T or sampled a subset $T' \subseteq T$ for which we have that the best junta on T' is almost as correlated with f as the best junta on T. Since the process samples a coordinate in T with constant probability in each round, after $O(k)$ iterations we are likely to succeed, giving us a set U of $O(k)$ coordinates that contains either T or T' (as above). Finally, we show we can estimate, up to a small additive error, the best correlation of a junta-on-U with f, given only approximate oracle access to the coordinates in $\mathcal{S}$. By the above discussion the estimate we get is lower bounded by the best correlation with a k-junta up to a small additive error. It is also upper bounded (trivially) with the best correlation of f with a $O(k)$-junta, since $|U| = O(k)$.

3.2 Techniques for Theorem 2

A limitation of the algorithm we described in the previous subsection is that it only samples one coordinate at a time. In particular, suppose we want to find T exactly, instead of a superset U of T. Then, the naive algorithm would need to consider all subsets of U of size k, estimating the best correlation with a junta on each of them. This gives a $\exp(O(k))$-query algorithm. It would be nicer if we can devise a sampling algorithm that outputs, with constant probability, many coordinates of T at a time. Such a sampling algorithm would reduce the number of possibilities for T in the second stage. In particular, consider the case that the nearest k-junta to f had significant amount of Fourier mass on higher levels, say at level $\approx k$ or maybe $\approx \sqrt{k}$. In this case it would be nice to be able to sample from the Fourier distribution of f, that would give us a large subset of T with constant probability. We note that sampling from the Fourier distribution of a Boolean function is easy for a quantum algorithm but hard for a randomized algorithm. Nevertheless, the (classical) algorithm we describe in this section takes inspiration from this, and samples subsets of size $\sqrt{k}$ according to the Fourier mass of f above level $\sqrt{k}$ of each subset, in time and query complexity $\exp(\widetilde{O}(\sqrt{k}))$.

We will start with the preliminary that we have reduced to the case of only having to consider the coordinates in $\mathcal{S} \subseteq [n]$ with $|\mathcal{S}| \leq O(k/\varepsilon^2)$, using our aforementioned algorithm from the previous section, incurring only a small additive loss in correlation with the closest k-junta. We start with the following definition that generalizes normalized influences of coordinates to normalized influences of sets of coordinates.

▶ **Definition 23.** *For a given subset $U \subseteq [n]$, we define its normalized influence as follows:*

$$\mathbf{NInf}_U[f] := \sum_{S\colon U \subseteq S} \frac{\widehat{f}(S)^2}{\binom{|S|}{|U|}}.$$

We also have the natural extension of $\mathbf{NInf}_U^{\leq k}[f] = \sum_{S\colon |S| \leq k, U \subseteq S} \frac{\widehat{f}(S)^2}{\binom{|S|}{|U|}}$, analogous to Definition 20.

This is a direct generalization of the quantity in Definition 20. In particular, we consider taking $|U| = \sqrt{k}$. Note there are $2^{\widetilde{O}(\sqrt{k})}$ such U within the coordinates in $\mathcal{S}$, and we can think of these normalized influences as once again defining a sub-probability distribution over subsets of size $\sqrt{k}$. It likely does not sum to 1, but rather sums to $\mathbf{W}_{\mathcal{S}}^{\geq \sqrt{k}}[f] \leq 1$. We show that these normalized influences at exactly level $\sqrt{k}$ can once again be approximated to within a constant factor via a sequence of random restrictions to f:

$$\frac{1}{2}\mathbf{NInf}_U^{\leq k}[f] \leq \sum_{i=0}^{2\sqrt{k}\log 10k} \mathop{\mathbf{E}}_{(J,z)\sim \mathcal{R}_{p^i}} \left[\widehat{f_{\bar{J}\to z}}(U)^2\right] \leq 3\mathbf{NInf}_U[f],$$

where $p = \left(1 - \frac{1}{2\sqrt{k}}\right)$. For more details on this statement, see Theorem 40.

We are now ready to outline the overall algorithm in Section 5. Suppose $T \subseteq \mathcal{S}$ is the subset on which the nearest k-junta (within $\mathcal{S}$) is defined. Our algorithm can then be broken down into two phases:

Phase 1. We get a proxy for $\mathbf{NInf}_U$ for all $|U| = \sqrt{k}$. This is achieved by performing a series of random restrictions to f.

We consider these proxies as a distribution, and sample a constant (this constant is actually dependent on ε, see Section 5 for details) number of subsets of size $\sqrt{k}$. With high probability, one of these is in our set of interest T, provided T has a non-negligible amount of Fourier mass above level $\sqrt{k}$.

We don't know which of the subsets we sample are actually in T, so we start a branching process. For each subset we sampled, we restrict f's values in that subset, and recursively sample from sets of size $\sqrt{k}$ using the steps described above. Our branching process will have depth at most $\sqrt{k}$ since at each level we sample $\sqrt{k}$ new coordinates, and T can have at most k relevant coordinates. This phase of our algorithm produces $2^{\widetilde{O}(\sqrt{k})}$ possible subsets of our target set T.

Phase 2. With high probability, one of the branches in the above process will have captured most of the coefficients of T that are relevant above level $\sqrt{k}$ on the Fourier spectrum. Each branch of this process represents a different possibility for what T may be, so for each branch we randomly restrict f so that the coordinates sampled in that branch are fixed, which effectively moves most of the mass of T to levels below $\sqrt{k}$. We then estimate all the Fourier coefficients of this restricted f below level $\sqrt{k}$, allowing us to get an estimate for the closest k-junta on any subset using these estimated coefficients. Each estimation of a Fourier coefficient requires $2^{\widetilde{O}(\sqrt{k})}$-queries to estimate to the desired accuracy, and there are $2^{\widetilde{O}(\sqrt{k})}$ Fourier coefficients to estimate, so overall we make at most $2^{\widetilde{O}(\sqrt{k})}$ queries. From there, for each possible subset of $B \subseteq T$ outputted by phase one, we brute force over all possible subsets of size k containing B, estimating the correlation f has with the closest k-junta on that subset using our estimated Fourier coefficients. This last step takes exponential time in k. We emphasize that while our runtime is exponential in k, our query complexity is only exponential in $\widetilde{O}(\sqrt{k})$.

In the entire above explanation, we have eliminated the dependence on ε for simplicity. We also only consider T for conceptual and analytic simplicity – in reality, we have no idea what T is, and indeed it is exactly what we are looking for. Therefore, more work must be done in order to show that we do not accidentally pick the wrong set, for which our estimates may be inaccurate. To get around this subtle issue, we further apply a noise operator in order to ensure that the significant parts of f lie below level roughly $\sqrt{k}$. We discuss this further in Section 5.2.

4 Finding a Small(er) Set of Influential Coordinate Oracles

In this section, we detail the process of constructing oracles to coordinates with large low-degree influence. We expand upon the techniques in [13], reducing the number of coordinates one needs to consider to produce a highly correlated k-junta (assuming one exists).

4.1 Approximate Oracles to Influential Coordinates

In this subsection we outline and generalize the methods used by [13] to achieve oracle access to coordinates with large low-degree infuence in f. We start with the following definitions from their paper, repeated here for clarity:

▶ **Definition 24** ([13, Def. 3.1]). *Let $\mathcal{D}$ be a set of functions mapping $\{\pm 1\}^n$ to $\{\pm 1\}$. We say that $\mathcal{D}$ is an oracle for the coordinates in $\mathcal{S}$ if*

- *for every $g \in \mathcal{D}$, there is some $i \in \mathcal{S}$ such that $g = \pm\mathsf{Dict}_i$; and*
- *for every $i \in \mathcal{S}$, there is some $g \in \mathcal{D}$ such that $g = \pm\mathsf{Dict}_i$.*

In other words, $\mathcal{D}$ is an oracle for $\mathcal{S}$ if $\mathcal{D} = \{\mathsf{Dict}_i : i \in \mathcal{S}\}$ "up to sign".

However, it is not tractable to achieve perfect access to such oracles, so we have to settle for the following weaker notion of approximate oracles:

▶ **Definition 25** ([13, Def. 3.2]). *Let $\mathcal{D}$ be a set of functions mapping $\{\pm 1\}^n$ to $\{\pm 1\}$. We say that $\mathcal{D}$ is an ν-oracle for the coordinates in $\mathcal{S}$ if*

- *for every $g \in \mathcal{D}$, there is some $i \in \mathcal{S}$ such that g is ν-close to $\pm\mathsf{Dict}_i$; and*
- *for every $i \in \mathcal{S}$, there is exactly one $g \in \mathcal{D}$ such that g is ν-close to $\pm\mathsf{Dict}_i$; and*
- *For every $g \in \mathcal{D}$, and $\delta > 0$, there is a randomized algorithm that compute $g(x)$ correctly on any $x \in \{\pm 1\}^n$ with probability at least $1 - \delta$, using $\mathsf{poly}(k, \log\frac{1}{\delta})$ queries to f.*

Lemma 3.6 in [13] establishes that we can achieve access to a set $\mathcal{D}$ of approximate oracles to $\mathcal{S} \supseteq \{i : \mathbf{Inf}_i^{\leq k}[f] \geq \varepsilon^2/k\}$ of bounded size. More specifically, we have the following corollary:

▶ **Corollary 26** ([13, Lemma 3.6]). *With $\mathsf{poly}(k, \frac{1}{\varepsilon}, \log\frac{1}{\delta}) \cdot \frac{1}{\nu}$ queries to f, we can gain access to an approximate oracle set $\mathcal{D}$ in the sense that for every coordinate i such that $\mathbf{Inf}_i^{\leq k}[f] \geq \frac{\varepsilon^2}{k}$, there exists a $g \in \mathcal{D}$ such that g is ν-close to $\pm\mathsf{Dict}_i$ with probability at least $1-\delta$. Furthermore, $|\mathcal{D}| \leq \mathsf{poly}(k, \frac{1}{\varepsilon}, \log(1/\delta))$.*

For our purposes, we take $\nu = 0.1$ and $\delta = 2^{-\mathsf{poly}(k, \frac{1}{\varepsilon})}$ in all our algorithms. Since we will make much fewer than $2^{\mathsf{poly}(k/\varepsilon)}$-many queries to the coordinate oracles, we can assume that all of our oracles are indeed $\nu = 0.1$ close to dictators/anti-dictators, since by a union bound this is true with high probability.

It is important to note that we do not have a description of which coordinates are influential: from an information theoretic standpoint this would require query complexity dependent on n. What we do have is oracle access to these coordinates in the sense that for all i such that $\mathbf{Inf}_i^{\leq k}[f] \geq \varepsilon^2/k$, there exists $g_i \in \mathcal{D}$ such that $g_i(x) \approx \pm\mathsf{Dict}_i(x)$, that is, $\mathcal{D}$ contains dictators or anti-dictators to every influential coordinate. Using simple techniques of local correction we can simplify this: we need only consider dictators to each coordinate in the oracle. Also, we can convert closeness on average x to high probability correctness for all x (i.e., a worst-case guarantee).

▶ **Lemma 27.** *Suppose f is ν-close to $\pm\mathsf{Dict}_i$. For any $x \in \{\pm1\}^n$, $\mathsf{LocalCorrect}(f, x)$ samples a random $y \sim \{\pm1\}^n$ and outputs $f(y)f(x \cdot y)$, where $x \cdot y$ is pointwise multiplication. Then,*

$$\forall x : \Pr_{y\sim\{\pm1\}^n}[\mathsf{LocalCorrect}(f, x) \neq \mathsf{Dict}_i(x)] \leq 2\nu.$$

Proof. Suppose that f is ν close to Dict_i. Then we have $\mathbf{Pr}_{y\sim\{\pm1\}^n}[f(y) \neq \mathsf{Dict}_i(y)] \leq \nu$, and since $x \cdot y$ has the same distribution as y, $\mathbf{Pr}_{y\sim\{\pm1\}^n}[f(x \cdot y) \neq \mathsf{Dict}_i(x \cdot y)] \leq \nu$. Let A be the event that $f(y) \neq \mathsf{Dict}_i(y)$ and let B be the event that $f(x \cdot y) \neq \mathsf{Dict}_i(x \cdot y)$. Clearly if $\mathsf{LocalCorrect}(f, x) \neq \mathsf{Dict}_i(x)$ then at least one of A and B must have occurred (since $\mathsf{Dict}_i(x) = \mathsf{Dict}_i(x \cdot y) \cdot \mathsf{Dict}_i(y)$). Thus, by the union bound, we have

$$\Pr_{y\sim\{\pm1\}^n}[\mathsf{LocalCorrect}(f, x) \neq \mathsf{Dict}_i(x)] \leq \mathbf{Pr}[A \cup B] \leq \mathbf{Pr}[A] + \mathbf{Pr}[B] \leq 2\nu$$

A similar argument shows that if f is ν close to $-\mathsf{Dict}_i$, then $\mathsf{LocalCorrect}(f, x)$ is not equal to $(-\mathsf{Dict}_i(y))(-\mathsf{Dict}_i(x \cdot y)) = \mathsf{Dict}_i(y)\mathsf{Dict}_i(x \cdot y) = \mathsf{Dict}_i(x)$ with probability at most 2ν. ◀

Given a noisy black box computing h which is ν-close to $g = \pm\mathsf{Dict}_i$, local correction will compute Dict_i with high probability, on every input x. Critically, we can treat potentially faulty $\pm\mathsf{Dict}_i$ oracles as correct Dict_i oracles provided suitably many repetitions.

▶ **Corollary 28.** *If f is ν-close to $\pm\mathsf{Dict}_i$ for $\nu = 0.1$, then repeating $\mathsf{LocalCorrect}(f, x)$ independently $\mathsf{poly}(k, 1/\varepsilon)$ times and taking the majority outcome results in an incorrect value for $\mathsf{Dict}_i(x)$ with probability at most $2^{-\mathsf{poly}(k,1/\varepsilon)}$.*

Proof. Clear from applying the first bound in Fact 6 with $N = O(\mathsf{poly}(k/\varepsilon))$ and $\eta = (1 - 2\nu - 0.5) = 0.3$ in this case. ◀

We also show that restricting our attention to $\mathcal{S}$ we have not lost more than ε in the best correlation of f with a k-junta. This is proved in the following claim.

▷ Claim 29. Let $f : \{\pm1\}^n \to \{\pm1\}$ and let $g : \{\pm1\}^n \to \{\pm1\}$ be a k-junta on U. Let $\tau > 0$. Take

$$S = \left\{ i \in U \;\middle|\; \mathbf{Inf}_i^{\leq k}[f] \geq \frac{\tau^2}{k} \right\}$$

Then, there is a junta on S with correlation at least $\mathbf{E}[fg] - \tau$ with f.

Proof. To prove this claim, we define a function on the set S such that the loss in correlation is at most τ. Consider:

$$g'(x) = g_{\mathsf{avg},S}(x) = \mathop{\mathbf{E}}_{y}[g(y) | y_S = x_S].$$

First, we note g' is a function over only the variables in S. Second, it is bounded in $[-1,1]$, so it is not quite Boolean, but it can be randomized rounded to a Boolean function, with the expected correlation with f equaling $\mathbf{E}[fg']$. Thus, it suffices to show that $\mathbf{E}[fg'] \geq \mathbf{E}[fg] - \tau$ to deduce that there exists a randomize rounding of g' to a Boolean function g'' with $\mathbf{E}[fg''] \geq \mathbf{E}[fg] - \tau$. We also recall that

$$\widehat{g'}(T) = \begin{cases} \widehat{g}(T) & \text{if } T \subseteq S \\ 0 & \text{otherwise} \end{cases}$$

We thus have:

$$|\mathbf{E}[fg] - \mathbf{E}[fg']| = \left| \sum_{\substack{T \not\subseteq S \\ T \subseteq U}} \widehat{f}(T)\widehat{g}(T) \right| \leq \sqrt{\sum_{\substack{T \not\subseteq S \\ T \subseteq U}} \widehat{f}(T)^2} \leq \sqrt{\sum_{i \in U \setminus S} \sum_{\substack{T \ni i \\ T \subseteq U}} \widehat{f}(T)^2}$$

$$\leq \sqrt{\sum_{i \in U \setminus S} \mathbf{Inf}_i^{\leq k}(f)} \leq \sqrt{k \cdot \frac{\tau^2}{k}} = \tau \qquad ◀$$

Finally, the below corollary summarizes what we have achieved in this section.

▶ **Corollary 30.** *With* $\mathsf{poly}(k, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$ *queries to* f*, we can gain access to an approximate oracle set* $\mathcal{D}$ *for a set of coordinates* $\{i : \mathbf{Inf}_i^{\leq k} \geq \frac{\varepsilon^2}{k}\} \subseteq \mathcal{S} \subseteq [n]$*. Moreover, these coordinates and oracles satisfy the following properties.*

- *For every coordinate* $i \in \mathcal{S}$*, there exists a* $g \in \mathcal{D}$ *such that* g *is* 0.1*-close to* Dict_i *with probability at least* $1 - \delta$*.*
- $\mathsf{dist}(f, \mathcal{J}_{n,k}) - \mathsf{dist}(f, \mathcal{J}_{\mathcal{S},k}) \leq \varepsilon$.
- $|\mathcal{S}| \leq \mathsf{poly}(k, 1/\varepsilon, \log(1/\delta))$.
- *For any algorithm* A *that uses at most* q *queries to* $\mathcal{D}$*, we can use* LocalCorrect *from Lemma 27 with error* δ/q *to assume that we actually have perfect access to each coordinate oracle, up to an additive loss of* δ *in confidence and a multiplicative overhead of* $\mathsf{poly}(\log(q/\delta))$ *in query complexity.*

Proof. The first and the third bullet point follow from Corollary 26. The second bullet point follows from Claim 29. To achieve the last point, we can use Corollary 28 every time we make a "query" to an oracle in our algorithm. Thus every "query" to an oracle $g \approx \pm\mathsf{Dict}_i$ at x involves $\mathsf{poly}(\log(q/\delta))$ many repetitions of $\mathsf{LocalCorrect}(g, x)$, which results in an incorrect value with probability at most $\delta/2q$, as noted above. Recall that Corollary 26 guarantees that we can output $g(x)$ correctly with probability $1 - \delta/2q$ with only a $\mathsf{poly}(k, \log(q/\delta))$ queries to f. Since we only ever make at most q queries to our coordinate oracles, we can assume that $\mathsf{LocalCorrect}(g, x) = \mathsf{Dict}_i(x)$ in all queries. This happens with probability at least $1 - \delta$ by the union bound. ◀

Therefore, for the rest of this paper, we will assume that we have oracle access to *exact dictators.*

4.2 Implicit Access to an Underlying Junta

An important consequence of having coordinate oracles is that it allows us to reduce the input size of the function dramatically. Suppose $f : \{\pm 1\}^n \to \{\pm 1\}$ and we have $\mathcal{D} = \{g_1, \ldots, g_{k'}\}$ are randomized algorithms that for any $x \in \{\pm 1\}^n$ output $g_i(x) = \mathsf{Dict}_{j_i}(x) = x_{j_i}$. We have that $j_1, \ldots, j_{k'} \in [n]$ are a set of k' distinct coordinates. Let $U = \{j_1, \ldots, j_{k'}\}$. We want to get access to the following function: $g(x_1, \ldots, x_{k'}) = \mathbf{E}[f(y) | y_{j_1} = x_1, y_{j_2} = x_2, \ldots, y_{j_{k'}} = x_{k'}]$. More precisely, given $x_1, \ldots, x_{k'}$ we want to sample uniformly from all $y \in \{\pm 1\}^n$ that satisfy $y_{j_1} = x_1, y_{j_2} = x_2, \ldots, y_{j_{k'}} = x_{k'}$ and apply f on this y.

The following algorithm that runs in $\mathsf{poly}(k, \log(1/\delta))$ time samples y from such a distribution.

Algorithm 1 Sampling a uniformly random input consistent with the oracles' values.

```
   Input: $f$ (target function), $\mathcal{D} = \{g_1, \ldots, g_{k'}\}$ (coordinate oracles),
          $(x_1, \ldots, x_{k'}) \in \{\pm 1\}^{k'}$
   Output: A vector $y \in \{\pm 1\}^n$ with $(g_1(y), \ldots, g_{k'}(y)) = (x_1, \ldots, x_{k'})$
1  Sample $y \sim \{\pm 1\}^n$ and let $z \in \{\pm 1\}^{k'}$ be the vector of evaluations of $\{g_1, \ldots, g_{k'}\}$ on
     $y$;
2  while $z \neq x$ do
3      repeat
4          Let $y'$ be a copy of $y$, but flip each bit independently with probability $\frac{1}{k'}$;
5          Let $z'$ be the vector of evaluations of $\{g_1, \ldots, g_{k'}\}$ on $y'$;
6      until $\mathsf{dist}(x, z') < \mathsf{dist}(x, z)$
7      $y = y'$;
8      $z = z'$;
9  return $y$
```

▶ **Theorem 31.** *Algorithm 1 with probability* $1 - \delta$ *runs in time* $\mathsf{poly}(k', \log(1/\delta))$.

Proof. We focus on the number of iterations of the inner repeat loop. Given (y, z) with $z \neq x$ we analyze the time it takes to find a (y', z') with $\mathsf{dist}(z', x) < \mathsf{dist}(z, x)$. Since $x \neq z$ without loss of generality we can assume that $x_1 \neq z_1$. To get (y', z') with $\mathsf{dist}(z', x) < \mathsf{dist}(z, x)$, it suffices to sample a vector y' with $y'_{j_1} = x_1$ and $y'_{j_2} = y_{j_2}, y'_{j_3} = y_{j_3}, \ldots, y'_{j_{k'}} = y_{j_{k'}}$. Indeed, since we are flipping each coordinate with probability $1/k'$ the probability of sampling such a y' is exactly $1/k' \cdot (1 - 1/k')^{k'-1} \geq 1/(ek')$. Thus, we get that the runtime of the repeat loop is stochastically dominated by a geometric random variable with success probability $1/(ek')$. Thus with probability at least $1 - \delta/k'$, it finishes after $O(k' \cdot \log(k'/\delta))$ iterations. We run the inner repeat loop at most k'-times, thus by union bound, with probability at least $1 - \delta$ the entire process end after at most $O(k'^2 \cdot \log(k'/\delta))$ executions of line 5. We note that execution line 5 actually requires k' queries to $g_1, \ldots, g_{k'}$, each of them takes $\mathsf{poly}(k) = \mathsf{poly}(k')$ time. thus overall, with probability at least $1 - \delta$, our algorithm run in time $\mathsf{poly}(k', \log(1/\delta))$. ◀

▶ **Theorem 32.** *Algorithm 1 samples uniformly from the set of inputs* $\{y' : (g_1(y'), \ldots, g_{k'}(y')) = (x_1, \ldots, x_{k'}))\}$.

Proof. Let $U = \{j_1, \ldots, j_{k'}\}$ be the set of coordinates for which $\{g_1, \ldots, g_{k'}\}$ are oracles to. Algorithm 1 certainly samples a vector y with $y_{j_1} = x_1, \ldots, y_{j_{k'}} = x_{k'}$. We want to show additionally that Algorithm 1 samples $y_{\overline{U}}$ uniformly at random. In fact, at any point in the algorithm the distribution over $y_{\overline{U}}$ is uniform. This is clearly true in the first step where $y \sim \{\pm 1\}^n$, and remains true along the algorithm as we apply independent noise to coordinates in $\overline{U}$ and decide whether to apply the noise or not according to the value of y_U which is independent of $y_{\overline{U}}$. ◀

We will consider algorithms computing non-Boolean function like $g = f_{\mathsf{avg},S}$ for some subset $S \subseteq [n]$. Note that g is a function whose range in $[-1,1]$, but not necessarily a Boolean function.

▶ **Theorem 33** (Formal version of Theorem 5). *Let $f : \{\pm 1\}^n \to \{\pm 1\}$, $\mathcal{D} = \{g_1, \ldots, g_{k'}\}$ be a set of coordinate oracles. Let g be a function from $\{\pm 1\}^{k'} \to [-1,1]$ defined by $g(x) = \mathbf{E}[f(y) | g_1(y) = x_1, \ldots, g_{k'}(y) = x_{k'}]$. Then g has a randomized algorithm in the sense of Definition 15 computing it that runs in expected time* $\mathsf{poly}(k')$.

Proof. Given $x = (x_1, \ldots, x_{k'})$ apply Algorithm 1 on f, $\mathcal{D}$ and x to get a vector $y \in \{\pm 1\}^n$. Return $f(y)$. It is clear that since y is a uniform input subject to $g_1(y) = x_1, \ldots, g_{k'}(y) = x_{k'}$ that our algorithm is a randomized algorithm for g. ◀

4.3 Influential Coordinate Oracles

As above, denote as $\mathcal{S}$ the superset of the low-degree influential coordinates of f, and $\mathcal{D}$ as the set of approximate oracles to said coordinates, obtained via Corollary 26 with parameter $\nu = 0.1$. As we discussed in Section 4.1, we assume (with a small loss in error probability, and a small multiplicative factor on query complexity) that we have exact access to dictators for each influential coordinate. We work towards proving the following improved version of a corollary that appeared in [13]:

The idea will be to take $\mathcal{D}$, a set of $k' = \mathsf{poly}\left(k, \frac{1}{\varepsilon}\right)$ coordinate oracles, and somehow "prune" it down to a set $\mathcal{D}'$ of at most $O(\frac{k}{\varepsilon^2})$ coordinate oracles, such that that the loss in the most correlated junta on this smaller set of coordinates is at most ε

$$\max_{g \in \mathcal{J}_{\mathcal{D},k}} \mathbf{E}[fg] - \max_{g \in \mathcal{J}_{\mathcal{D}',k}} \mathbf{E}[fg] \leq \varepsilon.$$

4.4 Reducing the Number of Oracles to Consider

Starting with a set of $\mathsf{poly}(k/\varepsilon)$ set of oracles $\mathcal{D}$ for a set $\mathcal{S}$ containing the influential coordinates of f, our goal in this section is to prune the number of oracles to $O(k/\varepsilon^2)$ in a way that incurs only a small loss in correlation with the nearest k-junta. [13] achieved their theorem by noting that applying a standard noise operator to f did not affect its proximity to the nearest k-junta significantly, while also guaranteeing that at most $\frac{k^2}{\varepsilon^2}$ coordinates could have large influence. They then were able to estimate the influence of every coordinate in $\mathcal{D}$ despite only having (approximate) oracle access to the influential coordinates, and thus were able to determine which oracles were actually oracles to influential coordinates, of which there were less than k^2/ε^2.

Our approach, as explained at a high level in Section 3, is to estimate the normalized influence of each coordinate in $\mathcal{S}$, which is done via a sequence of random restrictions to f. In words, the below algorithm estimates for each coordinate $i \in \mathcal{S}$ the quantity $\lambda_i^{\approx 2^d} = \mathbf{E}_{(J,z) \sim \mathcal{R}_{2^{-d}}}[\widehat{f_{\bar{J} \to z}}(\{i\})^2]$, where $(J,z) \sim \mathcal{R}_{2^{-d}}$ parameterize a 2^{-d}-random restriction to f. Then, λ_i is defined to be sum over a series of random restrictions $d = 0, ..., \log 10k$ of $\lambda_i^{\approx 2^d}$. The core idea of our algorithm is that this sum over Fourier coefficients on the first level of restricted versions of f is a proxy for $\mathbf{NInf}_i[f]$. In other words, we have the following theorem:

▶ **Theorem 34.** *Let $f : \{\pm 1\}^{k'} \to \mathbb{R}$, where $k' = |\mathcal{D}|$. Let $i \in [k']$. Let*

$$\lambda_i[f] = \sum_{m=0}^{\log(10k)} \lambda_i^{\approx 2^m}[f], \qquad \textit{where} \qquad \lambda_i^{\approx 2^m}[f] = \mathop{\mathbf{E}}_{(J,z)\sim\mathcal{R}_{2^{-m}}}[\widehat{f_{\bar{J}\to z}}(\{i\})^2].$$

Then, $\frac{1}{2}\mathbf{NInf}_i^{\le k}[f] \le \lambda_i[f] \le 2\mathbf{NInf}_i[f]$.

We postpone the proof of Theorem 34 to Section 4.5. The definition of λ_i naturally gives rise to an algorithm for estimating λ_i that we present next. The algorithm would return for each $i \in [k']$ an estimate $\widetilde{\lambda}_i$ that would be close to λ_i with high probability.

Algorithm 2 Estimating λ_i.

```
Input: $f : \{\pm 1\}^{k'} \to [-1, 1]$ along with randomized algorithm A computing $f$ (recall
        Def. 15). Parameters $1 - \delta$ (confidence), $\varepsilon$ (additive error) and $k$.
Output: Estimates $(\widetilde{\lambda}_1, \dots, \widetilde{\lambda}_{k'})$ for $(\lambda_1, \dots, \lambda_{k'})$.
Let $m = \mathsf{poly}(k, k', 1/\varepsilon, \log(1/\delta))$
Initialize $\widetilde{\lambda}_i = 0$ for all $i \in [k']$;
for $d = 0$ to $\log 10k$ do
    Initialize $\widetilde{\lambda}_i^{\approx 2^d} = 0$ for all $i \in [k']$;
    repeat $m$ times
        Let $(J, z) \sim \mathcal{R}_{2^{-d}}$ be a $2^{-d}$-random restriction.
        Estimate $\widehat{f_{\bar{J}\to z}}(\{j\})$ for all $j \in J$ up to additive error $\frac{\varepsilon}{6\log(10k)}$ with
          probability $1 - \delta/\mathsf{poly}(k, k', m)$ using Claim 16 and algorithm A.
        Denote by $\widetilde{f_{\bar{J}\to z}}(\{j\})$ the estimated Fourier coefficient.
        Update $\widetilde{\lambda}_j^{\approx 2^d} = \widetilde{\lambda}_j^{\approx 2^d} + \widetilde{f_{\bar{J}\to z}}(\{j\})^2$ for all $j \in J$.
    Let $\widetilde{\lambda}_i^{\approx 2^d} = \widetilde{\lambda}_i^{\approx 2^d}/m$ for all $i \in [k']$;
Let $\widetilde{\lambda}_i = \sum_d \widetilde{\lambda}_i^{\approx 2^d}$;
return $(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \dots, \widetilde{\lambda}_{k'})$
```

▶ **Lemma 35.** *With probability at least $1 - \delta$ we have that for all $i \in [k']$ it holds that $|\widetilde{\lambda}_i - \lambda_i| \le \varepsilon$.*

Proof. If $j \notin J$ the Fourier coefficient of $\widehat{f_{\bar{J}\to z}}$ is 0 and so our estimate is correct in that case. In the case $j \in J$, each estimation of the Fourier coefficient is correct up to additive error $\eta = \varepsilon/6\log(10k)$ with probability at least $1 - \delta/\mathsf{poly}(k, k', m)$. Thus, we get that $\widetilde{f_{\bar{J}\to z}}(\{j\})^2 = (\widehat{f_{\bar{J}\to z}}(\{j\}) \pm \eta)^2 = \widehat{f_{\bar{J}\to z}}(\{j\})^2 \pm 2\eta|\widehat{f_{\bar{J}\to z}}(\{j\})| \pm \eta^2 = \widehat{f_{\bar{J}\to z}}(\{j\})^2 \pm 3\eta$. Furthermore, we have that $\mathbf{E}_{(J,z)\sim\mathcal{R}_{2^{-d}}}[\widehat{f_{\bar{J}\to z}}(\{j\})^2] = \lambda_j^{\approx 2^d}$, thus by Fact 6 we have that the empirical mean of $m = \mathsf{poly}(1/\varepsilon, \log(k), \log(k'), \log(1/\delta))$ copies of $\widetilde{f_{\bar{J}\to z}}(\{j\})^2$ is within additive error $\varepsilon/(2\log(10k))$ from $\lambda_j^{\approx 2^d}$ with probability at least $1 - \delta/(k'\log(10k))$. By union bound, all these estimates are within the error bound, and we get that $|\widetilde{\lambda}_j^{\approx 2^d} - \lambda_j^{\approx 2^d}| \le 3\eta + \varepsilon/(2\log(10k)) \le \varepsilon/(\log(10k))$. Overall, we get that $|\widetilde{\lambda}_j - \lambda_j| \le \varepsilon$ for all $j \in [k']$ with probability at least $1 - \delta$. ◀

With Algorithm 2 in hand, we are ready to present the pruning procedure.

Algorithm 3 Reduce Number of Oracles.

```
Input: $f$ (target function), $\mathcal{D}$ (influential coordinate oracles, where $\mathcal{D}$ are oracles for $\mathcal{S}$). Parameters $\varepsilon$ and $\delta$.
Output: A subset $\mathcal{D}' \subseteq \mathcal{D}$ of size $O(\frac{k}{\varepsilon^2})$ such that we lose at most $\varepsilon$ in correlation with $f$.
Initialize $\mathcal{D}' = \emptyset$;
Let $m = O((k + \log(1/\delta))/\varepsilon^2)$
repeat $m$ times
    Let $\{g_1, \dots, g_{k'}\} = \mathcal{D} - \mathcal{D}'$, and $\{g_{k'+1}, \dots, g_{|\mathcal{D}|}\} = \mathcal{D}'$
    Sample $z \in \{\pm 1\}^{|\mathcal{D}'|}$. Let $f' : \{\pm 1\}^{k'} \to \mathbb{R}$ be the function defined by
        $f'(x_1, \dots, x_{k'}) = \mathop{\mathbf{E}}_{y \sim \{\pm 1\}^n}[f(y) | g_1(y) = x_1, \dots, g_{k'}(y) = x_{k'}, g_{k'+1}(y) = z_1, \dots, g_{k'+|\mathcal{D}'|}(y) = z_{|\mathcal{D}'|}]$.
    and let A be the randomized algorithm for $f'$ from Theorem 33.
    Apply Algorithm 2 on $f'$ using the randomized algorithm A for $f'$ with confidence $1 - \frac{\delta}{2m}$ and accuracy $\frac{\varepsilon^2}{48 \cdot |\mathcal{S}|} \implies \widetilde{\lambda} = (\widetilde{\lambda}_1, \dots, \widetilde{\lambda}_{k'})$.
    Let our distribution $P$ be defined by $\widetilde{\lambda}$, normalized appropriately.
    Sample $i \sim P$, and add $g_i$ to $\mathcal{D}'$.
return $\mathcal{D}'$
```

▶ **Lemma 36.** *With probability at least $1 - \delta$, Algorithm 3 returns a set of oracles $\mathcal{D}'$ to a subset of coordinates $\mathcal{S}' \subseteq \mathcal{S}$, such that*

$$\max_{g \in \mathcal{J}_{\mathcal{S},k}} \mathbf{E}[fg] - \max_{g \in \mathcal{J}_{\mathcal{S}',k}} \mathbf{E}[fg] \leq \varepsilon.$$

To prove Lemma 36, which tells us our algorithm succeeds and directly implies Theorem 4, we will need a few more lemmas.

We denote the event $\mathcal{E}$ that in the entire execution of Algorithm 3 all $\widetilde{\lambda}_i$ were $\varepsilon^2/(48 \cdot |\mathcal{S}|)$ close to the real λ_i. We note that by union bound this event happens with probability at least $1 - \delta/2$.

Suppose T is the (unknown) set of k oracles for which the best-k junta approximating f is a junta on T. We want to show that our algorithm either samples all the coordinates in T, or it samples a subset T' of T that captures all but $\varepsilon^2/4$ of the Fourier mass of f on T.

▷ Claim 37. Assume the event $\mathcal{E}$ happens. Then, with probability at least $1 - \delta/2$, after m iterations, we will have either:

1. sampled i for all $i \in T$, our target set;
2. sampled i for all $i \in T' \subseteq T$, where $\sum_{S \subseteq T'} \widehat{f}(S)^2 \geq \sum_{S \subseteq T} \widehat{f}(S)^2 - \varepsilon^2/4$.

Proof. In each iteration, assume we have not yet satisfied either items. Let V be the subset of coordinates in T that we have not yet sampled. Let $T' = T \setminus V$. By assumption,

$$\varepsilon^2/4 < \sum_{S \subseteq T} \widehat{f}(S)^2 - \sum_{S \subseteq T'} \widehat{f}(S)^2 = \sum_{S \subseteq T : S \cap V \neq \emptyset} \widehat{f}(S)^2.$$

Let $\mathcal{S}'' = \mathcal{S} \setminus \mathcal{S}'$. We have that $|\mathcal{S}''| = k'$. Now note that up to relabeling of coordinates f' from Algorithm 3 is the same as $(f_{\mathsf{avg},\mathcal{S}})_{\mathcal{S}' \to z}$, where z was randomly chosen. For brevity, denote by $f_z = (f_{\mathsf{avg},\mathcal{S}})_{\mathcal{S}' \to z}$. Note that for any fixed z, f_z is a function that depends only on the coordinates in $\mathcal{S}''$.

By Fact 19, we have

$$\mathop{\mathbf{E}}_{z}\left[\sum_{\emptyset\neq S\subseteq V}\widehat{f_z}(S)^2\right]=\sum_{R:\emptyset\neq(R\cap\mathcal{S}'')\subseteq V}\widehat{f_{\mathsf{avg},\mathcal{S}}}(R)^2=\sum_{\substack{R\subseteq\mathcal{S}:\\ \emptyset\neq(R\cap\mathcal{S}'')\subseteq V}}\widehat{f}(R)^2\geq\sum_{R\subseteq T:R\cap V\neq\emptyset}\widehat{f}(R)^2$$
$$>\varepsilon^2/4. \qquad (2)$$

Next, by applying Theorem 34, for any fixed z, we have

$$\sum_{i\in V}\lambda_i[f_z]\geq\frac{1}{2}\sum_{i\in V}\mathbf{NInf}_i^{\leq k}[f_z]\geq\frac{1}{2}\sum_{\emptyset\neq S\subseteq V}\widehat{f_z}(S)^2.$$

By the assumption that $\mathcal{E}$ happens, the $\widetilde{\lambda_i}$ are $\frac{\varepsilon^2}{48\cdot|\mathcal{S}|}$-accurate, and we get that

$$\sum_{i\in V}\widetilde{\lambda_i}[f_z]\geq\frac{1}{2}\sum_{\emptyset\neq S\subseteq V}\widehat{f_z}(S)^2-\frac{\varepsilon^2}{48\cdot|\mathcal{S}|}\cdot|V|\geq\frac{1}{2}\sum_{\emptyset\neq S\subseteq V}\widehat{f_z}(S)^2-\frac{\varepsilon^2}{48}.$$

On the other hand by applying Theorem 34 again we see that

$$\sum_{i\in\mathcal{S}''}\lambda_i[f_z]\leq 2\cdot\sum_{i\in\mathcal{S}''}\mathbf{NInf}_i[f_z]=2\cdot\mathbf{Var}[f_z]\leq 2$$

and thus $\sum_{i\in\mathcal{S}''}\widetilde{\lambda_i}[f]\leq 2+k'\cdot\frac{\varepsilon^2}{48\cdot|\mathcal{S}|}\leq 2+\frac{\varepsilon^2}{48}\leq 3$ (under the assumption that $\mathcal{E}$ happens). Overall, the probability to sample an element from V is at least

$$\frac{1}{3}\cdot\left(\frac{1}{2}\sum_{\emptyset\neq S\subseteq V}\widehat{f_z}(S)^2-\frac{\varepsilon^2}{48}\right)=\frac{1}{6}\sum_{\emptyset\neq S\subseteq V}\widehat{f_z}(S)^2-\frac{\varepsilon^2}{3\cdot 48}$$

By taking expectation over z, and using Equation (2) we see that the probability to sample an element from V overall is at least

$$\mathop{\mathbf{E}}_{z}\left[\frac{1}{6}\sum_{\emptyset\neq S\subseteq V}\widehat{f_z}(S)^2-\frac{\varepsilon^2}{3\cdot 48}\right]\geq\frac{1}{6}\cdot\frac{\varepsilon^2}{4}-\frac{\varepsilon^2}{3\cdot 48}>\frac{\varepsilon^2}{30}.$$

We get that in each iteration as long as we don't satisfy Items (1) and (2) above, we sample an element from $i\in T$ with probability at least $\varepsilon^2/30$. By repeating the process $m=O(\frac{k+\log(1/\delta)}{\varepsilon^2})$ times we would sample all of T, or get stuck at some T' satisfying Item (2), with probability at least $1-\delta/2$, using Fact 6. ◁

Next, we show that finding T' is almost as good as finding T in the sense that the best correlation by juntas-on-T' with f is up to small additive error the best correlation by juntas-on-T with f.

▶ **Lemma 38.** *Suppose we have some subset T such that $\sum_{S\subseteq T}\widehat{f}(S)^2=c$, and we then identified a subset $T'\subseteq T$ such that $\sum_{S\subseteq T'}\widehat{f}(S)^2\geq c-\frac{\varepsilon^2}{4}$. Then*

$$\left|\max_{g\in\mathcal{J}_{T,k}}\mathbf{E}[fg]-\max_{g\in\mathcal{J}_{T',k}}\mathbf{E}[fg]\right|\leq\varepsilon$$

Proof. We know that $\mathrm{argmax}_{g\in\mathcal{J}_{T,k}} E[fg] = \mathsf{sgn}(f_{\mathsf{avg},T})$ and similarly $\mathrm{argmax}_{g\in\mathcal{J}_{T',k}} E[fg] = \mathsf{sgn}(f_{\mathsf{avg},T'})$. Then we have that

$$\begin{aligned}\left|\max_{g\in\mathcal{J}_{T,k}} \mathbf{E}[fg] - \max_{g\in\mathcal{J}_{T',k}} \mathbf{E}[fg]\right| &= \mathbf{E}[f(x)(\mathsf{sgn}(f_{\mathsf{avg},T}(x_T)) - \mathsf{sgn}(f_{\mathsf{avg},T'}(x_{T'}))] \\ &= \mathop{\mathbf{E}}_{x_T}\left[\mathop{\mathbf{E}}_{x_{\overline{T}}}[f(x_T, x_{\overline{T}})]\,(\mathsf{sgn}(f_{\mathsf{avg},T}(x_T)) - \mathsf{sgn}(f_{\mathsf{avg},T'}(x_{T'}))\right] \\ &= \mathop{\mathbf{E}}_{x_T}\left[f_{\mathsf{avg},T}(x_T)\,(\mathsf{sgn}(f_{\mathsf{avg},T}(x_T)) - \mathsf{sgn}(f_{\mathsf{avg},T'}(x_{T'}))\right] \\ &\le 2\mathop{\mathbf{E}}_{x_T}\left[|f_{\mathsf{avg},T}(x_T) - f_{\mathsf{avg},T'}(x_{T'})|\right] \\ &\qquad (\text{Since } z(\mathsf{sgn}(z) - \mathsf{sgn}(z')) \le 2|z - z'| \text{ for all } z, z' \in \mathbb{R}) \\ &\le 2\sqrt{\mathop{\mathbf{E}}_{x_T}\left[\left(f_{\mathsf{avg},T}(x_T) - f_{\mathsf{avg},T'}(x_{T'})\right)^2\right]} \\ &= 2\sqrt{\sum_{S\subseteq T}\widehat{f}(S)^2 - 2\sum_{S\subseteq T'}\widehat{f}(S)^2 + \sum_{S\subseteq T'}\widehat{f}(S)^2} \\ &\le 2\sqrt{\frac{\varepsilon^2}{4}} = \varepsilon\,. \end{aligned}$$ ◀

Proof of Lemma 36. Let g be the k-junta that maximizes $\mathbf{E}[fg]$ among all k-juntas on $\mathcal{S}$. Let T be the set of variables on which g depends. By Claim 37 we either sample oracles to all of T or to a subset T' for which

$$\sum_{S\subseteq T'}\widehat{f}(S)^2 \ge \sum_{S\subseteq T}\widehat{f}(S)^2 - \varepsilon^2/4.$$

In the second case, by Lemma 38, we incur a loss in correlation of at most ε with our nearest k-junta. In the first case, we lose no correlation with the closest k-junta, and by a union bound our probability of failure is at most δ. ◀

The above concludes the proof of Lemma 36. Finally, Theorem 4 is implied by Lemma 36, as shown below.

▶ **Theorem 39** (Theorem 4, restated). *Let $\varepsilon > 0$, $k \in \mathbb{N}$, and $k' = C(k/\varepsilon^2)$ for some universal constant C. Then, there exists an algorithm that given f, k, ε makes at most poly$(k, 1/\varepsilon)$ queries to f and returns a number α such that with probability at least* 0.99

1. $\alpha \le \max_{g\in\mathcal{J}_{n,k'}} \mathbf{E}[fg] + O(\varepsilon)$
2. $\alpha \ge \max_{g\in\mathcal{J}_{n,k}} \mathbf{E}[fg] - O(\varepsilon)$

Proof. Set $\delta = 2^{-\mathsf{poly}(k,1/\varepsilon)}$. We first apply Corollary 26 from [13]. This gives us $\mathsf{poly}(k, \frac{1}{\varepsilon}, \log(1/\delta)) = \mathsf{poly}(k/\varepsilon)$ coordinate oracles $\mathcal{D}$ to coordinates $\mathcal{S}$ that includes all coordinates i with $\mathbf{Inf}_i^{\le k}[f] \ge \frac{\varepsilon^2}{k}$. By Claim 29 we see that

$$\max_{g\in\mathcal{J}_{\mathcal{S},k}} \mathbf{E}[fg] \ge \max_{g\in\mathcal{J}_{n,k}} \mathbf{E}[fg] - \varepsilon$$

Next, we apply Algorithm 3 to get a subset $\mathcal{D}' \subseteq \mathcal{D}$ to coordinates $\mathcal{S}' \subseteq \mathcal{S}$ such that with high probability

$$\max_{g\in\mathcal{J}_{\mathcal{S}',k}} \mathbf{E}[fg] \ge \max_{g\in\mathcal{J}_{\mathcal{S},k}} \mathbf{E}[fg] - \varepsilon$$

We take α to be the estimation of the correlation of the best junta on $\mathcal{S}'$ with f. By Claim 12 we have that $\max_{g\in\mathcal{J}_{\mathcal{S}'}}\mathbf{E}[fg] = \mathbf{E}[|f_{\mathsf{avg},\mathcal{S}'}(x)|]$. To estimate the latter, we use a randomized algorithm that computes $f_{\mathsf{avg},\mathcal{S}'}$ given by Theorem 33. We randomly sample $O(1/\varepsilon^2)$ many values for x and estimate for each of them $|f_{\mathsf{avg},\mathcal{S}'}(x)|$ up to additive error $\varepsilon/2$ via the randomized algorithm with expected value $f_{\mathsf{avg},\mathcal{S}'}(x)$.

Assume that α is a ε-additive approximation to $\max_{g\in\mathcal{J}_{\mathcal{S}'}}\mathbf{E}[fg]$. In this case, we claim that α satisfies both items from the theorem's statement. Indeed,

1. $\alpha \le \max_{g\in\mathcal{J}_{\mathcal{S}'}}\mathbf{E}[fg] + \varepsilon \le \max_{g\in\mathcal{J}_{n,k'}}\mathbf{E}[fg] + \varepsilon$.
2. $\alpha \ge \max_{g\in\mathcal{J}_{\mathcal{S}'}}\mathbf{E}[fg] - \varepsilon \ge \max_{g\in\mathcal{J}_{\mathcal{S}',k}}\mathbf{E}[fg] - \varepsilon \ge \max_{g\in\mathcal{J}_{\mathcal{S},k}}\mathbf{E}[fg] - 2\varepsilon \ge \max_{g\in\mathcal{J}_{n,k}}\mathbf{E}[fg] - 3\varepsilon$.

Next, we analyze the number of queries of our algorithm. Obtaining the initial set of coordinate oracles $\mathcal{D}$ takes $\mathsf{poly}(k, 1/\varepsilon, \log(1/\delta)) = \mathsf{poly}(k, 1/\varepsilon)$ queries. Then, we go on to run Algorithm 3 that makes $m = O((k + \log(1/\delta))/\varepsilon^2)$ iterations, each making $\mathsf{poly}(k, 1/\varepsilon, \log(1/\delta))$ queries. Next, to estimate $\mathbf{E}[|f_{\mathsf{avg},\mathcal{S}'}(x)|]$ we require $\mathsf{poly}(1/\varepsilon)$ samples from randomized algorithm for $f_{\mathsf{avg},\mathcal{S}'}(x)$ each such sample translate to $\mathsf{poly}(k, 1/\varepsilon)$ samples to f. Finally, we note that each "query" to an oracle incurs an overhead of $\mathsf{poly}(\log(k, 1/\varepsilon))$ queries to f along with an $o(1)$ additive loss in confidence by Corollary 30. Overall, we make $\mathsf{poly}(k, 1/\varepsilon)$ queries. ◀

4.5 Proof of Theorem 34

We now present the proof of Theorem 34.

Proof of Theorem 34. We express λ_i in terms of the Fourier spectrum of f. Using Fact 19,

$$\begin{aligned}\lambda_i &= \sum_{m=0}^{\log(10k)} \sum_{S:S\ni i} \widehat{f}(S)^2 \cdot \Pr_{J\subseteq_{2^{-m}}[k']}[S\cap J = \{i\}] \\ &= \sum_{m=0}^{\log(10k)} \sum_{S:S\ni i} \widehat{f}(S)^2 \cdot \Pr_{J\subseteq_{2^{-m}}[k']}[|S\cap J| = 1]\cdot\frac{1}{|S|} \\ &= \sum_{S:S\ni i} \frac{\widehat{f}(S)^2}{|S|}\cdot \sum_{m=0}^{\log(10k)} \Pr_{J\subseteq_{2^{-m}}[k']}[|S\cap J| = 1]\end{aligned}$$

It therefore suffices to show that for any non-empty set S such that $|S|\le k$ it holds that

$$\frac{1}{2} \le \sum_{m=0}^{\log(10k)} \Pr_{J^{(m)}\subseteq_{2^{-m}}[k']}[|S\cap J^{(m)}| = 1] \le 2\,. \tag{3}$$

From which it is clear that $\lambda_i \le 2\cdot\sum_{S:S\ni i}\frac{\widehat{f}(S)^2}{|S|} = 2\cdot\mathbf{NInf}_i[f]$ and similarly $\lambda_i \ge \frac{1}{2}\sum_{\substack{S\ni i,\\|S|\le k}}\frac{\widehat{f}(S)^2}{|S|} = \frac{1}{2}\mathbf{NInf}_i^{\le k}[f]$.

We move to prove Equation (3). The first observation is that an equivalent way to sample $J^{(m)}\subseteq_{2^{-m}}[k']$ is to sample m independent set $J_1^{(m)},\ldots,J_m^{(m)}\subseteq_{1/2}[k']$ and take their intersection $J^{(m)} = J_1^{(m)}\cap\cdots\cap J_m^{(m)}$. Furthermore, by linearity of expectation

$$\begin{aligned}\sum_{m=0}^{\infty} \mathop{\mathbf{Pr}}_{J^{(m)} \subseteq_{2^{-m}} [k']}[|S \cap J| = 1] &= \sum_{m=0}^{\infty} \mathop{\mathbf{E}}_{\substack{J_1^{(m)} \subseteq_{1/2} [k'],\\ J_2^{(m)} \subseteq_{1/2} [k'],\\ \cdots}} \left[\mathbb{1}_{|S \cap J_1^{(m)} \cap \cdots \cap J_m^{(m)}| = 1}\right] \\ &= \mathop{\mathbf{E}}_{\substack{J_1 \subseteq_{1/2} [k'],\\ J_2 \subseteq_{1/2} [k'],\\ \cdots}} \left[\sum_{m=0}^{\infty} \mathbb{1}_{|S \cap J_1 \cap \cdots \cap J_m| = 1}\right]\end{aligned}$$

which in essence means that the choices for $J_1^{(1)}, J_1^{(2)}, \ldots$ can be the same set J_1, and similarly for any J_i.

To analyze the latter expectation, we note that it can be described as the expected value of the following random process:

```
$X \leftarrow 0$
for $i = 1, 2, \ldots, \log(10k)$ do
    if $S = \emptyset$ then
        halt!;
    if $|S| = 1$ then
        increment $X$;
    Sample $J_i \subseteq_{1/2} [k']$;
    $S \leftarrow S \cap J_i$;
```

It therefore suffices to show that the expected value of the above random process is bounded in $[1/2, 2]$. In the analysis, we consider also the infinite horizon process that keeps on going until $S = \emptyset$. We observe that the expected values of both processes depend only on the size of the initial S from symmetry. For any $t \in \{0, 1, \ldots, k'\}$, denote by F_t the expected value of the infinite horizon process starting with a set S of size t. For the finite horizon process with i iterations, we let the expected value be denoted by $F_t^{(i)}$. We observe that $F_0 = 0$, and furthermore that $F_1 = 2$ since starting from a set of size 1 the random variable X would behave like geometric random variable with $p = 1/2$. Similarly, $F_1^{(i)} = 2 - \frac{1}{2^{i-1}}$ as it is the minimum of i and a geometric random variable with $p = 1/2$.

Furthermore, for the infinite horizon process, we observe that we have the following recurrence

$$F_t = \sum_{a=0}^{t} \frac{\binom{t}{a}}{2^t} \cdot F_a,$$

for $t \geq 2$ or equivalently

$$F_t \cdot (1 - 2^{-t}) = \sum_{a=0}^{t-1} \frac{\binom{t}{a}}{2^t} \cdot F_a.$$

We show by induction that $1/2 < F_t^{(\log 10k)} \leq 2$ for $t \geq 1$. The base case $t = 1$ was discussed above. Applying the induction hypothesis we have

$$F_t \cdot (1 - 2^{-t}) = \sum_{a=0}^{t-1} \frac{\binom{t}{a}}{2^t} \cdot F_a \leq \sum_{a=0}^{t-1} \frac{\binom{t}{a}}{2^t} \cdot 2 \leq (1 - 2^{-t}) \cdot 2.$$

Dividing both sides by $(1 - 2^{-t})$ gives the inequality $F_t \leq 2$, which implies that $F_t^{(\log 10k)} \leq 2$.

For the lower bound, we consider the indicator random variable $Y_t^{(i)}$, where $t = |S|$, which equals 1 if $|S| = 1$ at some point during the above process before iteration i. We note that $Y_t^{(\log 10k)}$ is a lower bound for the value of X in the finite horizon process, and Y_t is a lower bound for the value of X at the end of the infinite horizon process. First, we claim that $\mathbf{E}[Y_t] = \mathbf{Pr}[Y_t = 1] \geq 2/3$ for all $t \geq 1$. The base case of $t = 1$ is certainly true, and we also have, similar to before, that

$$\begin{aligned}\mathbf{E}[Y_t] \cdot (1 - 2^{-t}) &= \sum_{a=0}^{t-1} \frac{\binom{t}{a}}{2^t} \mathbf{E}[Y_a] \\ &\geq 0 \cdot \frac{1}{2^t} + 1 \cdot \frac{t}{2^t} + \frac{2}{3} \cdot \underbrace{\sum_{a=2}^{t-1} \frac{\binom{t}{a}}{2^t}}_{1 - \frac{2+t}{2^t}} \\ &= \frac{2}{3} + \frac{t - \frac{2}{3}(2+t)}{2^t} \geq \frac{2}{3} + \frac{t/3 - 4/3}{2^t} \geq \frac{2}{3} - \frac{2/3}{2^t} = \frac{2}{3} \cdot (1 - 2^{-t})\end{aligned}$$

which holds for all $t \geq 2$, and thus $\mathbf{Pr}[Y_t = 1] \geq 2/3$. However, this only holds for the infinite horizon random process. Let A be the event that $S = \emptyset$ by iteration $\log 10k$, and note that $\mathbf{Pr}[A] = \mathbf{Pr}[\mathrm{Bin}(|S|, \frac{1}{10k}) = 0] \geq \mathbf{Pr}[\mathrm{Bin}(k, \frac{1}{10k}) = 0] = \left(1 - \frac{1}{10k}\right)^k \geq 1 - \frac{k}{10k} = 0.9$. Finally, we claim that for all $t \geq 2$ we have that $\mathbf{Pr}[Y_t^{(\log 10t)}] \geq 1/2$. Note that for Y_t to happen, it must be the case that either $\overline{A}$ happens or $Y_t^{(\log 10t)}$ happens. Thus, by a union bound

$$\tfrac{2}{3} \leq \mathbf{Pr}[Y_t = 1] \leq \mathbf{Pr}[Y_t^{(\log 10t)} = 1] + \mathbf{Pr}[\overline{A}] \leq \mathbf{Pr}[Y_t^{(\log 10t)} = 1] + 0.1 \ ,$$

which implies $\mathbf{Pr}[Y_t^{(\log 10t)} = 1] > 1/2$. Finally, $F_t^{(\log 10t)} \geq \mathbf{Pr}[Y_t^{(\log 10t)} = 1] > 1/2$ as desired. ◀

5 A $2^{\widetilde{O}(\sqrt{k})}$-query Tolerant Junta Tester

In this section, we prove Theorem 2. Throughout this section, we assume that we already applied Algorithm 3 to reduce the number of coordinate oracles to $O(k/\varepsilon^2)$. We denote by $\mathcal{D}$ the set of oracles we get, and by $\mathcal{S} \subseteq [n]$ the set of coordinate to which they are oracles to. Suppose that the best k-junta approximation of f is a junta-on-T, for a set $T \subseteq \mathcal{S}$ of size k. We call T the "target set". Note that T is unknown to the algorithm, and in fact, identifying T (or a close approximation to T) from all subsets of size k of $\mathcal{S}$ is the crux of the problem.

We start with the observation that if we were somehow able to identify all of the variables of T that capture most of the Fourier mass above level κ, then we could simply restrict f by randomly fixing these variables, leaving us with the task of identifying the best k-junta approximation of f, given that we know the best k-junta has most its Fourier mass below level κ. For the latter case, there are only $\binom{|\mathcal{S}|}{\kappa}$ Fourier coefficients to estimate, and estimating these to sufficient accuracy allows one to estimate the the correlation f has with any subset $U \subseteq \mathcal{S}$ such that $|U| \leq k$.

We are now ready to present the details of the algorithm. The algorithm can be broken down into two main steps. First, we find, with high probability, a set $B \subseteq T$ that captures almost all Fourier mass of T above level κ. This first step, which we call "phase one", closely resembles the techniques in Section 4 in that we utilize a series of random restrictions to estimate normalized influences. The main difference is that rather than considering normalized influences of individual coordinates, we now consider normalized influences of sets of size κ. The goal of phase one is to produce at least one subset B of our target set T

which effectively captures most of the Fourier mass within T above level κ. Once we have done that, we have reduced to the scenario of the closest k-junta to f having most of its Fourier mass below level κ, which can be solved via estimating all of the Fourier coefficients below level κ.

5.1 Phase One: The Higher Levels

First, we prove an analogous theorem to Theorem 34, which relates $\lambda_U[f]$ to $\mathbf{NInf}_U[f]$ for all U:

▶ **Theorem 40.** *Let $f : \{\pm 1\}^\ell \to \mathbb{R}$. Let $U \subseteq [\ell]$, where $\ell = |\mathcal{D}|$ and $|U| \leq k$. Let*

$$\lambda_U[f] = \sum_{m=0}^{2|U|\log(10k)} \lambda_U^{\approx p^{-m}}[f], \qquad \textit{where} \qquad \lambda_U^{\approx p^{-m}}[f] = \mathop{\mathbf{E}}_{(J,z)\sim\mathcal{R}_{p^m}}[\widehat{f_{\bar{J}\to z}}(U)^2]$$

for $p = 1 - \frac{1}{2|U|}$. Then, $\frac{1}{2} \cdot \mathbf{NInf}_U^{\leq k}[f] \leq \lambda_U[f] \leq 3 \cdot \mathbf{NInf}_U[f]$.

Again, we postpone the proof of this to the end of this section in Section 5.3. The definition of $\lambda_U[f]$ is naturally algorithmic, and therefore we can design the following algorithm to approximate the values of $\lambda_U[f]$ for all sets U of size $\kappa = \sqrt{\varepsilon k}$.

Algorithm 4 Estimating λ_U's.

```
Input: $f : \{\pm 1\}^{k'} \to [-1, 1]$ along with a randomized algorithm A computing $f$
       (recall Def. 15). Parameters $1 - \delta$ (confidence), $\varepsilon$ (additive error) and $k$.
Output: Estimates $\{\widetilde{\lambda}_U\}_{|U|=\kappa}$ for $\{\lambda_U\}_{|U|=\kappa}$.
Let $m = \mathsf{poly}(k, k', 1/\varepsilon, \log(1/\delta))$
Initialize $\widetilde{\lambda}_U = 0$ for all $U \subseteq [k']$, $|U| = \kappa = \sqrt{\varepsilon k}$
Let $p = \left(1 - \frac{1}{2\kappa}\right)$
for $d = 0$ to $2\kappa \log 10k$ do
    Initialize $\widetilde{\lambda}_U^{\approx p^{-d}} = 0$ for all $U \subseteq [k']$ such that $|U| = \kappa$
    repeat $m$ times
        Let $(J, z) \sim \mathcal{R}_{p^d}$ be a $p^d$-random restriction.
        Estimate $\widehat{f_{\bar{J}\to z}}(U)$ for all $U \subseteq J$ of size $\kappa$ up to additive error $\frac{\varepsilon}{12\kappa\log(10k)}$ with
         probability $1 - \frac{\delta}{\binom{k'}{\kappa} m \cdot 2\kappa \log(10k)}$ using Claim 16 and algorithm A. Denote by
         $\widetilde{f_{\bar{J}\to z}}(U)$ the estimated Fourier coefficient.
        Update $\widetilde{\lambda}_U^{\approx p^{-d}} = \widetilde{\lambda}_U^{\approx p^{-d}} + \widetilde{f_{\bar{J}\to z}}(U)^2$ for all $U \subseteq J$ of size $\kappa$.
    Let $\widetilde{\lambda}_U^{\approx p^{-d}} = \widetilde{\lambda}_U^{\approx p^{-d}}/m$ for all $U \subseteq J$ of size $\kappa$;
Let $\widetilde{\lambda}_U = \sum_d \widetilde{\lambda}_U^{\approx p^{-d}}$;
return $\{\widetilde{\lambda}_U\}_{|U|=\kappa}$
```

▶ **Lemma 41.** *With probability at least $1 - \delta$ we have that for all $U \subseteq [k']$ of size κ it holds that $|\widetilde{\lambda}_U - \lambda_U[f]| \leq \varepsilon$.*

Proof. This proof closely follows that of Lemma 35. If $U \not\subseteq J$ the Fourier coefficient of $\widehat{f_{\bar{J}\to z}}(U)$ is 0 and so our estimate is correct in that case. In the case $U \subseteq J$, each estimation of the Fourier coefficient is correct up to additive error $\eta = \frac{\varepsilon}{12\kappa\log(10k)}$ with probability at least $1 - \delta/\exp(k, k', m)$. Thus, we get that $\widetilde{f_{\bar{J}\to z}}(U)^2 = (\widehat{f_{\bar{J}\to z}}(U) \pm$

$\eta)^2 = \widehat{f_{\bar{J}\to z}}(U)^2 \pm 2\eta|\widehat{f_{\bar{J}\to z}}(U)| \pm \eta^2 = \widehat{f_{\bar{J}\to z}}(U)^2 \pm 3\eta$. Furthermore, we have that $\mathbf{E}_{(J,z)\sim\mathcal{R}_{p^d}}[\widehat{f_{\bar{J}\to z}}(U)^2] = \lambda_U^{\approx p^{-d}}$, thus by Fact 6 we have that the empirical mean of $m = \mathsf{poly}(1/\varepsilon, \mathsf{poly}(k), \mathsf{poly}(k'), \log(1/\delta))$ copies of $\widetilde{f_{\bar{J}\to z}}(U)^2$ is within additive error $\varepsilon/(4\kappa\log(10k))$ from $\lambda_U^{\approx p^{-d}}$ with probability at least $1 - \frac{\delta}{\binom{k'}{\kappa} m\cdot 2\kappa\log(10k)}$. By union bound, all these estimates are within the error bound, and we get that

$$\left|\widetilde{\lambda}_U^{\approx p^{-d}} - \lambda_U^{\approx p^{-d}}\right| \leq 3\eta + \varepsilon/(4\kappa\log(10k)) \leq \varepsilon/(2\kappa\log(10k)).$$

Overall, we get that $|\widetilde{\lambda}_U - \lambda_U[f]| \leq \varepsilon$ for all $|U| = \kappa$ with probability at least $1-\delta$. ◀

Since we are sampling sets of size κ, we need to sample at most $k/\kappa = \sqrt{k/\varepsilon} =: \alpha$ distinct subsets of T of size κ in order to capture all the potential mass of T above level κ.

Algorithm 5 Branching Process.

```
Input: $f$ (target function), $\mathcal{D}$ (where $\mathcal{D}$ are coordinate oracles for $\mathcal{S}$) a current depth
       $t$, a current subset $\mathcal{D}' \subseteq \mathcal{D}$ of coordinate oracles, $\varepsilon$, $\delta$
Output: Return collection of subsets of $\mathcal{D}$ of size at most $k$.
Let $\alpha = k/\kappa = \sqrt{k/\varepsilon}$
Let $r = O(1/\varepsilon^2)$ and $\ell = 2(r+1)^{3\alpha+\log(2/\delta)}$
/* $r+1$ is the branching factor, and $\ell$ is an upper bound on the number
   of nodes in the branching process (the process depth is
   $3\alpha+\log(2/\delta)$).                                                */
if $t = 3\alpha+\log(2/\delta)$ or $|\mathcal{D}'| > k-\kappa$ then
    return $\{\mathcal{D}'\}$
Let $\{g_1, ..., g_{k'}\} = \mathcal{D} - \mathcal{D}'$ and $\{g_{k'+1}, ..., g_{|\mathcal{D}|}\}$ where $k' = |\mathcal{D}| - |\mathcal{D}'|$
Sample $z \in \{\pm 1\}^{|\mathcal{D}'|}$. Let $f' : \{\pm 1\}^{k'} \to \mathbb{R}$ be the function defined by
    $f'(x_1, \ldots, x_{k'}) =$
      $\mathbf{E}_{y\sim\{\pm 1\}^n}[f(y)|g_1(y) = x_1, \ldots, g_{k'}(y) = x_{k'}, g_{k'+1}(y) = z_1, \ldots, g_{|\mathcal{D}|}(y) = z_{|\mathcal{D}'|}]$,
  and let A be the randomized algorithm for $f'$ from Theorem 33.
Apply Algorithm 4 on $f'$ using the randomized algorithm A for $f'$ with confidence
  $1 - \frac{\delta}{2\ell}$ and accuracy $\frac{\varepsilon^2}{48\cdot\binom{|\mathcal{D}|}{\kappa}} \implies \widetilde{\lambda} = \{\widetilde{\lambda}_U\}_{|U|=\kappa}$.
Let our distribution $P$ be defined by $\widetilde{\lambda}$, normalized appropriately
Sample $M_1, ..., M_r \sim \widetilde{\lambda}$
Let $\mathcal{L} = \{\}$.
for $M = \emptyset, M_1, ..., M_r$ do
    $\mathcal{L} = \mathcal{L} \cup \mathrm{BranchingProcess}(f, \mathcal{D}, t+1, \mathcal{D}' \cup \{g_i : i \in M\}, \varepsilon, \delta)$
return $\mathcal{L}$
```

▶ **Lemma 42.** *With probability at least $1-\delta$, at least one of the subsets Algorithm 5 returns is a set of coordinate oracles to $B \subseteq T$ such that*

$$\mathbf{E}_{z}\left[\sum_{\substack{S\subseteq T\setminus B\\ |S|>\kappa}} \widehat{f_{B\to z}}(S)^2\right] \leq \varepsilon^2/4. \tag{4}$$

The reason for Equation (4) becomes clear in Section 5.2, where we show that assuming the inequality, we lose at most an additive error of $\varepsilon/2$ to the nearest k-junta if we ignore the Fourier mass above level κ after restricting B. As before, in order to prove the above lemma, we prove a claim capturing the algorithm's progress towards satisfying Equation (4).

We denote the event $\mathcal{E}$ that in the entire execution of Algorithm 5 all of the $\widetilde{\lambda}_U$ were $\varepsilon^2/48 \cdot \binom{|\mathcal{D}|}{\kappa}$ close to the real λ_U. We note that by a union bound, this happens with probability at least $1-\delta/2$.

Suppose again that T is the (unknown) set of k coordinates for which the best k-junta approximating f is a junta on T. If T has Fourier mass less than $\varepsilon^2/4$ above level κ then one of the subsets that Algorithm 5 will return is the empty set, which satisfies the claim. Therefore, henceforth we assume that T has at least $\varepsilon^2/4$ Fourier mass above level κ. We show that in such a case, each M_i for $i = 1, \ldots, r$ will be a subset of T with probability at least $\Omega(\varepsilon^2)$.

▷ Claim 43. Assume $\mathcal{D}'$ are coordinate oracles to $\mathcal{S}' \subseteq T$. Suppose also that

$$\mathop{\mathbf{E}}_{z}\left[\sum_{\substack{S\subseteq T\setminus\mathcal{S}'\\|S|>\kappa}} \widehat{f_{\mathcal{S}'\to z}}(S)^2\right] > \varepsilon^2/4.$$

Then, conditioned on $\mathcal{E}$, when running the Branching Process on $\mathcal{D}'$, each M_i will be with probability at least $\varepsilon^2/40$ a collection of κ new coordinate oracles to coordinates in T.

Proof. Similar to the proof of Claim 37, denote by $f_z = (f_{\mathsf{avg},\mathcal{S}})_{\mathcal{S}'\to z}$, and note that f' is up to relabeling of coordinates the same function as f_z. Denote $V \subseteq T$ as the part of the target set we have not yet sampled, so $V = T \setminus \mathcal{S}'$. Then, using our assumption, we have that

$$\begin{aligned}
\varepsilon^2/4 &< \mathop{\mathbf{E}}_{z}\left[\sum_{\substack{S\subseteq V\\|S|>\kappa}} \widehat{f_{\mathcal{S}'\to z}}(S)^2\right]\\
&= \sum_{\substack{S\subseteq V\\|S|>\kappa}} \sum_{\substack{R\subseteq[n]:\\R\cap\overline{\mathcal{S}'}=S}} \widehat{f}(R)^2 && \text{(Fact 19)}\\
&= \sum_{\substack{S\subseteq V\\|S|>\kappa}} \sum_{\substack{R\subseteq\mathcal{S}:\\R\cap\overline{\mathcal{S}'}=S}} \widehat{f}(R)^2 && (\text{if } R\not\subseteq\mathcal{S} \text{ then } R\cap\overline{\mathcal{S}'}\neq S)\\
&= \sum_{\substack{S\subseteq V\\|S|>\kappa}} \sum_{\substack{R\subseteq\mathcal{S}:\\R\cap\overline{\mathcal{S}'}=S}} \widehat{f_{\mathsf{avg},\mathcal{S}}}(R)^2\\
&= \mathop{\mathbf{E}}_{z}\left[\sum_{\substack{S\subseteq V\\|S|>\kappa}} \widehat{f_z}(S)^2\right].
\end{aligned}$$

Next, by applying Theorem 40, we have that

$$\sum_{\substack{U\subseteq V\\|U|=\kappa}} \lambda_U[f_z] \geq \frac{1}{2}\sum_{U\subseteq V} \mathbf{NInf}_U^{\leq k}[f_z] \geq \frac{1}{2}\sum_{U\subseteq V:|U|=\kappa}\ \sum_{S:U\subseteq S\subseteq V} \frac{\widehat{f_z}(S)^2}{\binom{|S|}{|U|}} = \frac{1}{2}\sum_{\substack{|S|>\kappa\\S\subseteq V}} \widehat{f_z}(S)^2$$

Then, using the assumption that $\mathcal{E}$ happens, the $\widetilde{\lambda_U}$ are $\frac{\varepsilon^2}{48\cdot\binom{|S|}{\kappa}}$-accurate, and we get that

$$\sum_{\substack{U\subseteq V\\|U|=\kappa}} \widetilde{\lambda_U}[f_z] \geq \frac{1}{2}\sum_{\substack{|S|>\kappa\\S\subseteq V}} \widehat{f_z}(S)^2 - \frac{\varepsilon^2}{48\cdot\binom{|S|}{\kappa}}\cdot\binom{k}{\kappa} \geq \frac{1}{2}\sum_{\substack{|S|>\kappa\\S\subseteq V}} \widehat{f_z}(S)^2 - \frac{\varepsilon^2}{48}.$$

On the other hand, again by applying Theorem 40, we have that

$$\sum_{\substack{U\subseteq\mathcal{S}\\|U|=\kappa}} \lambda_U[f_z] \le 3 \sum_{\substack{U\subseteq\mathcal{S}\\|U|=\kappa}} \mathbf{NInf}_U[f_z] \le 3\mathbf{W}^{\ge\kappa}[f_z] \le 3.$$

This implies that $\sum_U \widetilde{\lambda_U} \le 3 + \frac{\varepsilon^2}{48\cdot\binom{|S|}{\kappa}} \cdot \binom{|S|}{\kappa} \le 4$. Overall, the probability to sample $U \subseteq V$ is at least

$$\frac{1}{4}\left(\frac{1}{2}\sum_{\substack{|S|>\kappa\\S\subseteq V}} \widehat{f_z}(S)^2 - \frac{\varepsilon^2}{48}\right) = \frac{1}{8}\sum_{\substack{|S|>\kappa\\S\subseteq V}} \widehat{f_z}(S)^2 - \frac{\varepsilon^2}{4\cdot 48}.$$

Taking an expectation over z, we see that the probability to sample a subset of V is at least

$$\mathop{\mathbf{E}}_{z}\left[\frac{1}{8}\sum_{\substack{|S|>\kappa\\S\subseteq V}} \widehat{f_z}(S)^2 - \frac{\varepsilon^2}{4\cdot 48}\right] \ge \frac{1}{8}\cdot\frac{\varepsilon^2}{4} - \frac{\varepsilon^2}{4\cdot 48} \ge \frac{\varepsilon^2}{40}.$$ ◀

We are now ready to prove Lemma 42.

Proof of Lemma 42. By Claim 43, if our special set T has at least $\varepsilon^2/4$ mass on the levels above κ, then if we sample according to our distribution $\widetilde{\lambda} = \{\widetilde{\lambda}_U\}_{|U|=\kappa}$, we will see $U \subseteq T$ with probability at least $\varepsilon^2/40$. Then, if we sample $r = O(\varepsilon^{-2})$ subsets in Algorithm 5, applying the multiplicative Chernoff bound in Fact 6, we see at least one subset of T with probability at least $p \ge 0.9$ each time we sample $M_1, ..., M_r$ in Algorithm 5. In order for Algorithm 5 to successfully find B_i with the desired property, it suffices to have sampled from T at least α times in our branching process. Therefore, we can treat our $N := (3\alpha + \log(2/\delta))$ depth branching process as a $X = \mathrm{Bin}(N, p)$ random variable. Applying a standard Chernoff bound (second case in Fact 6), we have that our probability of failure is

$$\begin{aligned}
\mathbf{Pr}[X < \alpha] &= \mathbf{Pr}[\overline{X} < \tfrac{\alpha}{N}]\\
&= \mathbf{Pr}[\overline{X} < 0.9 - (0.9 - \tfrac{\alpha}{N})]\\
&\le \exp(-2N(0.9 - \tfrac{\alpha}{N})^2) && \text{(Using Fact 6)}\\
&\le \exp(-2N(0.81 - 2\tfrac{\alpha}{N}))\\
&\le \exp(-1.5N + 4\alpha)\\
&\le \exp(-\log(2/\delta)) = \delta/2.
\end{aligned}$$

This shows that, by a union bound with event $\mathcal{E}$, one of the branches of our algorithm find's a B_i satisfying Equation (4) with probability at least $1 - \delta$. ◀

▷ Claim 44. The query complexity of phase one of the algorithm for constant δ (failure probability) is $2^{\widetilde{O}(\sqrt{k/\varepsilon})}$.

Proof. All of our queries to f in phase one come from estimating fourier coefficients using Claim 16 in Algorithm 4. We require that the estimated Fourier coefficients be accurate to within $1/\mathsf{poly}(k, 1/\varepsilon)$ with confidence $1 - O(1/\ell) = 1 - 2^{-\widetilde{\Omega}(\sqrt{k/\varepsilon})}$, which is possible via Fact 6 with query complexity $\mathsf{poly}(k/\varepsilon)$. However, we do this $O(\ell) = 2^{\widetilde{O}(\sqrt{k/\varepsilon})}$ times during the branching process, which yields the final overall query complexity. ◁

5.2 Phase Two: The Lower Levels

Now, we are ready to use Algorithm 5. Our strategy will be to take the subsets outputted from Algorithm 5 one at time, randomly fixing those coordinates, and then treating this restricted version of f as if all its Fourier mass were below level κ (recall that $\kappa = \sqrt{\varepsilon k}$). Let T be the target set of size k on which there exists a k-junta which best approximates f. Assume that the first part of the algorithm is successful in yielding at least one $B \subseteq T$ such that:

$$\mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\Bigg[\sum_{\substack{S\subseteq T\setminus B\\ |S|>\kappa}} \widehat{f_{B\to z}}(S)^2\Bigg] \le \varepsilon^2/4. \tag{5}$$

Let g be the maximizer of $\max_{g'\in\mathcal{J}_T} \mathbf{E}[fg']$. Recall that by Claim 12 we have that $g = \mathsf{sgn}(f_{\mathsf{avg},T})$ and

$$\mathsf{corr}(f,\mathcal{J}_T) = \mathbf{E}[fg] = \mathop{\mathbf{E}}_{y\in\{\pm1\}^T}[|f_{\mathsf{avg},T}(y)|] = \mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\ \mathop{\mathbf{E}}_{x\in\{\pm1\}^{T\setminus B}}\left|(f_{\mathsf{avg},T})_{B\to z}(x)\right| \tag{6}$$

$$= \mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\ \mathop{\mathbf{E}}_{x\in\{\pm1\}^{T\setminus B}}\left|\sum_{S\subseteq T\setminus B}\widehat{f_{B\to z}}(S)\chi_S(x)\right| \tag{7}$$

Furthermore, using the assumption in Eq. (5) it is an easy calculation to show that (7) equals

$$\mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\ \mathop{\mathbf{E}}_{x\in\{\pm1\}^{T\setminus B}}\left|\sum_{S\subseteq T\setminus B,|S|\le\kappa}\widehat{f_{B\to z}}(S)\chi_S(x)\right| \pm \varepsilon/2.$$

Similarly, for any set $U \subseteq \mathcal{S}$ of size k containing B (think of U as a candidate for T) we have that the best correlation between a junta-on-U and f is

$$\mathsf{corr}(f,\mathcal{J}_U) = \mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\ \mathop{\mathbf{E}}_{x\in\{\pm1\}^{U\setminus B}}\left|\sum_{S\subseteq U\setminus B}\widehat{f_{B\to z}}(S)\chi_S(x)\right|. \tag{8}$$

Now, however, the right hand side in Eq. (8) is not necessarily approximated by the low-degree counterpart as above for T. Indeed, we would like to estimate Eq. (8) for all candidates $U \subseteq \mathcal{S}$ of size k containing B, and pick the set with best estimated correlation. Based on our assumption on T, we can replace $\sum_{S\subseteq U\setminus B}\widehat{f_{B\to z}}(S)\chi_S(x)$ with its low-degree part $\sum_{S\subseteq U\setminus B,|S|\le\kappa}\widehat{f_{B\to z}}(S)\chi_S(x)$ for $U = T$, but its not clear whether we can do it in general.

In particular, if U satisfies

$$\mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\Bigg[\sum_{\substack{S\subseteq U\setminus B,\\ |S|>\kappa}} \widehat{f_{B\to z}}(S)^2\Bigg] > \varepsilon^2/4, \tag{9}$$

then taking the low-degree part can give an overestimate to the correlation with the best junta on U.[5] We settle for an estimate that is ε-accurate for the target set T assuming it satisfies Equation (5), and is not overestimating by more than ε for any other set $U \supseteq B$ of size k. Towards this goal, we first apply a noise operator that would essentially eliminate most of the contribution from sets larger than $\sqrt{k/\varepsilon}\log(1/\varepsilon)$ regardless of whether U satisfies Eq. (9) or not. This is captured by the following claim.

[5] To see a simple example of how this can happen, consider $f(x,y) = 1 - x - y + xy$. Then one can verify that $\mathbf{E}[|f(x,y)|] = 1 < 1.5 = \mathbf{E}[|1 - x - y|]$.

▷ Claim 45. Let $\rho = 1 - \sqrt{\varepsilon/k}$, $z \in \{\pm 1\}^B$ and denote by $h = f_{B\to z}$ and $h^{\mathsf{low}} = h^{\leq (\sqrt{k/\varepsilon})\cdot \log(1/\varepsilon)}$ (i.e., h^{low} is the truncated Fourier expansion of h that zeroes out all Fourier coefficients above level $(\sqrt{k/\varepsilon})\cdot\log(1/\varepsilon))$. For any $U : B \subseteq U \subseteq \mathcal{S}$ it holds that

$$\left|\mathsf{corr}\left(T_\rho h, \mathcal{J}_U\right) - \mathsf{corr}\left(T_\rho h^{\mathsf{low}}, \mathcal{J}_U\right)\right| \leq \varepsilon.$$

Proof. We have

$$\begin{aligned}
&\left|\mathsf{corr}\left(T_\rho h, \mathcal{J}_U\right) - \mathsf{corr}\left(T_\rho h^{\mathsf{low}}, \mathcal{J}_U\right)\right| \\
&= \left| \mathop{\mathbf{E}}_{x\in\{\pm1\}^{U\setminus B}} \left| \sum_{S\subseteq U\setminus B} \widehat{h}(S)\chi_S(x)\rho^{S}\right| - \mathop{\mathbf{E}}_{x\in\{\pm1\}^{U\setminus B}} \left| \sum_{\substack{S\subseteq U\setminus B,\\ |S|\leq(\sqrt{k/\varepsilon})\cdot\log(1/\varepsilon)}} \widehat{h}(S)\chi_S(x)\rho^{S}\right|\right| \\
&\leq \mathop{\mathbf{E}}_{x\in\{\pm1\}^{U\setminus B}} \left| \sum_{\substack{S\subseteq U\setminus B,\\ |S|>(\sqrt{k/\varepsilon})\cdot\log(1/\varepsilon)}} \widehat{h}(S)\chi_S(x)\rho^{|S|}\right| \\
&\leq \sqrt{\mathop{\mathbf{E}}_{x\in\{\pm1\}^{U\setminus B}} \left(\sum_{\substack{S\subseteq U\setminus B,\\ |S|>(\sqrt{k/\varepsilon})\cdot\log(1/\varepsilon)}} \widehat{h}(S)\chi_S(x)\rho^{|S|}\right)^2} \\
&= \sqrt{\sum_{\substack{S\subseteq U\setminus B,\\ |S|>(\sqrt{k/\varepsilon})\cdot\log(1/\varepsilon)}} \widehat{h}(S)^2\rho^{2|S|}} \leq \sqrt{\rho^{2(\sqrt{k/\varepsilon})\cdot\log(1/\varepsilon)}} \leq \varepsilon.
\end{aligned}$$

◀

Next, we show that applying a noise operator to f does not affect its correlation with a set U of size k, under the condition that most of the Fourier mass of $f_{B\to z}$ falls on the lower levels, i.e., $\mathbf{E}_z\left[\sum_{S\subseteq U\setminus B, |S|\geq\sqrt{k}} \widehat{f_{B\to z}}(S)^2\right] \leq \varepsilon^2/4$. Recall that this is what was guaranteed with high probability from the output of Algorithm 5 for our target set T.

▷ Claim 46. Let $\rho = 1 - \sqrt{k/\varepsilon}$. Given $U : B \subseteq U \subseteq \mathcal{S}$ such that $\mathbf{E}_z\left[\sum_{\substack{S\subseteq U\setminus B,\\ |S|\geq\kappa}} \widehat{f_{B\to z}}(S)^2\right] \leq \varepsilon^2/4$, we have that

$$\left|\mathop{\mathbf{E}}_z \mathsf{corr}(T_\rho(f_{B\to z}), \mathcal{J}_U) - \mathop{\mathbf{E}}_z \mathsf{corr}(f_{B\to z}, \mathcal{J}_U)\right| \leq 1.2\varepsilon.$$

Proof. Similar to the proof of Claim 45, we have

$$\begin{aligned}
&\left| \mathop{\mathbf{E}}_{\substack{z\in\{\pm1\}^B\\ x\in\{\pm1\}^{U\setminus B}}} \left|\sum_{S\subseteq U\setminus B} \widehat{f_{B\to z}}(S)\chi_S(x)\right| - \mathop{\mathbf{E}}_{\substack{z\in\{\pm1\}^B\\ x\in\{\pm1\}^{U\setminus B}}} \left|\sum_{S\subseteq U\setminus B} \widehat{f_{B\to z}}(S)\chi_S(x)\cdot\rho^{|S|}\right|\right| \\
&\leq \mathop{\mathbf{E}}_{\substack{z\in\{\pm1\}^B\\ x\in\{\pm1\}^{U\setminus B}}} \left|\sum_{S\subseteq U\setminus B} \widehat{f_{B\to z}}(S)\chi_S(x)(1-\rho^{|S|})\right|
\end{aligned}$$

$$\le \sqrt{\mathop{\mathbf{E}}_{\substack{z\in\{\pm1\}^B\\ x\in\{\pm1\}^{U\setminus B}}}\left(\sum_{S\subseteq U\setminus B}\widehat{f_{B\to z}}(S)\chi_S(x)(1-\rho^{|S|})\right)^2}$$

$$= \sqrt{\mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\left[\sum_{S\subseteq U\setminus B}\widehat{f_{B\to z}}(S)^2\cdot(1-\rho^{|S|})^2\right]}$$

$$\le \sqrt{\mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\left[\sum_{S\subseteq U\setminus B:|S|\le\kappa}\widehat{f_{B\to z}}(S)^2\cdot(1-\rho^{|S|})^2+\sum_{S\subseteq U\setminus B:|S|>\kappa}\widehat{f_{B\to z}}(S)^2\cdot(1-\rho^{|S|})^2\right]}$$

$$\le \sqrt{(1-\rho^{\kappa})^2+\varepsilon^2/4}\le\sqrt{\varepsilon^2+\varepsilon^2/4}\le 1.2\cdot\varepsilon. \quad ◀$$

The next lemma gives an algorithm that on any B, satisfying Equation (5), outputs $U : B\subseteq U\subseteq\mathcal{S}$ with $\mathsf{corr}(f,\mathcal{J}_U)\ge\mathsf{corr}(f,\mathcal{J}_T)-O(\varepsilon)$, with high probability.

▶ **Lemma 47** (Algorithm and Analysis for Phase-Two)**.** *Let $\varepsilon,\delta>0$. There's an algorithm that with probability at least $1-\delta$, gives ε-accurate estimates $\widetilde{c_U}$ to*

$$c_U=\mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\mathop{\mathbf{E}}_{x\in\{\pm1\}^{T\setminus B}}\left|\sum_{S\subseteq U\setminus B:|S|\le\sqrt{k/\varepsilon}\cdot\log(1/\varepsilon)}\widehat{f_{B\to z}}(S)\chi_S(x)\rho^{|S|}\right|$$

for all $U : B\subseteq U\subseteq\mathcal{S}$ of size k simultaneously. We return $(U,\widetilde{c_U})$ for the set U with maximal $\widetilde{c_U}$.

Complexity *The procedure uses $\log(1/\delta)2^{\widetilde{O}(\sqrt{k/\varepsilon})}$ queries and runs in time $\log(1/\delta)2^{k\cdot\widetilde{O}(1/\varepsilon)}$.*

Correctness *In the case where all estimates are ε-accurate, the following holds. If $B\subseteq T$ satisfies Equation (5), the above procedure would return $(U,\widetilde{c_U})$ with $\widetilde{c_U}\ge\mathsf{corr}(f,\mathcal{J}_T)-3.2\varepsilon$. Moreover, regardless of whether T and B satisfy Equation (5), we have $\widetilde{c_U}\le\mathsf{corr}(f,\mathcal{J}_U)+2\varepsilon$.*

Proof. First we show that we can estimate all c_U up to error ε simultaneously with high probability using the aforementioned query complexity and running time. We sample $t=O(\log(1/\delta)/\varepsilon^2)$ different $z\in\{\pm1\}^B$, and estimate for each value of z the Fourier coefficients of $\widehat{f_{B\to z}}(S)$ of all sets $S\subseteq\mathcal{S}$ of size at most $\zeta=\sqrt{k/\varepsilon}\cdot\log(\frac{2}{\varepsilon})$ up to additive error $\varepsilon/\binom{k}{\le\zeta}=2^{-\widetilde{\Omega}(\sqrt{k/\varepsilon})}$ with probability $1-\frac{\delta}{t\cdot\binom{k}{\le\zeta}}$, which is possible via Fact 6 with $\log(1/\delta)2^{\widetilde{O}(\sqrt{k/\varepsilon})}$ queries. Fact 6 guarantees that with probability $1-\delta$ for all sampled z, all estimated low-degree Fourier coefficients are within the additive error bound, in which case we have estimates for all c_U up to error ε simultaneously with probability $1-\delta$.

Next, we show the correctness of the procedure. On the one hand, in the assumed case, i.e., that T satisfies $\mathbf{E}_z\left[\sum_{S\subseteq T\setminus B,|S|\ge\kappa}\widehat{f_{B\to z}}(S)^2\right]\le\frac{\varepsilon^2}{4}$, we will have by Claim 45 and Claim 46 that

$$c_T\ge\mathsf{corr}(f,\mathcal{J}_T)-2.2\varepsilon \tag{10}$$

Since we output the set U with maximal $\widetilde{c_U}$, and since all estimates are correct up to ε we know that we output U with

$$\widetilde{c_U}\ge\widetilde{c_T}\ge c_T-\varepsilon. \tag{11}$$

Combining Equations (10) and (11) together we get

$$\widetilde{c_U} \geq c_T - \varepsilon \geq \mathsf{corr}(f, \mathcal{J}_T) - 3.2\varepsilon.$$

We move to prove the furthermore part, i.e., that $\widetilde{c_U} \leq \mathsf{corr}(f, \mathcal{J}_U) + 2\varepsilon$ regardless of whether T and B satisfy Equation (5). We start by showing that for any set U (whatsoever) we have that $\mathsf{corr}(f, \mathcal{J}_U) \geq c_U - \varepsilon$. Indeed, by Claim 45 we have

$$c_U \approx_\varepsilon \mathop{\mathbf{E}}_{\substack{z \in \{\pm 1\}^B \\ x \in \{\pm 1\}^{U \setminus B}}} \left| \sum_{S \subseteq U \setminus B} \widehat{f_{B \to z}}(S) \chi_S(x) \rho^{|S|} \right|$$

and since the noise operator can only reduce ℓ_1-norm (see Fact 13), we see that for all $z \in \{\pm 1\}^B$ it holds that

$$\mathop{\mathbf{E}}_{x \in \{\pm 1\}^{U \setminus B}} \left| \sum_{S \subseteq U \setminus B} \widehat{f_{B \to z}}(S) \chi_S(x) \rho^{|S|} \right| \leq \mathop{\mathbf{E}}_{x \in \{\pm 1\}^{U \setminus B}} \left| \sum_{S \subseteq U \setminus B} \widehat{f_{B \to z}}(S) \chi_S(x) \right|$$

Thus,

$$\begin{aligned} c_U &\leq \varepsilon + \mathop{\mathbf{E}}_{\substack{z \in \{\pm 1\}^B \\ x \in \{\pm 1\}^{U \setminus B}}} \left| \sum_{S \subseteq U \setminus B} \widehat{f_{B \to z}}(S) \chi_S(x) \rho^{|S|} \right| \\ &\leq \varepsilon + \mathop{\mathbf{E}}_{\substack{z \in \{\pm 1\}^B \\ x \in \{\pm 1\}^{U \setminus B}}} \left| \sum_{S \subseteq U \setminus B} \widehat{f_{B \to z}}(S) \chi_S(x) \right| = \varepsilon + \mathsf{corr}(f, \mathcal{J}_U) \end{aligned}$$

Since $|c_U - \widetilde{c_U}| \leq \varepsilon$, we get that $\widetilde{c_U} \leq c_U + \varepsilon \leq \mathsf{corr}(f, \mathcal{J}_U) + 2\varepsilon$. ◀

After phase one, we can apply Lemma 47 to each B from phase one, and get a set $U_B : B \subseteq U_B \subseteq \mathcal{S}$ of size k, along with an estimate of the correlation of f to $\mathcal{J}_{U_B}$. This leads to the proof of Theorem 2 which we restate next.

▶ **Theorem 48.** *Given a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$, it is possible to estimate the distance of f from the class of k-juntas to within additive error ε with probability $2/3$ using $2^{\widetilde{O}(\sqrt{k/\varepsilon})}$ adaptive queries to f. In particular, when ε is constant, this yields a $2^{\widetilde{O}(\sqrt{k})}$-query algorithm. However, the algorithm still requires $\exp(k/\varepsilon)$ time.*

Proof. Let $\varepsilon_0 = \varepsilon/6$

1. We first apply the result of [13] to reduce the down to only $\mathsf{poly}(k, 1/\varepsilon_0)$ coordinates. This incurs a loss in correlation of at most ε_0, and fails with probability at most δ_1, which we can set to be $1/20$, by Corollary 26.
2. Next, we apply our Theorem 4, which reduces the number of oracles we have to consider down to $O(k/\varepsilon_0^2)$, incurs an additive loss in correlation of at most ε_0, and fails with probability at most $\delta_2 = 1/20$.
3. Then, we run phase 1 of our algorithm, which fails with probability at most $\delta_3 = 1/20$ by Lemma 42.
4. Finally, we apply Lemma 47 to every B outputted by Algorithm 5 to get a set U_B and an estimate $\widetilde{C_{U_B}}$ for the correlation of f with $\mathcal{J}_{U_B}$ We iterate on all sets B returned by phase-1 and return U_B with the highest estimate of correlation.
   There are $\ell = O(\frac{1}{\varepsilon_0^2})^{3\sqrt{k/\varepsilon_0} + \log(2/\delta_3)} = 2^{\widetilde{O}(\sqrt{k/\varepsilon_0})}$ branches, and thus if we apply the algorithm from Lemma 47 with $\delta = 1/(20\ell)$, we get that all this step fail with probability at most $1/20$ by a union bound.

By a union bound, each of these steps succeeds with probability at least $1-4/20 \geq 2/3$. In the case all steps succeeds, we return a set U with $\widetilde{c_U} \geq \mathsf{corr}(f, \mathcal{J}_{n,k}) - 5.2\varepsilon_0$. In addition, the moreover part in Lemma 47 guarantees that $\widetilde{c_U} \leq \mathsf{corr}(f, \mathcal{J}_U) + 2\varepsilon_0 \leq \mathsf{corr}(f, \mathcal{J}_{n,k}) + 2\varepsilon_0$. We get that the returned value is within $5.2\varepsilon_0 < \varepsilon$ of $\mathsf{corr}(f, \mathcal{J}_{n,k})$. Finally, since $\mathsf{dist}(f, \mathcal{J}_{n,k}) = \frac{1+\mathsf{corr}(f,\mathcal{J}_{n,k})}{2}$ we get that $\frac{1+\widetilde{c_U}}{2}$ is an $\varepsilon/2$-accurate approximation of $\mathsf{dist}(f, \mathcal{J}_{n,k})$. Finally, we note that the query complexities of phase 1 and phase 2 are both $2^{\widetilde{O}(\sqrt{k/\varepsilon})}$, but the runtime is exponential due to Lemma 47. ◀

Finally, we mention that if our goal is not to estimate to correlation with the nearest k-junta to f, but rather to simply estimate the most amount of Fourier mass any subset of k variables contains, then we have the following theorem with an improved dependence on ε:

▶ **Theorem 49.** *Given a Boolean function $f : \{\pm 1\}^n \to \{\pm 1\}$, it is possible to estimate the most mass any subset of at most k variables of f has to within additive error ε with probability $2/3$ using $2^{\widetilde{O}(\sqrt{k}\log(1/\varepsilon))}$ adaptive queries to f. In particular, when ε is constant, this yields a $2^{\widetilde{O}(\sqrt{k})}$-query algorithm. However, the algorithm still requires* $\exp(k\log(1/\varepsilon))$ *time.*

We leave the proof of this theorem, which involves simple modifications to the algorithm presented in this section, to Appendix A.

5.3 Proof of Theorem 40

We now present the proof of Theorem 40.

Proof of Theorem 40. The proof is very similar to the previous proof of Theorem 34, so we explain how to modify it to this case.

We express λ_U in terms of the Fourier spectrum of f.

$$\begin{aligned}
\lambda_U &= \sum_{m=0}^{2|U|\log(10k)} \sum_{S:S\supseteq U} \widehat{f}(S)^2 \cdot \Pr_{J\subseteq_{p^m}[\ell]}[S\cap J = U] \\
&= \sum_{m=0}^{2|U|\log(10k)} \sum_{S:S\supseteq U} \widehat{f}(S)^2 \cdot \Pr_{J\subseteq_{p^m}[\ell]}[|S\cap J| = |U|] \cdot \frac{1}{\binom{|S|}{|U|}} \\
&= \sum_{S:S\supseteq U} \frac{\widehat{f}(S)^2}{\binom{|S|}{|U|}} \cdot \sum_{m=0}^{2|U|\log(10k)} \Pr_{J\subseteq_{p^m}[\ell]}[|S\cap J| = |U|]
\end{aligned}$$

It suffices to show that for any non-empty set S of size at least $|U|$ and at most k it holds that

$$\sum_{m=0}^{2|U|\log(10k)} \Pr_{J\subseteq_{p^m}[\ell]}[|S\cap J| = |U|] \in [1/2, 3] \,. \tag{12}$$

Again, we can analyze the sum on the left hand side of Equation (12) as the expected final value of X in the following random process:

By symmetry the expected value depends only on the size of the initial set S. As before, we denote by F_t its expected value starting with a set S of size t with an infinite horizon, and $F_t^{(i)}$ as the expected value of X at the end of the above process with finite horizon i.

```
$X \leftarrow 0$
for $i = 1, 2, \ldots, 2|U|\log(10k)$ do
    if $|S| < |U|$ then
        halt!
    if $|S| = |U|$ then
        increase $X$
    Sample $J_i \subseteq_p [\ell]$
    $S \leftarrow S \cap J_i$
```

We start by analyzing $F_{|U|}$. In this case, X is a geometric random variable with stopping probability $1 - p^{|U|}$. Thus, its expectation is

$$F_{|U|} = 1/(1 - p^{|U|}) = 1/(1 - (1 - 1/2|U|)^{|U|}) \in [2, 3].$$

This implies that $F_{|U|}^{(2|U|\log(10k))} \le F_{|U|} \le 3$. For $t > |U|$ in the infinite horizon case we have the recurrence

$$F_t = \sum_{a=0}^{t} F_a \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) = a] = \sum_{a=|U|}^{t-1} F_a \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) = a] + F_t \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) = t] \quad (13)$$

or equivalently

$$F_t \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) < t] = \sum_{a=|U|}^{t-1} F_a \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) = a] \quad (14)$$

We prove by induction that for $t \ge |U|$ it holds that $F_t \le F_{|U|}$. The claim clearly holds for $t = |U|$. For $t > |U|$ we can apply induction and get

$$F_t \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) < t] \le \sum_{a=|U|}^{t-1} F_{|U|} \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) = a] \le F_{|U|} \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) < t],$$

and thus $F_t \le F_{|U|}$. This immediately implies that $F_t^{(2|U|\log(10k))} \le F_t \le 3$. On the other hand we prove that $F_t^{(2|U|\log(10k))} \ge 1/2$ as long as $t \le k$. To do so, we once again introduce the indicator random variable $Y_t^{(i)}$, where $t = |S|$, and which equals 1 if $|S| = |U|$ at some point during the above process before iteration i. We note that $Y_t^{(2|U|\log(10k))}$ is a lower bound for the value of X in the above process, and Y_t is a lower bound for the value of X at the end of the infinite horizon process. We note that the case $|U| = 1$ was already lower bounded in Section 4.5, where it was shown that $\mathbf{E}[Y_t^{(\log(10k))}] \ge 1/2$, and therefore $\mathbf{E}[Y_t^{(2|U|\log(10k))}] \ge 1/2$. It remains to show that the $\mathbf{E}[Y_t^{(2|U|\log(10k))}] \ge 1/2$ is true for any set $|U| \ge 2$.

First, we show that $\mathbf{Pr}[\mathrm{Bin}(t,p) < |U|] \le \frac{1}{2}\mathbf{Pr}[\mathrm{Bin}(t,p) = |U|]$. Towards this goal, it would suffice to prove that $3 \le \mathbf{Pr}[\mathrm{Bin}(t,p) = i+1]/\mathbf{Pr}[\mathrm{Bin}(t,p) = i]$ for $i < |U|$ and $t \ge |U| + 1$. This would suffice since in this case

$$\sum_{i=0}^{|U|-1} \mathbf{Pr}[\mathrm{Bin}(t,p) = i] \le \sum_{i=0}^{|U|-1} \frac{3^i}{3^{|U|}} \mathbf{Pr}[\mathrm{Bin}(t,p) = |U|] \le \frac{1}{2} \cdot \mathbf{Pr}[\mathrm{Bin}(t,p) = |U|].$$

Indeed, The ratio between the two aforementioned probabilities is

$$\begin{aligned}\frac{\mathbf{Pr}[\mathrm{Bin}(t,p)=i+1]}{\mathbf{Pr}[\mathrm{Bin}(t,p)=i]} &= \frac{\binom{t}{i+1}}{\binom{t}{i}}\cdot\frac{p^{i+1}(1-p)^{t-(i+1)}}{p^i(1-p)^{t-i}}\\ &= \frac{t-i}{i+1}\cdot\frac{p}{1-p}\geq\frac{2}{|U|}\cdot\frac{1-1/2|U|}{1/2|U|}=\frac{2-1/|U|}{1/2}\geq 3\end{aligned}$$

as needed. Now, we claim that $\mathbf{E}[Y_t]=\mathbf{Pr}[Y_t=1]\geq 2/3$ for all $t\geq 1$. The base case of $t=1$ is certainly true. Assuming we have $\mathbf{Pr}[\mathrm{Bin}(t,p)<|U|]\leq\frac{1}{2}\mathbf{Pr}[\mathrm{Bin}(t,p)=|U|]$ we have

$$\begin{aligned}\mathbf{E}[Y_t]\cdot\mathbf{Pr}[Bin(t,p)<t] &= \sum_{a=|U|}^{t-1}\mathbf{E}[Y_a]\cdot Pr[\mathrm{Bin}(t,p)=a]\\ &\geq \mathbf{Pr}[\mathrm{Bin}(t,p)=|U|]+\sum_{a=|U|+1}^{t-1}\mathbf{Pr}[\mathrm{Bin}(t,p)=a]\,\mathbf{E}[Y_a]\\ &\geq \mathbf{Pr}[\mathrm{Bin}(t,p)=|U|]+\frac{2}{3}\mathbf{Pr}[\mathrm{Bin}(t,p)\in[|U|+1,t-1]]\\ &= \frac{2}{3}\mathbf{Pr}[\mathrm{Bin}(t,p)<t]-\frac{2}{3}\mathbf{Pr}[\mathrm{Bin}(t,p)<|U|]+\frac{1}{3}\mathbf{Pr}[\mathrm{Bin}(t,p)=|U|]\\ &\geq \frac{2}{3}\mathbf{Pr}[\mathrm{Bin}(t,p)<t]\end{aligned}$$

which implies that $\mathbf{E}[Y_t]\geq 2/3$. Finally, let A be the event that $S=\emptyset$ by iteration $2|U|\log(10k)$, and note that

$$\begin{aligned}\mathbf{Pr}[A] &= \mathbf{Pr}[\mathrm{Bin}(|S|,(1-\tfrac{1}{2|U|})^{2|U|\log(10k)})=0]\\ &\geq \mathbf{Pr}[\mathrm{Bin}(k,e^{-\log(10k)})=0]=\mathbf{Pr}[\mathrm{Bin}(k,\tfrac{1}{10k})=0]\geq 0.9\end{aligned}$$

as was shown in the proof for Theorem 34 in Section 4.5. Finally, we claim that for all $t\geq 2$ we have that $\mathbf{Pr}[Y_t^{(2|U|\log 10k)}]\geq 1/2$. Indeed, we have that

$$\mathbf{Pr}[Y_t^{(2|U|\log 10k))}=1]\geq\mathbf{Pr}[Y_t=1]-\mathbf{Pr}[\overline{A}]\geq\tfrac{2}{3}-0.1\geq\tfrac{1}{2}.$$

as desired, provided $|S|\leq k$. ◀

6 Conclusions and Open Problems

We conclude by mentioning some future research directions. First, we believe some of the techniques discussed in this paper could lead to other interesting work in property testing, learning theory, or Boolean function analysis in general. In particular, the procedure in Algorithm 1 makes use of a random process to get access to an underlying junta, a subprocedure that could be useful in other learning or testing algorithms. In addition, we are able to approximate the quantities $\mathbf{NInf}_i$ and $\mathbf{NInf}_U$, that serve as key steps in our algorithms. These quantities seems natural on their own, and would likely find further applications in Analysis of Boolean functions. In particular, they seem to capture more accurately the intuition that "influences measures the importance of coordinates". While the total influence of a Boolean function can be any number between $\mathbf{Var}[f]$ and $n\cdot\mathbf{Var}[f]$ the total normalized influence equals exactly $\mathbf{Var}[f]$, and thus normalized influences can be seen as a distribution of the variance among the coordinates.

Interestingly, our algorithms strongly resemble certain quantum algorithms. In particular, the sampling of coordinates is done through the Fourier distribution, a process which can be done much more efficiently with a quantum algorithm (querying f in superposition, applying the Hadamard transform, and measuring). This idea was leveraged in [2] to provide fast quantum algorithms for testing juntas in the standard property testing regime. Indeed, if the nearest k-junta to f has its mass on higher levels (say above $\sqrt{k}$ or even $k/2$), then Fourier sampling is extremely effective and provides a cleaner way of sampling subsets according to the Fourier distribution than the related classical technique we provided in Section 5. However, the issue arises when the nearest k-junta has Fourier mass on lower levels (below $\log k$ or even a constant, for example). In this case, it is not clear to us how quantum algorithms provide any advantage over classical ones. An open question is whether quantum Fourier sampling techniques can be applied in a more clever way to give faster algorithms in the tolerant testing paradigm.

Finally, a clear open question is how good of a lower bound one can prove on the query complexity of the tolerant junta testing problem. Our main result Theorem 2, rules out strictly exponential-in-k query lower bounds for k-junta distance approximation. [27] proved a non-adaptive query complexity lower bound of 2^{k^η} for $(k, k, \varepsilon_1, \varepsilon_2)$-tolerant junta testing (given a particular choice of $0 < \varepsilon_1 < \varepsilon_2 < 1/2$), for any $0 < \eta < 1/2$. While this is quite close to our upper bound of $2^{\widetilde{O}(\sqrt{k})}$, our algorithm is highly adaptive, while the lower bound due to [27] applies only to nonadaptive algorithms. Therefore, another interesting direction would be to explore whether any nontrivial lower bounds apply to adaptive algorithms for tolerant (junta) testing and distance approximation.

References

1 Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Estimating the distance to a monotone function. *Random Structures & Algorithms*, 31(3):371–383, 2007. doi:10.1002/rsa.20167.

2 Andris Ambainis, Aleksandrs Belovs, Oded Regev, and Ronald de Wolf. Efficient quantum algorithms for (gapped) group testing and junta testing. In *SODA*, pages 903–922. SIAM, 2016.

3 Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, pcps, and nonapproximability-towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.

4 Eric Blais. Improved bounds for testing juntas. In *APPROX-RANDOM*, volume 5171 of *Lecture Notes in Computer Science*, pages 317–330. Springer, 2008.

5 Eric Blais. Testing juntas nearly optimally. In *STOC*, pages 151–158. ACM, 2009.

6 Eric Blais, Clément L. Canonne, Talya Eden, Amit Levi, and Dana Ron. Tolerant junta testing and the connection to submodular optimization and function isomorphism. *ACM Trans. Comput. Theory*, 11(4):24:1–24:33, 2019.

7 Guy Blanc, Jane Lange, and Li-Yang Tan. Testing and reconstruction via decision trees. *CoRR*, abs/2012.08735, 2020.

8 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *STOC*, pages 73–83. ACM, 1990.

9 Nader H. Bshouty. Almost optimal distribution-free junta testing. In *Computational Complexity Conference*, volume 137 of *LIPIcs*, pages 2:1–2:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

10 Sourav Chakraborty, Eldar Fischer, David García-Soriano, and Arie Matsliah. Junto-symmetric functions, hypergraph isomorphism and crunching. In *Computational Complexity Conference*, pages 148–158. IEEE Computer Society, 2012.

11 Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. *J. ACM*, 65(6):40:1–40:18, 2018.

12 Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Inf. Process. Lett.*, 90(6):301–305, 2004.

13 Anindya De, Elchanan Mossel, and Joe Neeman. Junta correlation is testable. In *FOCS*, pages 1549–1563. IEEE Computer Society, 2019.

14 Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *FOCS*, pages 549–558. IEEE Computer Society, 2007.

15 Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Rocco A. Servedio, and Andrew Wan. Efficiently testing sparse GF(2) polynomials. In *ICALP (1)*, volume 5125 of *Lecture Notes in Computer Science*, pages 502–514. Springer, 2008.

16 Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci.*, 68(4):753–787, 2004.

17 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 339–348. IEEE Computer Society, 1996. `doi:10.1109/SFCS.1996.548493`.

18 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

19 Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions (extended abstract). In *FOCS*, pages 68–80. IEEE Computer Society, 1988.

20 Michael J. Kearns and Dana Ron. Testing problems with sublearning sample complexity. *J. Comput. Syst. Sci.*, 61(3):428–456, 2000.

21 Esty Kelman, Subhash Khot, Guy Kindler, Dor Minzer, and Muli Safra. Theorems of kkl, friedgut, and talagrand via random restrictions and log-sobolev inequality. *Electron. Colloquium Comput. Complex.*, 27:9, 2020.

22 Amit Levi and Erik Waingarten. Lower bounds for tolerant junta and unateness testing via rejection sampling of graphs. In *ITCS*, volume 124 of *LIPIcs*, pages 52:1–52:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

23 Zhengyang Liu, Xi Chen, Rocco A. Servedio, Ying Sheng, and Jinyu Xie. Distribution-free junta testing. *ACM Trans. Algorithms*, 15(1):1:1–1:23, 2019.

24 Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning juntas. In *STOC*, pages 206–212. ACM, 2003.

25 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL: `http://www.cambridge.org/de/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/analysis-boolean-functions`.

26 Ryan O'Donnell and Karl Wimmer. Sharpness of KKL on Schreier graphs. *Electronic Communications in Probability*, 18:1–12, 2013. `doi:10.1214/ECP.v18-1961`.

27 Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. Approximating the distance to monotonicity of boolean functions. In *SODA*, pages 1995–2009. SIAM, 2020.

28 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Electron. Colloquium Comput. Complex.*, 010, 2004.

29 Michal Parnas, Dana Ron, and Alex Samorodnitsky. Proclaiming dictators and juntas or testing boolean formulae. In *RANDOM-APPROX*, volume 2129 of *Lecture Notes in Computer Science*, pages 273–284. Springer, 2001.

30 Mert Saglam. Near log-convexity of measured heat in (discrete) time and consequences. In *FOCS*, pages 967–978. IEEE Computer Society, 2018.

31 Rocco A. Servedio. Testing by implicit learning: A brief survey. In *Property Testing*, volume 6390 of *Lecture Notes in Computer Science*, pages 197–210. Springer, 2010.

32 Michel Talagrand. On Russo's Approximate Zero-One Law. *The Annals of Probability*, 22(3):1576–1587, 1994. `doi:10.1214/aop/1176988612`.

33 Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *FOCS*, pages 11–20. IEEE Computer Society, 2012.

34 Xiaojin Zhang. Near-optimal algorithm for distribution-free junta testing. *CoRR*, abs/1911.10833, 2019.

A Maximum k-Subset Fourier Mass Approximation

In this section, we sketch a proof of Theorem 49, which involves simple modifications and observations about our algorithm. The main difference is that we sample from the normalized influence subdistribution at a different Fourier level – namely, we let $\kappa := \sqrt{k}$ and $\alpha = k/\kappa = \sqrt{k}$ in Algorithm 4 and Algorithm 5, respectively (recall that before, $\kappa = \sqrt{\varepsilon k}$). This improves the query complexity dependence on ε in Phase 1.

▷ Claim 50. The query complexity of phase one of the algorithm for constant δ (failure probability) is $2^{\widetilde{O}(\sqrt{k}\log(1/\varepsilon))}$.

Proof. The proof is analogous to the proof of Claim 44, so we just point out the differences. We still require our Fourier coefficients to be accurate to within $1/\mathsf{poly}(k, 1/\varepsilon)$, and we require confidence $1 - O(1/\ell) = 1 - 2^{\widetilde{\Omega}(\sqrt{k}\log(1/\varepsilon))}$. However, now our branching process now has depth only $O(\sqrt{k})$, so we need only repeat this $O(\ell) = 2^{\widetilde{O}(\sqrt{k}\log(1/\varepsilon))}$ times, which yields the improved query complexity. ◁

In Phase 2, we argue that it is not necessary to apply a noise operator in order to only consider Fourier mass below level κ after Phase 1. Recall that we applied this noise operator in Section 5.2 in order to deal with the case that a particular U satisfied

$$\mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\left[\sum_{\substack{S\subseteq U\setminus B,\\ |S|>\kappa}} \widehat{f_{B\to z}}(S)^2\right] > \varepsilon^2/4. \tag{15}$$

If this happened, then we could not rule out the possibility that taking the low-degree part of f within U gives an *overestimate* to the correlation with the best k-junta. However, now we are not concerned with the junta correlation, but rather which set has the most mass, so we claim we do not have to worry about this possibility anymore. To see this, suppose we have identified $B \subseteq U$, and note that

$$\begin{aligned}\sum_{S\subseteq U}\widehat{f}(S)^2 &= \mathop{\mathbf{E}}_{x}[f(x)f_{\mathsf{avg},U}(x)]\\ &= \mathop{\mathbf{E}}_{z\in\{\pm1\}^B}\left[\mathop{\mathbf{E}}_{x}[f_{B\to z}(x)(f_{\mathsf{avg},U})_{B\to z}(x)]\right]\\ &= \mathop{\mathbf{E}}_{z}\left[\sum_{\substack{S\subseteq U\\ |S|\le\kappa}}\widehat{f_{B\to z}}(S)^2 + \sum_{\substack{S\subseteq U\\ |S|>\kappa}}\widehat{f_{B\to z}}(S)^2\right]\\ &\ge \mathop{\mathbf{E}}_{z}\left[\sum_{\substack{S\subseteq U\\ |S|\le\kappa}}\widehat{f_{B\to z}}(S)^2\right].\end{aligned}$$

Therefore, we no longer have to apply any noise operator, which negates the necessity of Claim 45 and Claim 46. It therefore suffices in Lemma 47 to estimate the mass of each set, rather than the correlation, as

$$m_U = \mathop{\mathbf{E}}_{z}\left[\sum_{\substack{S\subseteq U\\ |S|\le\kappa}}\widehat{f_{B\to z}}(S)^2\right].$$

To do so, as in the proof of Lemma 47 we let $t = O(\log(1/\delta)/\varepsilon^2)$ be the number of random samples of z we take. Then we estimate all the Fourier coefficients below level κ. This requires estimating $\widehat{f}(S)$ for all $S \subseteq \mathcal{S}$ of size at most κ up to additive error $\varepsilon/\binom{k}{\le\kappa} = 2^{\widetilde{\Omega}(\sqrt{k}\log(1/\varepsilon))}$ with probability $1 - \frac{\delta}{t\cdot\binom{k}{\le\kappa}}$, which is possible via Fact 6 with $\log(1/\delta)2^{\widetilde{O}(\sqrt{k}\log(1/\varepsilon))}$ queries. The rest of our argument and algorithm is exactly the same as in Section 5.

Arithmetic Circuit Complexity of Division and Truncation

Pranjal Dutta ✉
Chennai Mathematical Institute, India

Gorav Jindal ✉
Institut für Mathematik, Technische Universität Berlin, Germany

Anurag Pandey ✉
Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

Amit Sinhababu ✉
Aalen University, Germany

Abstract

Given polynomials $f, g, h \in \mathbb{F}[x_1, \ldots, x_n]$ such that $f = g/h$, where both g and h are computable by arithmetic circuits of size s, we show that f can be computed by a circuit of size $\mathrm{poly}(s, \deg(h))$. This solves a special case of division elimination for high-degree circuits (Kaltofen'87 & WACT'16). The result is an exponential improvement over Strassen's classic result (Strassen'73) when $\deg(h)$ is $\mathrm{poly}(s)$ and $\deg(f)$ is $\exp(s)$, since the latter gives an upper bound of $\mathrm{poly}(s, \deg(f))$.

Further, we show that any univariate polynomial family $(f_d)_d$, defined by the initial segment of the power series expansion of rational function $g_d(x)/h_d(x)$ up to degree d (i.e. $f_d = g_d/h_d \bmod x^{d+1}$), where circuit size of g is s_d and degree of g_d is at most d, can be computed by a circuit of size $\mathrm{poly}(s_d, \deg(h_d), \log d)$. We also show a hardness result when the degrees of the rational functions are high (i.e. $\Omega(d)$), assuming hardness of the integer factorization problem.

Finally, we extend this conditional hardness to *simple* algebraic functions as well, and show that for every prime p, there is an integral algebraic power series with its minimal polynomial satisfying a degree p polynomial equation, such that its initial segment is hard to compute unless integer factoring is easy, or a multiple of $n!$ is easy to compute. Both, integer factoring and computation of multiple of $n!$, are believed to be notoriously hard. In contrast, we show examples of transcendental power series whose initial segments are easy to compute.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Computational complexity and cryptography

Keywords and phrases Arithmetic Circuits, Division, Truncation, Division elimination, Rational function, Algebraic power series, Transcendental power series, Integer factorization

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.25

Funding *Pranjal Dutta*: Supported by Google Ph. D. Fellowship.
Gorav Jindal: Supported by Graduiertenkolleg "Facets of Complexity/Facetten der Komplexität" (GRK 2434).
Amit Sinhababu: Supported by DFG grant TH 472/5-1.

Acknowledgements We thank Himanshu Shukla for several discussions on the complexity of truncated power series, and for bringing the reference [13] to our attention. P. D. would like to thank CSE, IIT Kanpur for the hospitality. A. S. would like to thank the Institute of Theoretical Computer Science at Ulm University for the hospitality. We thank Thomas Thierauf and Nitin Saxena for discussions and feedback on the draft.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 25; pp. 25:1–25:36
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

An arithmetic circuit over an underlying field $\mathbb{F}$ is a natural model that represents a polynomial compactly (for definition see Appendix A). Arithmetic circuit complexity is the study of complexity (in terms of circuit size) of computing polynomial families. In this paper, we study two important questions in arithmetic circuit complexity. The first question is about the power of division in arithmetic circuits. The second question is about arithmetic circuit complexity of univariate polynomial families, defined by initial segments of various power series.

Complexity of division. In a classic result [45], Strassen showed that a polynomial $f(x_1, \ldots, x_n)$ of degree d, computed by an arithmetic circuit of size s using division, can also be computed by a division-free arithmetic circuit (i.e. only using addition and multiplication gates) of size $\mathrm{poly}(s, d)$.

Note that, arithmetic circuits can compute polynomials that have *exponential* degree wrt its size. For example, $g(x) := x^{2^s} - 1$, has $O(s)$-size circuit. Now, if we divide it by $h(x) := x - 1$, we get the polynomial $f(x) := 1 + x + \cdots + x^{2^s - 1}$. Strassen [45] gives an $\exp(s)$-size upper bound on the complexity of $f(x)$, whereas it is easy to see that $f(x)$ can be computed by just a $\mathrm{poly}(s)$-size circuit (see Remark 15). This leads to the following natural question.

▶ **Problem 1** ([23, Problem 5]). If a polynomial can be computed by an arithmetic circuit (with division) of size s, can it be computed by a division-free arithmetic circuit of size $\mathrm{poly}(s)$?

This question is still *open* [49] and it is unclear whether we should *expect* a positive answer. One can push the division gate at the top and show that if f has a s-size circuit (with division gates) then there exist polynomials g and h such that $f = g/h$, where both g and h have $\mathrm{poly}(s)$-size circuits. However, $\deg(f), \deg(g)$ and $\deg(h)$ can be $\exp(s)$, and it is not clear how to eliminate this division gate at the top without incurring *exponential* blowup. In fact, the division elimination method, due to Strassen [45], leads to an exponential blowup in size (see Section 1.3).

Even a special case of eliminating division is open, when $f = g/x^{2^s}$, and $\deg(g)$ and $\deg(f)$ are $\exp(s)$, but g has a s-size circuit. Solving this case would resolve a couple of interesting questions in algebraic complexity. We briefly discuss some of these implications in Section 4.

Complexity of truncated power series. The second part of the paper studies the complexity of families of univariate polynomials, defined by the initial segments (equivalently, *truncation*) of a power series. Power series are ubiquitous in all branches of mathematics. From the perspective of computer science, they are quite crucial because of their pervasiveness in enumeration and combinatorics. Efficient methods to compute truncations of power series allows us to compute number sequences emerging in enumerative combinatorics like Fibonacci numbers, Catalan numbers, and Bell numbers; thanks to the *generating functions* (see [39] for a survey). It also facilitates approximations of several irrational and transcendental numbers of interest, for example: $e, \pi, \sqrt{2}$, and $\zeta(3)$. The relation between truncations of power series and the theory of formal languages and context-free grammars, and the theory of codes is also well studied (see, for instance, [33, 4]). In complexity theory, computing truncations of power series has been crucial in results on polynomial factorization [17], division elimination in circuits [45], complexity of symmetric polynomials [5], and complexity of algebraic functions [29].

Easy and hard univariate families. A univariate polynomial family $(f_d)_d$, where f_d has degree d, is called *easy to compute*, if there is a poly$(\log d)$-size circuit computing f_d, otherwise we call it a *hard* family. Some examples of easy families are, $f_d := x^d$, $f_d := \sum_{i\in[d]} i^r\, x^i$, where $r \in \mathbb{N}$ (see [52]). A candidate *hard* family is the Pochhammer-Wilkinson polynomial $f_d := \prod_{i\in[d]} (x+i)$, for if it turns out to be easy, it would imply that integer factorization is also easy [32, 9].

One of the ultimate goals in algebraic complexity is to characterize "easy" and "hard" polynomial families (by showing explicit bounds). Can we give interesting examples of easy univariate polynomial families that can be defined via truncation of power series? Let us again look at the polynomial family $f_d := 1 + x + \cdots + x^d$; this has a $O(\log d)$-size circuit (Remark 15). Interestingly, it is also the initial segment of the power series expansion of $1/(1-x)$. In contrast, [31] showed that there exists a power series with $0-1$ coefficients such that their initial segments are *hard*. In fact, some of the famous candidate hard univariate polynomial families are those corresponding to initial segments of transcendental power series, for instance, $f_d := \sum_{i=0}^{d} x^i/i!$, and $f_d := \sum_{i\in[d]}(-1)^i x^i/i$, the truncations of e^x and $\log(1+x)$ respectively. Their hardness is known to imply that permanent requires superpolynomial size constant-free circuits, which implies the constant-free version of Valiant's hypothesis (the algebraic analog of $\mathsf{P} \neq \mathsf{NP}$ hypothesis) [9].

This motivates our second problem.

▶ **Problem 2.** Characterize (differentiate "easy" and "hard") polynomial families $(f_d(x))$, defined by the initial segment (upto degree d) of a power series $\sum_{i\geq 0} a_i\, x^i$.

Since the truncation of $1/(1-x)$ is easy to compute, as a natural first step towards the above Problem 2, we explore the complexity of initial segments of general rational functions $g(x)/h(x)$. Note that, rational function truncation is interesting, as any power series truncation up to some degree matches with a unique rational function (of given numerator and denominator degree) given by *Padé approximation* and this arises in many symbolic computational problems.

Subsequently, we study the complexity of initial segments of algebraic power series (eg. $\sqrt{1+x}$), and its connections to the central problems in algebraic complexity theory. Also, the examples of truncations of e^x and $\log(1+x)$ make us wonder whether all transcendental power series are likely to be hard. Towards this, we study truncations of transcendental power series as well.

▶ Remark 3. Very recently, [18] introduced the notion of *SOS-hardness* (in the sum-of-squares (SOS) representation). A family $(f_d)_d$ is *SOS-easy* if it can be written as $f_d = \sum_{i\in[s]} c_i\, g_i^2$, for $c_i \in \mathbb{F}$ such that $\sum_i |g_i|_0 = O(d^{1/2})$, where $|g_i|_0$ denotes the sparsity or the number of monomials in g_i. Otherwise, f_d is a *SOS-hard* family. The *minimal* SOS-representation captures its SOS-complexity. For formal definitions, refer to Section 8. [18] showed that the SOS-hard families are innately connected to proving $\mathsf{VP} \neq \mathsf{VNP}$ (for definitions, see Appendix A). Throughout the paper, we will talk about easy/hard families wrt. both the measures (circuit complexity and SOS-complexity[1]).

[1] Although there are polynomial families like $f_d := \sum_{i=0}^{d} x^i$, which are easy wrt. both the measures (see Lemma 67), in general, connection between these notions is unclear. Eg. $f_d := (x+1)^d$ is a candidate SOS-hard family, but has $O(\log d)$-size circuit. Conversely, a *random* $d^{1/2}$-sparse polynomial is trivially SOS-easy but *requires* $\omega(\log d)$-size circuit.

1.1 Our contributions

In this work, we make progress towards both Problems Problem 1 and Problem 2. Towards the division problem, we show the following Theorem 4. For more details, see Section 3 (Theorem 18 and Theorem 22).

▶ **Theorem 4** (Division by low-degree polynomial). *Suppose, f, g, h are polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ such that $f = g/h$. Then, f can be computed by an arithmetic circuit of size* $\operatorname{poly}(s_1, s_2, d_h)$*, where s_1 (respectively, s_2) is the circuit complexity of g (respectively h), and d_h is the degree of h.*

▶ Remark 5.

a. This result also holds when one replaces the circuit-size by *approximative* circuit-size; see Section 3.3 for details.

b. When $s_1, s_2 \leq s$, $\deg(h) = \operatorname{poly}(s)$, and $\deg(f) = \exp(s)$, our result is exponentially better than Strassen's division elimination [45] as the latter gives $\exp(s)$ upper bound.

Cofactor of a low-degree factor. If a multivariate polynomial $f = gh$, has size s, with $\gcd(g, h) = 1$, and $\deg(g) = \operatorname{poly}(s)$, then [23] showed that g has a $\operatorname{poly}(s)$-size circuit. Invoking Theorem 4, we can now conclude that the cofactor h has a $\operatorname{poly}(s)$-size circuit as well (Kaltofen claimed only a $\operatorname{poly}(s)$-size circuit with division, for computing h; see the last paragraph in [23, Section 4]). If g, h are not co-prime, then we need Factor Conjecture (see Section 4) to claim low complexity of h.

A related problem to division elimination is the truncation problem. Towards that, we initiate a systematic study by considering truncation of rational, algebraic and transcendental functions. For computing the initial segment of rational functions, we first generalize the observation that the initial segment of $1/(1-x)$ is easy to compute, via the *inverse identity*: $1/(1-x) = \sum_{i \geq 0} x^i$. It turns out that as long as the degree of the denominator is small, the degree-d truncation has low complexity (Theorem 6). We denote the ring of formal power series as $\mathbb{F}[[x]]$.

▶ **Theorem 6** (Truncation of low-degree rational function). *Suppose, g and h are two univariate polynomials in $\mathbb{F}[x]$ such that* $\deg(g) \leq d$, $\deg(h) = d_h$*, and g can be computed a circuit of size s. Let, $g/h \in \mathbb{F}[[x]]$. Then, truncation of g/h upto degree-d can be computed by a circuit of size* $\operatorname{poly}(s, d_h, \log d)$.

▶ Remark 7.

a. When g and h are both constant-degree polynomials, then the truncation, in fact, has a *small* SOS-complexity. For details, see Theorem 49.

b. We complement the above Theorem 6 upper bound by a conditional hardness result. In particular, we exhibit rational functions of high degree (e.g. $\Omega(d)$) whose degree-d truncations are hard to compute conditioned on the hardness of integer factorization or computation of $n!$. See Theorem 30 for more details.

Continuing the study of the complexity of truncated power series, we move on to algebraic power series. Here we work with constant-free circuits (i.e. constants like 2^n has to be built up from 1, requiring $O(\log n)$ many gates; for formal definition, see Section 5.2). It is not hard to show that the n-th coefficient of the *integral* power series expansion of $\sqrt{1+4x}$ (which has minpoly $y^2 = 1 + 4x$, of degree 2) is hard to compute (implying the truncation must be hard to compute, by a constant-free circuit as well) unless integer factoring is easy [11]; this follows from the well-known reductions: integer-factoring $\leq_{\mathsf{P}}$ computing $n!$ $\leq_{\mathsf{P}}$ computing $\binom{2n}{n}$ [2]; for a self-contained proof we refer to Theorem 72.

[2] here computation of an integer means, by a straight-line program or a constant-free circuit, see Definition 11

Can we show such a result for simple[3] algebraic functions when the minpoly has degree > 2? For instance, for $\sqrt[3]{1+9x}$? Here, 9 is just to make the power series integral. It is not at all clear, how the n-th coefficient of $\sqrt[3]{1+9x}$, namely $3^n/n! \prod_{j=0}^{n-1}(1-3j)$, helps in integer factoring (or in efficiently computing a multiple of $n!$). However, it turns out that the product of the n-th coefficients of $\sqrt[3]{1+9x}$ and $\sqrt[3]{(1+9x)^2}$, is a divisor of $3^n(3n)!/(n!)^3$; and computing it efficiently implies both the consequences. Exploiting the product of such binomial coefficients leads us to the following generalization; for details see Theorem 32 and Theorem 35.

▶ **Theorem 8** (Truncation of algebraic power series). *Let $k \in \mathbb{N}$. Then, there exists $1 \leq i < k$ with $i \in \mathbb{N}$, such that truncation of the integral power series $(1+k^2x)^{i/k}$ cannot have small constant-free circuits unless (i) integer factoring is easy (in the non-uniform setting) (see Algorithm 1), or (ii) some multiple of $n!$ is easy to compute (i.e. by a small straight-line program).*

▶ Remark 9.

a. [42] showed that if $n!$ is easy to compute, then integer factoring must be easy as well. However, it is not clear whether such statement can be drawn from some multiple of $n!$. Thus, (i) may not reduce to (ii) (& vice-versa). For details and definitions, see Section 6.

b. We also show that the hardness of the truncation of the above power series implies that permanent *requires* superpolynomial-size constant-free circuits, implying $\mathsf{VP}_0 \neq \mathsf{VNP}_0$; in fact, assuming GRH (Generalized Riemann Hypothesis), it implies $\mathsf{VP}_{\mathbb{C}} \neq \mathsf{VNP}_{\mathbb{C}}$. This is reminiscent of [9]. For details, we refer to Appendix H.

Finally, we move to the truncations of transcendental functions, where we show, to our surprise that there do exist some integral transcendental power series whose initial segments are easy to compute. Thus, transcendental power series *does not necessarily mean hard.* We refer the readers to Section 7.1 for the detailed formal statements.

▶ **Theorem 10** (Informal). *There are integeral transcendental power series whose truncations are easy.*

Therefore, Theorem 6–Theorem 10 together help in getting a good picture of the characterization sought in Problem 2.

1.2 Limitations of known techniques

We first discuss why standard techniques for division elimination and computing the truncations of power series do not yield the results we discover.

For the division problem, we first discuss why the division elimination method, due to Strassen [45], leads to an exponential blowup in size.

Strassen's division elimination. For $g(x_1,\ldots,x_n)/h(x_1,\ldots,x_n)$, wlog, assume that $h(0,\ldots,0) = 1$ (if not, then shift x_i by a *random value* α_i and get $h(\alpha_1,\ldots,\alpha_n)$ as a non-zero constant, which can be made 1, by scaling). Now, $f = g/h = g/\left(1-(1-h)\right) = g\sum_{i=0}^{\infty}(1-h)^i$. Here, we use the inverse identity: $1/(1-x) = \sum_{i\geq 0} x^i$. Assume that, f has degree d. Note that, $\tilde{f} := g\left(1+(1-h)+(1-h)^2+\cdots+(1-h)^d\right)$, has a poly$(s,\log d)$ size circuit. Moreover, as $1-h$ is constant-free, truncation of $\tilde{f}$ upto degree-d (denoted as $\mathrm{Hom}_{\leq d}\tilde{f}$), correctly computes f.

[3] here simple means that the degree of the minpoly of the algebraic functions and the degree of the coefficients of minpoly are both bounded by a constant

Howbeit, computationally, the truncation incurs a poly(d)-size multiplicative blowup. In general, given a polynomial f, computed by a circuit of size s, it is *unlikely* that we can always get poly($s, \log d$)-size circuit for the polynomial $\mathrm{Hom}_{\leq d} f$, unless, permanent has a small circuit (see Lemma 69 for a proof of this well-known fact). In fact, every method to eliminate divisions which uses truncation, (for instance, Newton iteration, see [48], Kaltofen's Hensel-lifting [22, 23], or allRootNI-technique via logarithmic-derivative [17]) give polynomial dependence on the degree (or the square-free part) of the quotient polynomial f; both can be large.

For computing the truncation of power series of rational functions, Kung and Treib [29] used Newton iteration which also works, more generally, for all algebraically functions. However, the problem with Newton iteration is that even though the precision doubles with each iteration, there is always an error term as well (see [29] for details). So, if we want to exactly compute the polynomial up to degree d, we need to truncate in order to get rid of the error terms. This again, due to the reasons described above, incurs a poly(d)-size multiplicative blowup, and is unlikely to be possible with an overhead bounded by poly($\log d$).

1.3 Proof idea

Our proofs are simple and use natural ideas combined with some subtle observations and careful maneuvering. We denote $\mathbf{x} = (x_1, \ldots, x_n)$.

Division by low-degree polynomial: Proof idea of Theorem 4. As a warm up, we first show a similar theorem for univariate polynomials which is a much simpler case, yet it constitutes the fundamental idea.

Division by a low-degree polynomial for univariates. Let g be a univariate polynomial in $\mathbb{F}[x]$, computable by an arithmetic circuit C, and we want to divide it by degree-d univariate polynomial h. We do this by splitting each gate of C into two parts – one computing the quotient and the other computing the remainder when divided by h (denoted by div h, and mod h respectively); they are computed corresponding to each gate of the circuit, in the bottom-up manner.

In case of a '$+'$ gate, the corresponding quotient and the remainder are precisely the sum of the quotients and the remainders corresponding to its children gates. While for a '$\times'$ gate with its children computing polynomials $p_1 = q_1 h + r_1$ and $p_2 = q_2 h + r_2$, we have $p_1 p_2 \bmod h = r_1 r_2 \bmod h$, and $p_1 p_2 \text{ div } h = q_1 q_2 h + q_2 r_1 + q_1 r_2 + r_1 r_2 \text{ div } h$. Thus, apart from combining the outputs of the children gates, we also need to compute the quotient and the remainder of the product of the remainders of the two children ($r_1 r_2$ div h and $r_1 r_2$ mod h), which is unclear. However, if we are in the regime where the degree of h is low, then both $r_1 r_2$ div h and $r_1 r_2$ mod h will have low degree. So, we can use a simple fact that every univariate polynomial of degree at most d is trivially computable by an arithmetic circuit of size $O(d)$. This is sufficient to complete the proof (see Section 3.1 for details).

Going from univariates to multivariates. Here, the strategy is to somehow exploit the core idea used in the univariate setting. The very first step is to view the polynomials $g(\mathbf{x})$ and $h(\mathbf{x})$ as univariates in x_n, and also see $h(\mathbf{x})$ as a monic polynomial in x_n (wlog) where the coefficients are polynomials in the variables $x_1, \ldots, x_{n-1}$,. This step is fairly standard and is achieved via an invertible linear transformation (see Appendix C).

Now, the obvious idea of splitting each gate in the circuit of g into two gates computing the quotient and remainder simultaneously, *fails* directly, as a polynomial whose degree with respect to x_n is bounded by d, *may not* be computable by a poly(d)-size circuit.

To overcome this, we need a subtler observation from the univariate case. Recall that apart from combining the output from children gates, the *only* extra quotient and remainder computation that need to be done locally for a "×" gate are $r_1r_2 \bmod h$ and r_1r_2 div h. Since, $\deg(r_1), \deg(r_2) \leq d-1$, we need to compute the quotient and remainder of a polynomial of degree at most $2d-2$. We show that when we divide a polynomial of degree d_1 by a polynomial of degree d_2, then there exists a circuit of size $O(d_1d_2)$ which takes as input the coefficients of both the polynomials and outputs the coefficients of the quotient and remainder polynomials (see Lemma 16). In the univariate case, this gives a multiplicative blowup of $O(d^2)$ which is *worse* than plugging in the trivial circuits of the quotient and the remainder (trivial circuit has size $O(d)$). However, the advantage this offers is that it also extends to the multivariate case (see Lemma 16). There, the degree refers to the degree wrt x_n, and instead of coefficients of the polynomials r_1r_2 and h as the inputs, we have the circuits for their coefficients (viewed as univariates in x_n) as inputs.

This also suggests the right structure to maintain in the circuit throughout. Since we also need the circuits for the coefficients of the remainder, we split each gates in the circuit of $g(\mathbf{x})$ into $d+1$ gates: d gates to maintain the remainder, and the $(d+1)$-th gate to maintain the quotient. Note that, since the degree in x_n is bounded by d, hence the degree (wrt x_n) of the remainder $\leq d-1$, and the d remainder gates compute the corresponding coefficients (which will be polynomials in $x_1, \ldots, x_{n-1}$). We also need the coefficients of $h(\mathbf{x})$, when viewed as a univariate in x_n; this can be efficiently done with a small blowup using standard techniques (see Lemma 61). It turns out that the above idea suffices in the multivariate setting, see Section 3.2 for details.

Going to border. It turns out that our proof technique is *robust* to taking approximations, in the sense of border (or approximative) complexity, used in algebraic and geometric complexity theory (see Section 3.3 for details). The only subtle difference from the non-border case is that here the degree of the approximate circuit for h can be *large* (over $\mathbb{F}(\epsilon)[\mathbf{x}]$), but thanks to *homogenization* (Lemma 62) which would keep the degree (in $\mathbf{x}$) low throughout.

Truncation of rational function: Proof idea of Theorem 6. Here, the core idea is to use *partial fraction decomposition* of rational functions. Over an algebraically closed field ($\mathbb{F} = \overline{\mathbb{F}}$), this allows us to decompose an arbitrary rational function $g(x)/h(x)$ (with $\deg(g) < \deg(h)$) as a sum of rational functions, each of the form $b/(x-a)^i$, where $a, b \in \mathbb{F}$ (see Lemma 24); this basically follows from factoring of h over $\mathbb{F}[x]$ (and thus the a's are roots of h).

When, $\deg(h)$ is small, number of such $b/(x-a)^i$ is also small. Moreover, the truncations of the $1/(x-a)^i$, for $a \neq 0$, is easy to compute (see Section 5.1). But there are two subtle issues to be handled: (i) what to do when $a = 0$? and (ii) what happens when $\deg(g) > \deg(h)$?

Theorem 4 along with some basic analysis turns out to be the savior for both the cases.

For the first issue, note that $a = 0$ implies x^m divides h for some $m \geq 1$. However, as $g/h \in \mathbb{F}[[x]]$, it turns out that x^m must also divide g, for such power series to exist (Lemma 65). Thus, it suffices to work with $g/h = g_1/h_1$, where $g_1 := g/x^m$ and $h_1 := h/x^m$, both being polynomials in $\mathbb{F}[x]$. But what happens to the size of g_1 ? Well, thanks to Theorem 4: as, m is small (because $m \leq \deg(h)$), it turns out that the circuit complexity of g_1 is also small.

For the second issue, note that, $\deg(g) > \deg(h)$ implies $\deg(g_1) > \deg(h_1)$. But thanks to Theorem 4 again. Of course, $g_1/h_1 = g_1 \text{ div } h_1 + (g_1 \bmod h_1)/h_1$. Thus, g_1 div h_1 and $g_1 \bmod h_1$ have small complexity and moreover $\deg(g_1 \bmod h_1) < \deg(h_1)$. Additionally, g_1 div h_1 has degree $< d$ (as $\deg(g) \leq d$). Thus, combining all these, the conclusion follows.

Extending to SOS-complexity. We remark that, similar proof works wrt SOS-complexity when both g and h have constant-degrees. This is mainly because $1/(1-x)^i$ has small SOS-complexity as SOS-model is *closed* under small derivatives (Lemma 51). For details, see Theorem 49.

Truncation of algebraic functions: Proof idea of Theorem 8. There are two parts of the proof. But before delving into that, it is not hard to show that $(1+k^2x)^{1/k}$ is an integral power series; this can be proved by some basic number-theoretic tools, for details see Theorem 77.

For the first part, we show that easiness of the truncation of each $(1+k^2x)^{i/k}$, for all $i \in [k-1]$, leads to an efficient integer factoring algorithm (Algorithm 1). This algorithm is a subtle generalization of the algorithm of [30]. Note that from binomial expansion, coefficient of x^d in $(1+k^2x)^{i/k}$ is $C_{d,i} := k^d/d! \cdot \prod_{j=0}^{d-1}(i-kj)$. Moreover, when the truncations are easy, the coefficients are also easily computable, just by subtracting two consecutive truncatations and substituting $x=1$. For a fixed i and $k \geq 3$, it is not clear how $C_{d,i}$ behaves (when $k=2$, it is $= \binom{2d}{d}/(2d-1)$). However, if we take product of all the d-degree coefficients (i.e. $\prod_{i\in[k-1} C_{d,i}$), it turns out to be a "nicer" quantity. In particular, one can show that this product is a divisor of the integer $N(d,k) := k^{(k-2)d}\,(dk)!/(d!)^k$. Moreover, $N(d,k)$ turns out to be easily computable as well.

Can we exploit any property of $N(d,k)$ which could help us factor an integer n? Well, as $N(d,k)$ is easy, computing gcd of $N(d,k)$ and n is also easy. If we can figure-out a d such that $\gcd(N(d,k),n) \neq 1,n$, we have already found a factor! So the aim is to somehow reduce the search space cleverly and find a suitable d. Wlog, one can assume that all the factors of n are greater than k (otherwise we can remove them by brute-force, as k is constant). Now, we try to find the smallest prime p dividing n. Of course, there must exist $t \in S := \{k, k^2, \ldots, k^\ell\}$, where $k^\ell \leq n/k$, such that $p \in [t+1, tk]$ (as these disjoint intervals cover $[n]$). Note that $|S| = \log n$. Also, trivially $p \mid N(t,k)$, as p divides the numerator but cannot divide the denominator. So, if the $\gcd(N(t,k),n) \neq n$, we are done. But, if the gcd becomes n, it simply implies all the prime factors of n must lie in the interval $[t+1, tk]$.

Unfortunately, this interval size is still huge and we cannot brute-force over it. But, we can further reduce our search space by binary search. This idea is similar to [30]; each time we halve the search interval to reduce the search space for candidate d such that $\gcd(N(d,k),n) \neq 1,n$. At first, we have two integers a,b with $a=1$ and $b=t$ such that the prime factors are in $[ak+1, bk]$. Fix $c=(a+b)/2$ and compute $\gcd(N(c,k),n)$. If the gcd is $\neq 1,n$, we are done, otherwise we branch accordingly into the first half or the second. When the gcd is 1, it must happen that the factors are in the second half i.e.$[ck+1, bk]$. When gcd $= n$, the factors are in the first half $[ak+1, ck]$. After at most $\log n$ steps, we must have either found a factor and if not, we have found a *small* interval $[sk, (s+1)k]$ of length k where all the prime factors lie. We can now brute-force to find the factors. For details, see Section 6.1 and Algorithm 1.

The second part eventually exploits and recurse on the fact that $(dk)!/(d!)^k$ is easy to compute and $(d!)^k$ is easy when $(d!)$ is easy, implying a clear pattern of recurrence from dk to d (Section 6.2).

Truncation of transcendental power series. Finally, for showing Theorem 10 about transcendental power series, we discover some explicit integral power series whose initial segments are *non-sparse* yet easy to compute. For this purpose, we use *stern sequences* (Section 7.1) and power series whose coefficients are multiplicative, and exploit their recursive structures. Conversely, we show hardness for the truncation of an integral transcendental power series defined via *holonomic sequences* (Section 7.2).

2 Preliminaries

Notation. We denote $\mathbf{x} = (x_1, \ldots, x_n)$. $[n]$ denotes the set $\{1, \ldots, n\}$. For a polynomial $f \in \mathbb{F}[\mathbf{x}]$, we denote up to degree-d part as $\mathrm{Hom}_{\leq d} f$ and $|f|_0$ as the sparsity or the number of monomials in f. For a differentiable function $f(x)$, we denote $f^{(k)}(x) := d^k f/dx^k$, as the k-th derivative of f. We also recall the definition of gcd of two polynomials f, g in the ring $\mathbb{F}[\mathbf{x}]$: $\gcd(f,g) =: h \Leftrightarrow h \mid f, h \mid g$, and $h' \mid f, g \Longrightarrow h' \mid h$. It is unique up to constant multiples.

Field. We denote the underlying field as $\mathbb{F}$ and assume that it is algebraically closed. All our results hold when the characteristic is large or not algebraically closed, as we can go to polynomial extensions and work with it.

Binomial series. For rational n, $(x+a)^n = \sum_{k \geq 0} \binom{n}{k} x^k a^{n-k}$, where $\binom{n}{k} = n \cdot (n-1) \cdot \cdots (n-k+1)/k!$.

div and mod operations. For polynomials f and $g \in \mathbb{F}[x]$, if $f = g \cdot h + r$, where $h, r \in \mathbb{F}[x]$ such that $\deg(r) < \deg(g)$, then h is called the quotient, denoted f div g, and r is called the remainder, denoted f mod g. Operation mod may not be well-defined in the multivariate settings, however, if one assumes g to be monic in a variable say x_n, it is always well-defined (by thinking g to be a univariate in x_n). A polynomial g is monic in x_n if the leading coefficient (the nonzero coefficient of highest degree) of x_n is a non-zero constant in $\mathbb{F}$. Of course, if $g \mid f$, then f div $g = h$ and f mod $g = 0$, irrespective of monic-ness.

Power series and truncation. A formal power series is a generalization of a polynomial, where the number of terms can be infinite. Formally, $A = \sum_{i \geq 0} A_i x^i$ with $A_i \in \mathbb{F}$, is a power series in the power series ring $\mathbb{F}[[x]]$. We define the degree d truncation $\mathrm{trunc}(A, d)$ of A to be $\mathrm{trunc}(A, d) := \sum_{0 \leq i \leq d} A_i x^i$. So, $\mathrm{trunc}(A, d)$ is always a polynomial of degree at most d.

▶ **Definition 11** (Straight Line Program). *An SLP (straight line program) P (for computing an integer) of length (or size) n is a sequence of integers $a_0, \ldots, a_n$ with $a_0 = 1$ and $a_k = a_i \circ a_j$ with $i, j < k$ for $\circ \in \{+, -, \times\}$. We say that the SLP P computes the integer a_n. For an integer N, we define the straight line complexity $\tau(N)$ of N to be the length of the smallest SLP computing N.*

▶ **Definition 12** (Algebraic and Transcendental Power Series). *A formal power series $f \in \mathbb{C}[[x]]$ is said to be algebraic if there exists a polynomial $g \in \mathbb{C}[x][t]$ such that $g(f) = 0$. Otherwise f is said to be transcendental.*

With abuse of notation, for integers, we will sometime use complexity of the integer (implying $\tau(\cdot)$ only). Sometimes we also allow division as a operation in straight line program (each time we mention if so). For a polynomial $f \in \mathbb{F}[\mathbf{x}]$, we define the complexity $L_{\mathbb{F}}(f)$ of f to be the length of the smallest division-free arithmetic circuit (with only $\{+, -, \times\}$ gates) computing f. We also define, the complexity $\tau_{\mathbb{F}}(f)$ of f to be the length of the smallest division and constant-free arithmetic circuit computing f (all the constants are made from 1), for formal definition see Section 5.2. We will remove subscript $\mathbb{F}$ when the underlying field is clear from the context.

3 Division elimination in high-degree circuits

This section deals with Problem 1, where the divisor has small degree and proves Theorem 4. Section 3.1 shows it in the univariate setting while Section 3.2 deals with the multivariate setting, and finally, Section 3.3 shows an analogous theorem in the border complexity setting. Here, we remark that formally, one should use $f_d = g_d/h_d$, with d as an index, however with abuse of notation, we use g/h throughout the paper.

3.1 Division of Univariate Polynomials

The following theorem deals with Problem 1 in the univariate setup.

▶ **Theorem 13.** *Let g, h be polynomials in $\mathbb{F}[x]$. If $L(g) = s$ and $\deg(h) = d$, then both $L(g \text{ div } h)$ and $L(g \bmod h)$ have complexity $O(sd)$.*

Proof. Suppose C is a circuit of size s which computes g. We split every gate Φ in C into two gates Φ_1 and Φ_2, to make a new circuit C', which computes both g div h and g mod h. If Φ is computing some polynomial ϕ in C, then Φ_1 computes the polynomial $\phi \bmod h$ and Φ_2 computes the polynomial ϕ div h.

The proof is inductive and traverses from bottom to the top. The base case is trivial. At some step, say that we are at a gate Φ. The children gate of ϕ are computing polynomials α and β. Let, $\alpha = q_1h + r_1$, $\beta = q_2h + r_2$ and $\phi = qh + r$, where the degrees of r, r_1, r_2 are smaller than d. So in the new circuit C', we have already computed r_1, q_1, r_2, q_2. If Φ is a $\pm$ gate then it is clear that $r = r_1 \pm r_2$ and $q = q_1 \pm q_2$. If Φ is a $\times$ gate then we have:

$$r \;=\; (r_1r_2) \bmod h\,, \text{ and } q \;=\; q_1q_2h + r_1q_2 + r_2q_1 + (r_1r_2) \text{ div } h.$$

We know that r is a polynomial of degree at most $d-1$. Since, $\deg(r_1r_2) \le 2d-2$, we get that $\deg((r_1r_2) \text{ div } h) \le d-2$. Therefore, we trivially have that: $L(r) = O(d)$ and $L((r_1r_2) \text{ div } h)) = O(d)$. Hence we can compute r, q using additional $O(d)$ many gates. Thus, C' has at most $O(sd)$ many gates. Hence $L(g \text{ div } h) = O(sd)$ (same for g mod h). ◀

▶ **Corollary 14.** *For $f, g, h \in \mathbb{F}[x]$, if $f = g/h$ with $L(g) = s$ and $\deg(h) = d$ then $L(f) = O(sd)$.*

▶ Remark 15. The polynomial $f_d := 1 + \cdots + x^d = (x^{d+1} - 1)/(x - 1)$ has $O(\log d)$ size circuit. This can also be shown via a recursive computation argument.

Can we expect both div and mod to have poly$(s, \log d)$-size circuits? We show that it is highly unlikely unless factoring is *easy*, see Theorem 54 for details.

3.2 Division of Multivariate Polynomials

This section deals with division in the multivariate setting. But before that, we solve a particular case (by folklore techniques) which will play a crucial role to prove the main Theorem 18. For a proof of the following Lemma 16, see Appendix E.

▶ **Lemma 16.** *Suppose $g = \sum_{i \le d_1} g_i x^i$ and $h = x^{d_2} + \sum_{i < d_2} h_i\, x^i$, in $\mathbb{F}[\mathbf{x}]$. Suppose $g = hq + r$, with $r = \sum_{i<d_2} r_i x^i$ and $q = \sum_{i \le d_1 - d_2} q_i x^i$. Then, there is a circuit of size $O(d_1\, d_2)$, whose inputs are all h_i, g_i and outputs are all r_i, q_i.*

Now we prove the following Lemma 17 which shows that both div and mod have low complexity when the divisor has low-degree and monic (in fact, constant leading-coefficient suffices).

▶ **Lemma 17** (Main Lemma)**.** *Let the polynomials $g, h \in \mathbb{F}[\mathbf{x}]$ such that h is monic in x_n, $L(g) = s_1, L(h) = s_2$, and $\deg_{x_n}(h) = d$. Then, both $L(g \text{ div } h), L(g \bmod h) \leq O((s_1 + s_2)\, d^2)$.*

Proof. Suppose C is a circuit of size s_2 which computes h and C_g is the circuit of size s_1 which computes g. By using Lemma 61, there is a circuit of size $O(s_2\, d^2)$, which computes $h_0, \cdots, h_{d-1}$.

Now, we split every gate F in C into $d+1$ gates $F_0, F_1, \ldots, F_d$. Suppose, the gate F is computing a polynomial P_F. Let $P_F \bmod h = \sum_{i<d} p_i x_n^i$. Then we want the property that F_i computes p_i for $i < d$. And if $i = d$ then F_i computes P_F div h.

Suppose F is a $+$ gate in C with children gates computing the polynomials a and b. Again express $a \bmod h = \sum_{i<d} a_i x_n^i$ and $b \bmod h = \sum_{i<d} b_i x_n^i$. It is clear that

$$(a+b) \bmod h \;=\; a \bmod h \;+\; b \bmod h.$$

Therefore $p_i = a_i + b_i$. It is also clear that P_F div h $=$ a div h $+$ b div h.

Suppose F is a $\times$ gate in C with children gates computing the polynomials a and b. Again express $a \bmod h = \sum_{i<d} a_i x_n^i$ and $b \bmod h = \sum_{i<d} b_i x_n^i$. It is clear that:

$$(a \cdot b) \bmod h \;=\; (a \bmod h \cdot b \bmod h) \bmod h.$$

For div , we have that:

$$P_F \text{ div } h \;=\; a \text{ div } h \cdot b \text{ div } h \cdot h + b \text{ div } h \cdot a \bmod h + a \text{ div } h \cdot b \bmod h + (a \bmod h \cdot b \bmod h) \text{ div } h.$$

We have already computed $a \bmod h, b \bmod h, a$ div h, b div h. So, we only need to compute $(a \bmod h \cdot b \bmod h) \bmod h$ and $(a \bmod h \cdot b \bmod h)$ div h. Since we have already computed a_i, b_i for all $i < d$, by using Lemma 16, we can compute all the p_i and $(a \bmod h \cdot b \bmod h)$ div h in $O((2d-2)d) = O(d^2)$ many gates. Therefore the new circuit has $O(s_1 d^2)$ has many gates. Also we used $O(s_2 d^2)$ gates to computes $h_0, \cdots, h_{d-1}$. Hence,

$$L(g \text{ div } h) = O((s_1+s_2)d^2), \text{ and } L(g \bmod h) = O((s_1+s_2)d^2)\,. \qquad ◀$$

The following theorem *settles* Problem 1, when the divisor has small degree (proving Theorem 4).

▶ **Theorem 18** (Division elimination for low-degree divisor)**.** *Let the polynomials $f, g, h \in \mathbb{F}[\mathbf{x}]$ such that $f = g/h$, with $L(g) = s_1, L(h) = s_2$, and $\deg(h) = d$. Then, $L(f) \leq O((s_1 + s_2)\, d^2)$.*

Proof. The above Lemma 17 shows that when h is monic in x_n, the upper bound holds. Let $\tau : \mathbb{F}[\mathbf{x}] \longrightarrow \mathbb{F}[\mathbf{x}]$, be an *invertible* monic transformation (sends $x_i \mapsto \alpha_i \cdot x_n + x_i$, where $\alpha_i \in \mathbb{F}$) s.t. $\tau(h)$ is *monic* wrt x_n, such transformation exists (Lemma 68). Note that, $L(\tau(g)) \leq s + n = O(s_1)$, and $L(\tau(h)) \leq s_2 + n = O(s_2)$. Moreover, as τ is degree-preserving, $\deg_{x_n}(\tau(h)) = d$.

So, apply Lemma 17 to conclude that $\tau(f) = \tau(g)$ div $\tau(h)$, has a circuit of size $O((s_1 + s_2)d^2)$. We apply τ^{-1} again (which is just a additive n-blowup) to finally deduce that

$$L(f) \;\leq\; O((s_1+s_2)\, d^2)\,. \qquad ◀$$

▶ Remark 19. This proof holds when one replaces L by τ, i.e. the constant-free circuit complexity (for definition, see Section 5.2). Note that, neither div nor mod introduce any new constant in the process. Moreover, one can choose the α_i to be explicit and poly$(\log d)$-computable so that τ is a monic invertible map. This establishes the claim.

3.3 Division in border complexity

The notion of border (equivalently, approximative) complexity is important in computer science. This concept popped up from early works on matrix multiplication and border rank of tensors, see [10]). Whether *approximation* of polynomials provides any additional computational power is a natural question which fundamentally motivated the foundation of Geometric Complexity theory (GCT). The notion of border complexity can be motivated through two ways: *topological* and *algebraic*, and both the perspectives are known to be *equivalent* [1]. For further details, we refer to [20, 36].

In this paper, we only work with algebraic approximation upper bounds. In the algebraic definition, one can talk about the *convergence* $\epsilon \to 0$. Here, one can see ϵ as a formal variable and $\mathbb{F}(\epsilon)$ as the function field. For an algebraic complexity class C, the approximation is defined as follows [7, Definition 2.1].

▶ **Definition 20** (Approximative closure of a class [7]). *Let C be an algebraic complexity class over field $\mathbb{F}$. A family (f_n) of polynomials from $\mathbb{F}[\mathbf{x}]$ is in the* class $\overline{C}(\mathbb{F})$ *if there are polynomials $f_{n;i}$ and a function $t : \mathbb{N} \mapsto \mathbb{N}$ such that g_n is in the class C over the field $\mathbb{F}(\epsilon)$ with $g_n(\mathbf{x}) = f_n(\mathbf{x}) + \epsilon f_{n,1}(\mathbf{x}) + \epsilon^2 f_{n,2}(x) + \cdots + \epsilon^{t(n)} f_{n,t(n)}(\mathbf{x})$.*

▶ **Definition 21** ([8, Defn.3.1]). *Let $f \in \mathbb{F}[\mathbf{x}]$. The* border complexity *$\underline{L}(f)$ is the smallest number r, such that there exists F in $\mathbb{F}(\epsilon)[\mathbf{x}]$ satisfying $F|_{\epsilon=0} = f$ and $L_{\mathbb{F}(\epsilon)}(F) \leq r$.*

Note that, the circuit of F may be using $1/\epsilon$ in an intermediate step. So, we cannot merely assign $\epsilon = 0$ and get a ϵ-free circuit. Also, the ϵ-degree can be exponential in its size (and thus cannot be interpolated), see [8, Theorem 5.7]). Thus, potentially $\underline{L}(f)$ can be significantly smaller than $L(f)$.

The above definition can be used to define closures of complexity class, e.g., $\overline{\mathsf{VP}}$. In this case, one can assume wlog that the degrees of g_n and $f_{n,i}$ are poly(n). It is known to be *closed under factoring* [8, Theorem 4.1]. However, the usual method of Hensel-lifting *does not* work when the given circuit class computes polynomials of super-polynomial degree. Also, Strassen's method would have a dependency on the degree of the final polynomial. However, we can prove Theorem 18 analogously, in the border sense.

▶ **Theorem 22** (Division elimination in border complexity). *Let $f, g, h \in \mathbb{F}[\mathbf{x}]$, such that $f = g/h$, with $\underline{L}(g) = s_1$, $\underline{L}(h) = s_2$, and $\deg(h) = d$. Then, $\underline{L}(f) \leq O(s_1\, d^2 + s_2\, d^4)$.*

Proof. By definition, there exists $G, H \in \mathbb{F}(\epsilon)[\mathbf{x}]$, of size at most s_1 and s_2, respectively, such that $G := g+\epsilon\cdot\tilde{g}(\mathbf{x}, \epsilon)$, and $H := h+\epsilon\cdot\overline{h}(\mathbf{x}, \epsilon)$, where $\tilde{g}, \overline{h} \in \mathbb{F}[\epsilon, \mathbf{x}]$. We note that, $\deg_{\mathbf{x}}(H)$ can be *larger* than d. However, using Lemma 62, we know that $L_{\mathbb{F}(\epsilon)}(\text{Hom}_{\leq d} H) \leq O(s_2 d^2) := s_2'$.

We denote $\tilde{H} := \text{Hom}_{\leq d} H$. It is important to observe that $\tilde{H}|_{\epsilon=0} = h$. By definition, there exists m (could be $\exp(s_2')$) such that

$$\tilde{H} := h + \epsilon \cdot \tilde{h}(\mathbf{x}, \epsilon) \;=\; h + \sum_{j \in [m]} \epsilon^j \cdot h_j(\mathbf{x})\,,\ \text{ where } h_j \,\in\, \mathbb{F}[\mathbf{x}]\,.$$

Let $\tau : \mathbb{F}[\mathbf{x}] \longrightarrow \mathbb{F}[\mathbf{x}]$, be an *invertible* monic transformation (sends $x_i \mapsto \alpha_i \cdot x_n + x_i$, where $\alpha_i \in \mathbb{F}$) s.t. $\tau(h)$ and each $\tau(h_j)$, for $j \in [m]$ is *monic* wrt x_n; such transformation exists (Lemma 68). Note that, $L_{\mathbb{F}(\epsilon)}(\tau(G)) \leq O(s_1)$ and $L_{\mathbb{F}(\epsilon)}(\tau(\tilde{H})) \leq O(s_2')$. Further, $\deg_{\mathbf{x}}(\tau(\tilde{H})) = d$, as τ is a degree-preserving map. We also have the following identities:

$$\tau(\tilde{H}) = \tau(h) + \epsilon \cdot \tau(\tilde{h}) \text{ and } \tau(G) = \tau(f) \cdot \tau(h) + \epsilon \cdot \tau(\tilde{g})\,.$$

By assumption, the leading coefficient of x_n in $\tau(\tilde{H})$ (call it α) is in $\mathbb{F}[\epsilon]$ (in fact, $\alpha \not\equiv 0 \bmod \epsilon$). This basically makes $\tau(\tilde{H})$ a monic polynomial over $\mathbb{F}(\epsilon)[\mathbf{x}]$. Therefore, div $\tau(\tilde{H})$ and mod$\tau(\tilde{H})$ now make sense over $\mathbb{F}(\epsilon)[\mathbf{x}]$. By simple division, we have

$$\tau(G) \text{ div } \tau(\tilde{H}) \;=\; \tau(f) + \epsilon \cdot \left(\left(\tau(\tilde{g}) - \tau(f) \cdot \tau(\tilde{h})\right) \text{ div } \tau(\tilde{H})\right) . \qquad (1)$$

Note that, Lemma 17 implies $L_{\mathbb{F}(\epsilon)}\left(\tau(G) \text{ div } \tau(\tilde{H})\right) = O((s_1 + s_2')d^2)$. By definition of $\underline{L}$ and Equation (1), it is trivial to conclude that $\underline{L}(\tau(f)) \leq O((s_1 + s_2')d^2) = O(s_1\, d^2 + s_2\, d^4)$. As τ is invertible, we can get back f by applying τ^{-1} (incurring n-additive blowup). This finally shows

$$\underline{L}(f) \;\leq\; O(s_1\, d^2 + s_2\, d^4) \,. \qquad ◀$$

4 Implications of division elimination in algebraic complexity

An affirmative solution to Problem 1 would have nontrivial applications in algebraic complexity. We briefly discuss some of them in the next few paragraphs.

Division elimination in border complexity. It is not clear whether a positive solution to Problem 1 would resolute to solving $\mathrm{VP} = \overline{\mathrm{VP}}$ (the converse direction is also not clear). Note that, an approximative circuit can use arbitrary scalars from the field $\mathbb{F}(\epsilon)$. So it is not clear if the polynomial computed by an approximative circuit of size s can be expressed as g/h, where $g, h \in \mathbb{F}[\epsilon, \mathbf{x}]$ can be computed by circuits (using constants from $\mathbb{F}$) of size poly(s). However, a special case of Problem 1, when the denominator is as simple as x^d, is open, and it has interesting implications as we discuss. The following example is from Bürgisser [8], which relates the complexity of *trailing coefficient* of a polynomial to the complexity of the polynomial itself.

Let us take a polynomial $f(\mathbf{x}, \epsilon) \in \mathbb{F}[\mathbf{x}, \epsilon]$ computed by an arithmetic circuit of size s (over $\mathbb{F}$). Suppose, $f := \sum_{i=d}^{D} C_i(\mathbf{x})\, \epsilon^i$ where C_i are polynomials in $\mathbb{F}[\mathbf{x}]$. The trailing coefficient of f wrt ϵ, which is the polynomial C_d can be computed by a circuit of size poly(s, d), by homogenization. Note that d can be exp(s). In contrast, it can be computed by an *approximative* circuit of size just s. The approximative circuit C' computes the polynomial f/ϵ^d (as $\lim_{\epsilon \to 0} f/\epsilon^d = C_d$). Note that, ϵ^d has $O(\log d)$-size circuit. Now, a positive solution to Problem 1 would imply that f/ϵ^d has a division-free circuit C of size poly$(s, \log d)$. We can simply put $\epsilon = 0$ in C and compute C_d.

Division elimination in polynomial factoring. Another interesting consequence of the above mentioned case of Problem 1 would be the proof of Factor conjecture [23, 8]: Any factor g of a given polynomial f can be computed by poly$(s, \deg(g))$-size circuit. Bürgisser [8] gave an approximative circuit of poly$(s, \deg(g))$ that involves division by ϵ^d where ϵ can be seen as a formal variable. See [8, 19] for various consequences of Factor conjecture.

Division elimination and gcd. It turns out that the existence of small circuits for gcd and division elimination can resolve the *radical conjecture* [17]: the squarefree-part or the radical of a multivariate polynomial f of size s, has size poly(s).

The gcd question [23, Problem 4] asks whether given polynomials $f_1, \ldots, f_m$, computed by a circuit size s, their gcd $g := \gcd(f_1, \ldots, f_m)$ has size poly(s). Currently, the best known bound (due to Kaltofen [23]) is poly$(s, \deg(g))$. It is not hard to show that a positive resolution to both Problem 1 and gcd would also resolve the aforementioned radical conjecture.

In fact, it would also lead to poly(s) bound for computing the *reduced rational function*. Given a rational function p/q computed by a circuit (with division gates) of size s, compute the numerator and denominator in the reduced form ($g/h = p/q$, where g and h are coprime) in poly(s). Kaltofen [23, Problem 4] showed a bound of poly($s, \deg(g), \deg(h)$). Note that getting numerator and denominator of reduced rational function in poly(s) directly implies solution to both high degree division and gcd questions.

▶ Remark 23. It is known that given a polynomial f, computed by poly(s), all its factors *cannot* be computed by poly(s)-size circuits. For eg. $x^{2^s} - 1$; it has factors of size exp(s) [33]. However, this does not give a counterexample for Problem 1 (as the cofactor of a hard factor is also expected to be hard).

5 Circuit complexity of rational function truncation

First, we deal with rational functions. We show both upper bound and conditional lower bound results (relating to integer factoring).

5.1 Upper bounds for rational function truncation

We show that complexity of truncation of rational functions where the degrees are small, has low complexity. For simplicity, we work with $\mathbb{F} = \overline{\mathbb{F}}$, an algebraically closed field. We first recall the following folklore decomposition.

▶ **Lemma 24** (Partial fraction decomposition). *Let $g(x)/h(x)$ be a rational function with $\deg(g) < \deg(h)$. If $h(x) = \prod_{i\in[k]}(x-a_i)^{d_i}$ is the factorization of $h(x)$ over $\mathbb{F}[x]$, then, there exist $b_{ij} \in \mathbb{F}$ s.t. :*

$$g(x)/h(x) \;=\; \sum_{i\in[k]}\sum_{j\in[d_i]} b_{ij}/(x-a_i)^j\,.$$

Here is an important lemma which plays a crucial role in the size upper bound of truncation.

▶ **Lemma 25.** *For any non zero $a \in \mathbb{F}$, we have $L\left(\mathrm{trunc}\left(1/(x-a), d\right)\right) = O(\log d)$.*

Proof. This follows from the inverse identity: $1/(a-x) = 1/a\,\sum_{i\geq 0}(x/a)^i$ and the fact that $L(\sum_{0\leq i\leq d}(x/a)^i) = O(\log d)$ (By using Remark 15). ◀

Now, we prove Theorem 6. For brevity, we state it again.

▶ **Theorem 26** (Truncation of low-degree rational function). *Suppose, g and h are two univariate polynomials in $\mathbb{F}[x]$ such that $\deg(g) \leq d$, $\deg(h) = d_h$, and g can be computed a circuit of size s. Let, $g/h \in \mathbb{F}[[x]]$. Then, truncation of g/h upto degree-d can be computed by a circuit of size* poly($s, d_h, \log d$).

Proof. The main idea is to use Lemma 24 and the low complexity of the truncation of inverse identity (Lemma 25). However, the given polynomial h may be divisible by x (i.e. $h(0) = 0$). In that case, let m be the highest power of x such that $x^m \mid h$ (i.e. $x^{m+1} \nmid h$). Note that, as $g/h \in \mathbb{F}[[x]]$, $x^m \mid g$ as well (Lemma 65). As $\deg(h) \leq d_h$, thus $m \leq d_h$.

By using Theorem 18, we know that $g_1 := g/x^m$ has a cicuit of size $O((s+\log d_h)\,d_h^2) =: s_1$. Trivially, $h_1 := h/x^m$ has degree $\leq d_h$, and $g/h = g_1/h_1$. Denote, $g_2 := g_1 \bmod h_1$. Obviously, $\deg(g_2) < \deg(h_1)$ and $g_1/h_1 = g_1 \text{ div } h_1 + g_2/h_1$. Invoking Theorem 13, one concludes that $L(g_1 \text{ div } h_1) = O(s_1\,d_h)$. Therefore, $L(\mathrm{trunc}(g_1 \text{ div } h_1, d)) = O(s_1\,d_h)$, as $\deg(g_1) < d$.

Let h_1 factors over $\mathbb{F}[x]$ as $h_1 := \prod_{i\in[k]} (x-a_i)^{d_i}$. Trivially, $\sum d_i \le d_h$. By using Lemma 24 on g_2/h_1, we know that there are constants $a_i, b_{ij} \in \mathbb{F}$ such that:

$$g_2(x)/h_1(x) \;=\; \sum_{i\in[k]} \sum_{j\in[d_i]} b_{ij}/(x-a_i)^j \,.$$

Note that, for any $a \in \mathbb{F}$ and $t \in \mathbb{N}$, $d^t/dx^t\,(1/(x-a)) = (-1)^t\, t! \cdot \big(1/(x-a)^{t+1}\big)$, and thus,

$$\mathrm{trunc}(1/(x-a)^{t+1}, d) \;=\; (-1)^t/t! \cdot d^t/dx^t\;(\mathrm{trunc}(1/(x-a), d)) \;+\; \sum_{i=d-t+1}^{d} \gamma_i\, x^i\;, \text{ where } \gamma_i \in \mathbb{F}\,.$$

Using the above identity and Lemma 63, we can show that

$$L\left(\mathrm{trunc}\left(\sum_{j\in[d_i]} b_{ij}/(x-a_i)^j, d\right)\right) \;=\; O(\log d \cdot d_i^2)\,.$$

To show this, note that $L(\mathrm{trunc}(1/(x-a_i), d)) = O(\log d)$, and using Lemma 63, we compute all its derivative till the d_i-th one which has a circuit of size $O(\log d \cdot d_i^2)$. Using the above identity, we can add $d_i - 1$ many monomials of the form cx^ℓ with $d - d_i + 2 \le \ell \le d$ (each monomial has trivial size of $O(\log d)$) to the circuit to obtain a circuit for $\mathrm{trunc}\left(\sum_{j\in[d_i]} b_{ij}/(x-a_i)^j, d\right)$, which still has size $O(\log d \cdot d_i^2)$. Thus, doing it for each a_i for $i \in [k]$, one obtains that

$$\begin{aligned} L\,(\mathrm{trunc}\,(g(x)/h(x), d)) &= L\,(\mathrm{trunc}\,(g_1(x)/h_1(x), d)) \\ &= L\,(g_1 \text{ div } h_1, d) \;+\; L\,(g_2/h_1, d) \\ &= O(s_1\, d_h) \;+\; L\left(\mathrm{trunc}\left(\sum_{i\in[k]}\sum_{j\in[d_i]} b_{ij}/(x-a_i)^j, d\right)\right) \\ &= \; O((s + \log d_h)\, d_h^3) \;+\; O(\log d \cdot \sum_{i\in[k]} d_i^2) \\ &= \; O(s\, d_h^3 \log d). \end{aligned}$$

◀

▶ Remark 27. Eventually, we can replace $g \in \mathbb{F}[[x]]$ with the given complexity $\mathrm{trunc}(g, d) = s$ and show that the exact same proof as above, works.

5.2 Hardness results for rational function truncation

Now, we give some evidence that we cannot expect logarithmic dependence on d_h in Theorem 26, unless integer factoring is *easy*. Before going into technicalities, we define *easy* sequence and constant-free complexity.

▶ **Definition 28** (Easy sequence). *A sequence $(a_n)_n$ of integers is said to be "easy to compute" if there exists a polynomial p such that straight line complexity of a_n, i.e. $\tau(a_n) \le p(\log n)$, for $n \ge 1$.*

If a sequence is not easy to compute, it is said to be hard. In fact, for most numbers N, one can show that $\tau(N) \ge \log N/\log\log N$ ("close" to the trivial upper bound) [14, 35]. It is believed that $(d!)$ is hard to compute. In fact, its hardness is deeply connected to the infamous integer factoring problem. [42] showed that $d!$ being easy to compute will imply factoring is easy in the non-uniform setting [4].

[4] However, this result does not imply that natural numbers can be factored in polynomial time in the Turing-Machine model, as the numbers used can be poly(n)-bits.

Constant-free circuit complexity. In the same spirit, one can define constant-free circuit complexity of polynomials where the given constants belong to the set $\{-1, 0, 1\}$[5]. We denote, $\tau(f)$ as the size of the minimal constant-free circuit computing f. Trivially, $L(f) \leq \tau(f)$.

It was shown in [3] that $(a_n)_{n \in \mathbb{N}}$, where $a_n := \binom{2n}{n}$, is easy implies $(n!)_{n \in \mathbb{N}}$ is easy. This proof is similar to [42]. This lemma will be crucial to prove the hardness result for truncations.

▶ **Lemma 29** (Lemma 6.3 in [3]). *If $a_n := \binom{2n}{n}$ has complexity $O(\log^c n)$, for some $c \in \mathbb{N}$, then $(n!)$ has complexity $O(\log^{c+1} n)$.*

In the following theorem, we show that constant-free complexity of the truncation of a power series with the denominator degree being *high,* is expected to be large, otherwise $n!$ is easy.

▶ **Theorem 30.** *If $\tau\left(\text{trunc}\left(1/(1+x)^{d+1}, m\right)\right) = O(\log^c d)$, for some constant $c \in \mathbb{N}$ and $m \in \{d-1, d\}$, then $(n!)$ is easy. In fact, $\tau(n!) = O(\log^{c+1} n)$.*

Proof. From the power series expansion (Section 2), it is easy to see that,

$$\text{trunc}\left(1/(1+x)^{d+1}, m\right) = \sum_{i=0}^{m} \binom{-d-1}{i} x^i .$$

Let us notice $\binom{-d-1}{i} = (-d-1)(-d-2)\ldots(-d-i)/i! = (-1)^i (d+i)!/i!\,d! = (-1)^i \binom{d+i}{i}$. Therefore,

$$\text{trunc}\left(1/(1+x)^{d+1}, d\right) - \text{trunc}\left(1/(1+x)^{d+1}, d-1\right) = (-1)^d \binom{2d}{d} x^d .$$

By assumption, $\tau\left((-1)^d \binom{2d}{d} x^d\right) = O(\log^c d)$. Therefore $\binom{2d}{d}$ has complexity $O(\log^c d)$, as desired (just by substituting $x = 1$, which gives an SLP). Invoking Lemma 29, we conclude. ◀

6 Hardness of Truncation of algebraic functions

In this section, we show conditional hardness of truncation of power series of algebraic functions with degree of its minpoly ≥ 3. In the first part, we show connection with integer factoring. In the second part, we show connection with computation of multiple of $(n!)$.

Throughout the section, we will be working with algebraic functions of the form $(1+k^2x)^{i/k}$, for $i, k \in \mathbb{N}$ with $i < k$. Here is a crucial claim. For a proof, we refer to Theorem 77.

▶ **Theorem 31.** *Fix $i, k \in \mathbb{N}$ with $i < k$. Then, $(1 + k^2x)^{i/k} \in \mathbb{Z}[[x]]$, i.e. it is an integral power series.*

6.1 Hardness of truncation of algebraic functions and integer factoring

Here, we show that if the truncation of each $(1+k^2x)^{i/k}$, for $i \in [k-1]$, has small constant-free circuit, then one can factor n in $\text{poly}(\log n)$ time, in the non-uniform setting. This would readily imply the first part of Theorem 8.

[5] To use 2^n in the circuit, one has to build up a circuit for 2^n, of size $\log n$, from 1; whereas in the usual sense of circuit size, constants are *free*. Thus, $f_d := 2^{2^d} x^d$ has $O(\log d)$-size circuit but *requires* $\Omega(d)$-size constant-free circuit.

▶ **Theorem 32.** *Let $k \in \mathbb{N}$. If $\tau(\mathrm{trunc}((1+k^2x)^{\frac{i}{k}}, d)) = O(\log^c d)$ (for some constant c) for all $i \in [k-1]$ then integer factorization (in the non-uniform setting) can be performed in polynomial time.*

Proof. Let, $(1+k^2x)^{\frac{i}{k}} = \sum_{d\geq 0} C_{d,i}\, x^d \in \mathbb{Z}[[x]]$, where the coefficient $C_{d,i}$ of x^d is equal to $\pm k^d(-i)\cdot(k-i)\cdot(2k-i)\cdots((d-1)k-i)/d!$. We see that the product of all $C_{d,i}$ is equal to:

$$\prod_{i\in[k-1]} C_{d,i} \;=\; \pm\frac{k^{(k-1)d}(k-1)!(dk)!}{(d!)^k k^d (kd-1)(kd-2)\cdots(kd-(k-1))}.$$

The assumption $\tau(\mathrm{trunc}((1+k^2x)^{\frac{i}{k}}, d)) = O(\log^c d)$ implies that $\tau(C_{d,i}) = O(\log^c d)$ (just by subtracting two consecutive truncations and substituting $x = 1$). This further implies that $\tau(\prod_{i\in[k-1]} C_{d,i}) = O(\log^c d)$, Let us define, for any $d \geq 1$,

$$N(d,k) \;:=\; \frac{k^{(k-2)d}(dk)!}{(d!)^k}\,.$$

We first argue that $N(d,k) \in \mathbb{N}$. This follows from the fact that $N(d,k) = \prod_{i\in[k-1]} C_{d,i}\cdot (kd-1)\cdots(kd-(k-1))/(k-1)!$, and $(k-1)!$ must divide $(kd-1)\cdots(kd-(k-1))$, by Fact 74.

Further, since k is constant, it implies that $\tau(N(d,k)) = O(\log^c d)$ (because the extra term has trivial $O(\log d)$-complexity).

Now, we describe how to find a non-trivial factor of a given integer n. We assume that all the primes dividing n are larger than k; otherwise we can remove all the prime factors smaller than $k+1$ (since k is a constant).

The idea is to first find a positive integer t such that all the primes dividing n are in the interval $[t+1, tk]$, by using an iterative algorithm; if such a t does not exist we would have already found a non-trivial factor of n (by the algorithm). As an *invariant*, we maintain an integer m such that all the prime divisors of n are greater than m. We start with $m = k$ and compute $\gcd(N(m,k), n)$ at each iteration. Since all the primes dividing n are greater than m (by assumption), we get that $\gcd(N(m,k), n) = \gcd((mk)!, n)$. If the $\gcd((mk)!, n) \neq 1, n$, we must have already found a non-trivial factor of n and we are done. Otherwise, we can have two cases: either (i) $\gcd((mk)!, n) = 1$, or (ii) $\gcd((mk)!, n) = n$.

If $\gcd((mk)!, n) = 1$ then we set $m \leftarrow mk$ and continue (because in this case all the primes dividing n must be greater than mk). Otherwise we have $\gcd((mk)!, n) = n$, and hence, all the primes dividing n are in the interval $[m+1, mk]$ and we stop with $t \leftarrow m$. We know that $t \leq \lceil n/k \rceil$ and this uses at most $\log_k n = \log n$ iterations. So, this step has given us an integer t such that all the primes dividing n are in the interval $[t+1, tk]$, and the time taken is $\mathrm{poly}(\log n)$, due to only $\log n$ many iterations and each step takes $\mathrm{poly}(\log n)$-time due to the fact that $\tau(N(d,k)) = O(\log^c d)$ implies gcd computation can be done in $\mathrm{poly}(\log n)$ (by euclidean algorithm).

Once, we know that all the primes are in an interval of the form $[t+1, tk]$, we now try to reduce the length of it to k so that, we can simply *brute force* to get a factor of n, otherwise of course our algorithm would already find a factor. The length reduction part is similar to binary search algorithm that we describe below.

To find a positive integer s such that all the primes dividing n are in the interval $[sk+1, (s+1)k]$ (Or we find a non-trivial factor of n), again we use an iterative algorithm. As an invariant, we maintain two positive integers a, b such that all the prime divisors of n are in the interval $[ak+1, bk]$. We start with $a = 1, b = t$. Our invariant is trivially true at the start.

At each iteration, we set $c = \lceil (a+b)/2 \rceil$ and compute $\gcd(N(c,k), n)$. Since $c \leq t$ and all the prime divisors of n are larger than t, we get that $\gcd(N(c,k), n) = \gcd((ck)!, n)$. Again, we argue in the same way as before. If the gcd is $\neq 1, n$, we have already found a non-trivial factor of n and we are done. Otherwise, we have two cases: either (i) $\gcd((ck)!, n) = 1$, or (ii) $\gcd((ck)!, n) = n$.

If $\gcd((ck)!, n) = 1$ then it is clear that all the primes dividing n are in the interval $[ck+1, bk]$ and hence we set $a \leftarrow c, b \leftarrow b$. If $\gcd((ck)!, n) = n$ then they all the primes dividing n are in the interval $[ak+1, ck]$ and hence we set $a \leftarrow a, b \leftarrow c$. This will terminate when $b - a \leq 1$. Hence we find the desired positive integer s. This uses at most $\log t = \log n$ iterations.

Now we just need to search for the prime divisors of n in the interval $[sk+1, (s+1)k]$ (an interval of constant length). Now, we brute force to finally find a non-trivial factor of n.

Similarly, this step also takes $\text{poly}(\log n)$ as each gcd computation takes $\text{poly}(\log n)$ time. So, we have successfully found a non-trivial factor of n by the end of this process, repeating this, we can get all the factors in $\text{poly}(\log n)$-time and we are done. ◀

We also refer to Algorithm 1 in Appendix I.

6.2 Hardness of truncation of algebraic functions and complexity of multiple of $(n!)$

In this section, we show that easiness of truncation of $(1+k^2x)^{i/k}$ shows that a multiple of $n!$ must be easy. Note that, this may not imply that $n!$ is easy, however, from complexity-theoretic point-of-view, it is believed to be hard because of non-trivial implications. Shub & Smale [44] proved: *If $n!$ is ultimately hard to compute, then* $\mathsf{P} \neq \mathsf{NP}$ *over the field of complex numbers..* Here, the computation is over Blum-Shub-Smale (BSS) model and can use complex numbers in the algorithm. In fact, a *stronger* version (known as τ-conjecture) connects $z(f)$, distinct integer roots of f with $\tau(f)$. Recently, [16] showed that a similar conjecture, in the SOS-model, would in fact imply explicit constructions of rigid matrices & $\mathsf{VP} \neq \mathsf{VNP}$. For similar related works, we refer to [25, 27].

Before discussing and stating the formal result, we need an important notion of complexity, which is closely related to τ-complexity.

▶ **Definition 33** (Ultimately easy). *A sequence of integers (a_n) is* ultimately easy *if there exists another sequence (b_n) such that $\tau(a_n\, b_n) \leq \text{poly}(\log n)$ for all large enough n.*

▶ **Definition 34** (Ultimate complexity). *Define the* ultimate complexity *of an integer n as the minimum τ-complexity of its multiple, i.e. $\tau_1(n) = \min_{b \in \mathbb{Z}\setminus\{0\}} \tau(b \cdot n)$.*

It is clear that Definition 33 can be stated wrt τ_1. We remark that $\tau_1(n_1 \cdot n_2) \leq \tau_1(n_1) + \tau_1(n_2) + 1$, for any $n_1, n_2 \in \mathbb{Z}$.

Following the same spirit as above, we prove the second part of Theorem 8.

▶ **Theorem 35.** *Fix $k \in \mathbb{N}$. Suppose, for each $i \in [k-1]$, there exists some constant c such that $\tau(\text{trunc}\left((1+k^2 \cdot x)^{i/k}, d\right) = O(\log^c d)$, for large enough d. Then, $(n!)_{n\in\mathbb{N}}$ is ultimately easy.*

Proof. Let, $(1+k^2x)^{\frac{i}{k}} = \sum_{d\geq 0} C_{d,i}\, x^d \in \mathbb{Z}[[x]]$. From the hypothesis, it follows that there exists c such that $\tau(C_{d,i}) \leq \log^c d$, for each $i \in [k-1]$ (subtract two consecutive terms and substitute $x = 1$). Further, from the proof in Section 6.1 (and following the same notation), we know that

$$\prod_{i\in[k-1]} C_{d,i} \;=\; \pm\frac{k^{(k-2)d}(k-1)!(dk)!}{(d!)^k(kd-1)(kd-2)\cdots(kd-(k-1))}\,.$$

Let us define, $a(d,k) := (dk)!/(d!)^k$. Note that, $a(d,k) \in \mathbb{N}$ (it is the multinomial coefficient $\binom{dk}{d,\ldots,d}$). Further, $k^{(k-2)d}\cdot a(d,k) = \prod_{i\in[k-1]} C_{d,i}\cdot(kd-1)\cdots(kd-(k-1))/(k-1)!$, and $(k-1)!$ must divide $(kd-1)\cdots(kd-(k-1))$, by Fact 74. As k is constant, each $kd-i$ can be computed in $O(\log d)$-time trivially. Further, $\tau(\prod_{i\in[k-1]} C_{d,i}) \le O(\log^c d)$. As τ is additive over multiplication, it follows that

$$\tau(k^{(k-2)d}\cdot a(d,k)) \;\le\; O(\log^c d) \;\Rightarrow\; \tau_1(a(d,k)) \;\le\; O(\log^c d)\,.$$

Now we recurse by noticing the following trivial identity that $n! \;=\; n!/\left(\lfloor n/k\rfloor)!\right)^k\cdot\left((\lfloor n/k\rfloor)!\right)^k$.

We know by the above relation on $a(d,k)$ (and replacing $d := \lfloor n/k\rfloor$ for some integer n) that

$$\tau_1\left(\frac{(k\cdot\lfloor n/k\rfloor)!}{(\lfloor n/k\rfloor)!^k}\right) \;\le\; O(\log^c n)\,.$$

Further, any integer n can be written as $n \;=\; k\cdot\lfloor n/k\rfloor + j$ for some $j \le k-1$. Note that $k\cdot\lfloor n/k\rfloor + j$ has complexity at most $\log n$ for each $j \in [k-1]$. So, multiplying $k\cdot\lfloor n/k\rfloor + j$ for $j\in[k-1]$, it is straightforward to deduce that

$$\tau_1\left(\frac{n!}{\left(\lfloor n/k\rfloor!\right)^k}\right) \;\le\; O(\log^c n)\,. \tag{2}$$

As, $n! \;=\; n!/\left(\lfloor n/k\rfloor)!\right)^k\cdot\left((\lfloor n/k\rfloor)!\right)^k$, and $\tau_1\left(\left(\lfloor n/k\rfloor!\right)^k\right) \le \tau_1\left(\lfloor n/k\rfloor!\right) + O(1)$; use Equation (2):

$$\begin{aligned}\tau_1(n!) \;&\le \tau_1\left(\lfloor n/k\rfloor!\right) \;+\; O(\log^c n) \;+\; O(1)\\ &\le \tau_1\left(\lfloor n/k^2\rfloor!\right) \;+\; O(\log^c n) \;+ O(\log^c n) \;+\; O(1)\\ &\;\vdots\\ &\le \log_k n\cdot O(\log^c n) = O(\log^{c+1} n)\,.\end{aligned}$$

Therefore, $(n!)$ is ultimately easy to compute, as we wanted. ◀

7 Complexity of the truncation of transcendental power series

In this section, we show examples where the truncation of transcendental power series is easy. We also complement this by showing the existence of *integral* transcendental power series which is conditionally hard.

7.1 The truncation of transcendental power series can be easy

In this section, we show two examples of integral transcendental power series whose truncations are easy.

7.1.1 Transcendental series corresponding to the Stern Sequence is easy

▶ **Definition 36** (The Stern sequence). *The sequence $(a_n)_{n\geq 0}$ given by $a_0 = 0, a_1 = 1$, and when $n \geq 1$, by $a_{2n} = a_n$ and $a_{2n+1} = a_n + a_{n+1}$, is called the Stern sequence.*

The generating function $A(x) \overset{\text{def}}{=\!=} \sum a_n x^n$ of the Stern sequence has the following properties.

▶ **Theorem 37** (Lemma 2.1 and Theorem 2.2 in [12]). *If $A(z)$ is the generating function of the Stern sequence, then*

1. $A(x^2) = A(x)\left(\frac{x}{x^2+x+1}\right)$.
2. *The function $A(x)$ is transcendental.*

Now we prove the following Theorem 38 which shows that its truncation has small circuit.

▶ **Theorem 38.** *For the generating function $A(x)$ of the Stern sequence, we have*

$$L\left(\operatorname{trunc}\left(A(x), d\right)\right) \;=\; O(\log^2 d)\,.$$

Proof. By using Theorem 37, we obtain that:

$$A(x) = (x^2+1)A(x^2) + \frac{A(x^2)}{x}. \tag{3}$$

Suppose $B_d(x) \overset{\text{def}}{=\!=} \operatorname{trunc}\left(A(x), \lfloor \frac{d}{2} \rfloor + 1\right)$. Notice that the degree of $C_d(x) \overset{\text{def}}{=\!=} (x^2 + 1)B_d(x^2) + B_d(x^2)/x$ is at most $2\lfloor d/2 \rfloor + 4$ and $\operatorname{trunc}(C_d(x), d) = \operatorname{trunc}\left(A(x), d\right)$. Hence we can compute $\operatorname{trunc}\left(A(x), d\right)$ from $C_d(x)$ by subtracting at most 4 monomials, which can be done using $O(\log d)$ gates. Also $B_d(x)$ can be computed from $\operatorname{trunc}\left(A(x), \lfloor d/2 \rfloor\right)$ using $O(\log d)$ gates. Hence we obtain the following recurrence:

$$L\left(\operatorname{trunc}\left(A(x), d\right)\right) \;\leq\; L\left(\operatorname{trunc}\left(A(x), \lfloor d/2 \rfloor\right)\right) \;+\; O(\log d)\,.$$

This implies, $L\left(\operatorname{trunc}\left(A(x), d\right)\right) = O(\log^2 d)$. ◀

7.1.2 Transcendental power series whose coefficients are multiplicative

The sequence $(f_n)_{n\geq 0}$ is defined as: $f_0 = 1, f_1 = 1, f_2 = -1$, $f_p = 1$ for all odd primes p and $f_{ab} = f_a f_b$. We look at the corresponding generating function $F(x) \overset{\text{def}}{=\!=} \sum f_n x^n$.

▶ **Theorem 39** ([13, Theorem 2]). *The power series $F(x)$ is transcendental.*

Now we prove the following Theorem 40 which shows that truncation of $F(x)$ is easy.

▶ **Theorem 40.** *For $F(x)$, we have $L\left(\operatorname{trunc}\left(F(x), d\right)\right) = O(\log^2 d)$.*

Proof. We use the notation $\nu_2(m)$ to denote the highest power of 2 which divides $m \in \mathbb{N}$. We partition the set $[d]$ into $\lfloor \log d \rfloor$ sets $S_0, S_1, S_2, \ldots, S_{\lfloor \log d \rfloor}$ such that $k \in S_i$ iff $\nu_2(k) = i$. We define the set $O_m \overset{\text{def}}{=\!=} \{k \mid k \leq m \text{ and } k \text{ is odd}\}$. Now, notice that $S_i = \{2^i k \mid k \in O_{\lfloor d/2^i \rfloor}\}$. For a set $S \in \mathbb{N}$, we define the polynomial $g_S \overset{\text{def}}{=\!=} \sum_{i\in S} x^i$. Observe that:

$$\operatorname{trunc}\left(F(x), d\right) \;=\; 1 \;+\; \sum_{i=1}^{\lfloor \log d \rfloor} (-1)^i g_{S_i}\,.$$

Trivially, $g_{S_i} \;=\; g_{O_{\lfloor d/2^i \rfloor}}(x^{2^i})$. Also notice that $g_{O_m} \;=\; g_{[m]} - g_{\lfloor \frac{m}{2} \rfloor}(x^2)$. Therefore, $L(g_{O_m}) = (\log m)$, which implies that $g_{S_i} = O(\log d)$. Hence, $L\left(\operatorname{trunc}\left(F(x), d\right)\right) \;=\; O(\log^2 d)$. ◀

▶ Remark 41. Note that, there are power series like $\sum_{i\geq 0} x^{i!}$ which are transcendental and their truncations up to degree d are easy to compute. However, the series is highly *sparse* and degree-d truncations has only $\text{poly}(\log d)$ monomials, hence the easiness is trivial. The examples we discover in this work are of dense power series.

7.2 The truncation of Transcendental power series can be hard

A sequence $(h_n)_{n\geq 0}$ is called holonomic if it satisfies the recurrence of the form:

$$a_r(n)\,h_{n+r} \,+\, a_{r-1}(n)\,h_{n+r-1} \,+\, \cdots \,+\, a_0(n)h_n \;=\; 0\,,$$

where a_i are polynomials in n. The corresponding generating function, $H(x) \overset{\text{def}}{=\!=} \sum h_n x^n$, is said to be a *holonomic function*.

Consider the holonomic sequence $f_n = (n!)$ defined by $f_0 = 1$ and $f_{n+1} - (n+1)f_n = 0$. Also consider the corresponding generating function $F(x) = \sum_{n\geq 0} n!x^n$. We now show that $F(x)$ is transcendental and that truncation of $F(x)$ is (conditionally) hard to compute. To this end, we need the following Lemma 42, which follows directly from Proposition 2 in [28].

▶ **Lemma 42** ([28]). *If $F(x) = \sum_{n\geq 0} f_n x^n$ is a power series in $\mathbb{C}[[x]]$ and the radius of convergence of $F(x)$ is zero then $F(x)$ is transcendental.*

▶ **Corollary 43.** *The power series $F(x) = \sum_{n\geq 0} n!x^n$ is transcendental.*

Proof. It is clear that the radius of convergence of $F(x)$ is zero (follows from the ratio test). Hence Lemma 42 implies that $F(x)$ is transcendental. ◀

▶ **Theorem 44.** *If $\tau(\text{trunc}(F(x), d)) = \text{poly}(\log d)$ then $(d!)$ has complexity $\text{poly}(\log d)$.*

Proof. We know that $d!x^d = \text{trunc}(F(x), d) - \text{trunc}(F(x), d-1)$. Setting $x = 1$, we conclude. ◀

8 SOS-complexity of truncation

A univariate polynomial $f(x) \in \mathbb{F}[x]$ over a field $\mathbb{F}$ is computed as a *sum-of-squares* (SOS) if

$$f \;=\; \sum_{i=1}^{s} c_i f_i^2 \;, \tag{4}$$

for some *top-fanin* s, where $f_i(x) \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$.

▶ Remark 45. In real analysis, the SOS representation of a polynomial $f(x) \in \mathbb{R}[x]$, is defined where the coefficients $c_i > 0$ (in fact, we can take $c_i = 1$, by taking $\sqrt{c_i}$ inside f_i); thus the definition makes sense only for non-negative polynomials f. In this sense, (Equation (4)) is a *weighted* SOS. However, we will skip the term "weighted" (also because $\mathbb{F}$ can be $= \mathbb{C}$ here).

▶ **Definition 46** (Support-sum size $S_{\mathbb{F}}(f)$, [18]). *The* size *of the representation of f in Equation* (4) *is the* support-sum, *the sum of the support size (or sparsity) of the polynomials f_i. The* support-sum size *of f, denoted by $S_{\mathbb{F}}(f)$, is defined as the minimum support-sum of f.*

We will often refer to $S_{\mathbb{F}}(f)$ as the SOS-complexity of f. Note that, it is *sub-additive*, i.e. for two polynomials $f, g \in \mathbb{F}[x]$, we have $S_{\mathbb{F}}(f + g) \leq S_{\mathbb{F}}(f) + S_{\mathbb{F}}(g)$.

Let $|f|_0$ denote the sparsity of f. For any field $\mathbb{F}$ of characteristic $\neq 2$, we have $|f|_0^{1/2} \leq S_{\mathbb{F}}(f) \leq\; 2\,|f|_0 + 2$. The lower bound can be shown by counting monomials. The upper bound is because $f \;=\; (f+1)^2/4 \,-\, (f-1)^2/4$. In particular, the SOS-model is *complete* when $\text{char}(\mathbb{F}) \neq 2$. We will drop the subscript $\mathbb{F}$ when it is clear or unnecessary in the context.

▶ **Definition 47** (SOS-hardness, [18]). *An "explicit" univariate $(f_d(x))_d$, where f_d is of degree d in $\mathbb{F}[x]$, is SOS-hard if $S(f_d) = \omega(d^{1/2})$.*

▶ Remark 48. If $S(f_d) = O(d^{1/2})$, we call (f_d) *SOS-easy*. Eg. $f_d = \sum_{i=0}^{d} x^i$ is SOS-easy (Lemma 67).

It was shown in [18] that an SOS-hard family, with $S(f_d) \geq d^{1/2+\epsilon}$, for $\epsilon = \omega\left(\sqrt{\frac{\log\log d}{\log d}}\right)$, implies VP $\neq$ VNP. We want to characterize the SOS-easy and SOS-hard families, via natural operations like division and truncation. Towards that, we show the following Theorem 49. We assume $\mathbb{F} = \overline{\mathbb{F}}$ (otherwise we can go to small extensions).

▶ **Theorem 49** (Truncation is SOS-easy). *Let $g, h \in \mathbb{F}[x]$ are both constant-degree polynomials s.t. $g/h \in \mathbb{F}[[x]]$. Then, truncation of g/h upto degree-d is SOS-easy,i.e. $S(\text{trunc}(g/h, d)) = O(d^{1/2})$.*

Before proving this, we need a few important lemmas.

▶ **Lemma 50.** *Let $f \in \mathbb{F}[x]$. Then, $S(f^{(k)}) \leq O(k\, S(f))$.*

Proof. Let $f = \sum_{i=1}^{s} c_i f_i^2$ be the *minimal* SOS representation with $|f_i|_0 = t_i$, i.e. $\sum_{i\in[s]} t_i = S(f)$. Trivially, $f^{(k)} = \sum_{i\in[s]} f_i^{2^{(k)}}$. Using the Leibniz rule (Lemma 66), we have

$$f_i^{2^{(k)}} = \begin{cases} 2\displaystyle\sum_{j=0}^{\frac{k}{2}-1} \binom{k}{j} \cdot f_i^{(j)} \cdot f_i^{(k-j)} + \binom{k}{k/2}\left(f_i^{(k/2)}\right)^2 & \text{if}\, k \equiv 0 \bmod 2 \\ 2\displaystyle\sum_{j=0}^{\frac{k-1}{2}} \binom{k}{j} \cdot f_i^{(j)} \cdot f_i^{(k-j)} & \text{if}\, k \equiv 1 \bmod 2 \end{cases}$$

Write each $f_i^{(j)} \cdot f_i^{(k-j)}$ as

$$f_i^{(j)} \cdot f_i^{(k-j)} = 1/4 \cdot (f_i^{(j)} + f_i^{(k-j)})^2 - 1/4 \cdot (f_i^{(j)} - f_i^{(k-j)})^2 .$$

Note that, $|f_i^{(j)}|_0 \leq t_i$, for each $i \in [s]$ and $j \in [0, k]$. Thus, $f_i^{2(k)}$ has a representation with support-sum at most $\lceil \frac{k+1}{2} \rceil \cdot 4 \cdot t_i \leq O(k\, t_i)$. Applying this to each $i \in [s]$ shows that $f^{(k)}$ has a SOS representation with support-sum at most $O\left(k \cdot \sum_i t_i\right) = O(k\, S(f))$; and the conclusion follows. ◀

▶ **Lemma 51.** *$S\left(\text{trunc}\left(1/(x-a)^j, d\right)\right) \leq O\left(j \cdot \sqrt{d+j}\right)$, for any $j \in \mathbb{Z}_{\geq 0}$.*

Proof. Let $g_d(x) := \text{trunc}(1/x - a, d) = -1/a \cdot \left(\sum_{i=0}^{d} (x/a)^i\right)$. By differentiation, it follows that $(1/(x-a))^{(j-1)} = (-1)^{j-1} \cdot (j-1)! \cdot \left(1/(x-a)^j\right)$. Thus, one can conclude that

$$\text{trunc}\left(1/(x-a)^j, d\right) = (-1)^{j-1}/(j-1)! \cdot g_{d+j-1}^{(j-1)}(x) .$$

Note that, $S_{\mathbb{F}}(g_{d+j-1}(x)) = O\left(\sqrt{d+j-1}\right)$ (Lemma 67). Using Lemma 50, the conclusion follows. ◀

Now, we are well-equipped to prove Theorem 49.

Proof of Theorem 49. This proof is very similar to that of Theorem 26. Let m be the highest power of x such that $x^m \mid h$ (i.e. $x^{m+1} \nmid h$). Note that, as $g/h \in \mathbb{F}[[x]]$, $x^m \mid g$ as well (Lemma 65). Suppose, $\deg(h) =: d_h$. Thus $m \le d_h$. As d_h is a constant, so is m. Note that, $g_1 := g/x^m$ and $h_1 := h/x^m$ are both constant degree polynomials.

By definition, $g/h = g_1/h_1$. Let $g_2 := g_1 \bmod h_1$. Hence, $g_1/h_1 = g_1 \text{ div } h_1 + g_2/h_1$ and $\deg(g_2) < \deg(h_1)$. Finally, $\text{trunc}(g_1/h_1, d) = g_1 \text{ div } h_1 + \text{trunc}(g_2/h_1, d)$. However, $S(g_1 \text{ div } h_1) = O(1)$, as it has constant degree. Thus, it suffices to bound $S(\text{trunc}(g_2/h_1, d))$.

Suppose, h_1 factors over $\mathbb{F}[x]$, as $h_1 := \prod_{i\in[k]} (x - a_i)^{d_i}$. Moreover, using Lemma 24, we know that there are constants $a_i, b_{ij} \in \mathbb{F}$ such that

$$g_2(x)/h_1(x) = \sum_{i\in[k]} \sum_{j\in[d_i]} b_{ij}/(x-a_i)^j \,.$$

Therefore,

$$\text{trunc}(g_2/h_1, d) = \sum_{i\in[k]} \sum_{j\in[d_i]} b_{ij} \cdot \text{trunc}\left(1/(x-a_i)^j, d\right) \,.$$

Note that, d_i and k are constants. Using Lemma 51 and sub-additivity property of S, the conclusion follows. ◀

▶ Remark 52.

1. It is unclear how to extend this proof to non-constant degree polynomials g and h.
2. It is unclear whether $S(g/h)$ is small, when $h \mid g$ and $S(g)$ is small and $\deg(h)$ is small.

9 Constant-free complexity of $\bmod x^d$ and PosSLP

In this section, we investigate constant-free complexity of computing $\bmod x^d$ and its intrinsic connection with the positivity questions (i.e. PosSLP, for definition, see Problem 56).

▶ **Problem 53** (Modular complexity). If we have $L(f) = s$ for some $f \in \mathbb{C}[x]$, what is complexity of $f \bmod x^d$?

We prove a conditional lower bounds on the constant-free complexity of $f \mod x^d$.

▶ **Theorem 54.** *If* $\tau(f) = s$ implies $\tau(f \mod x^d) = \text{poly}(s, \log d)$ *for all* $f \in \mathbb{Z}[x]$ *then* $\binom{2n}{n}_{n\in\mathbb{N}}$ *has complexity* $\text{poly}(\log n)$.

Proof. Suppose $m = 2^{\lceil \log d \rceil}$. Consider $\sqrt{1+4x}$, by Lemma 71, we know that $\sqrt{1+4x} \in \mathbb{Z}[[x]]$. By using Newton's iteration, we can compute a polynomial $g \in \mathbb{Z}[x]$ such that $g \bmod x^m = \sqrt{1+4x} \mod x^m$ and $\tau(g) = O(m) = (\log d)$ (Using Newton's iteration, see Theorem 6.5 in [21], also [29]). Now $g \mod x^d = \text{trunc}(\sqrt{1+4x}, d)$. Our assumption implies that $L(\text{trunc}(\sqrt{1+4x}, d) = \text{poly}(\log d)$. By a similar argument as in the proof of Theorem 72, we get that $\binom{2n}{n}_{n\in\mathbb{N}}$ has complexity $\text{poly}(\log n)$.

An alternative proof: we know $\tau((x+1)^{2n}) = O(\log n)$. Now see that $((x+1)^{2n}) \bmod x^{n+1} - ((x+1)^{2n}) \bmod x^n = x^n\binom{2n}{n}$. Therefore the assumption in the statement of the theorem implies that $\binom{2n}{n}_{n\in\mathbb{N}}$ has complexity $\text{poly}(\log n)$. ◀

Theorem 54 demonstrates that computing remainders $\bmod x^d$ should be hard. Now we pose the following simpler problem.

▶ **Problem 55** (Special divisibility question). If we have $\tau(f) = s$ for some $f \in \mathbb{C} = \mathbb{Z}[x]$, what is complexity of deciding if $f \mod x^d = 0$, *i.e.*, decide if x^d divides f? Here the input is a circuit C of size s which computes f.

It turns out that the question essentially reduces to decide the positivity of a number, computed by an SLP (Theorem 60).

▶ **Problem 56** (PosSLP [2])**.** Given an SLP P (without divisions), decide if the integer computed by P is positive?

▶ Remark 57. [2] proved that that the Generic Task of Numerical Computation is polynomial-time equivalent to PosSLP and also showed that PosSLP lies in the counting hierarchy CH.

▶ **Proposition 58** (Folklore)**.** *Given an an SLP P (with divisions) of length n computing a rational number $\frac{p}{q}$, there exist a division free SLP $Q = (q_0, q_1, \ldots, q_{6n})$ such that $q_{6n-1} = p$ and $q_{6n} = q$.*

Proof. Suppose $P = (a_0, a_1, \ldots, a_n)$. We split every gate a_i in P to two gates b_i and c_i such that $a_i = \frac{b_i}{c_i}$. Now notice that:

$$\frac{b_1}{c_1} + \frac{b_2}{c_2} = \frac{b_1c_2 + b_2c_1}{c_1c_2}.$$
$$\frac{b_1}{c_1} \cdot \frac{b_2}{c_2} = \frac{b_1b_2}{c_1c_2}.$$

This implies the claimed SLP Q. ◀

▶ **Lemma 59.** *Given two SLP P_1, P_2 (with divisions) of length n computing the rational numbers $\frac{a}{b}$ and $\frac{p}{q}$ respectively, problem of deciding $\left|\frac{a}{b}\right| > \left|\frac{p}{q}\right|$ is in $\mathsf{P}^{\mathrm{PosSLP}}$.*

Proof. By using Proposition 58, we first obtain SLPs $Q = (q_0, q_1, \ldots, q_{6n})$ and $R = (r_0, r_1, \ldots, r_{6n})$ such that $q_{6n-1} = a, q_{6n} = b$ and $r_{6n-1} = p, r_{6n} = q$. Using the PosSLP oracle, we find the signs of $\frac{a}{b}$ and $\frac{p}{q}$. After finding the signs, we can find SLPs (of length $6n+1$) which compute $|a|, |b|, |p|, |q|$. This implies an SLP of length $24n+7$ which computes $|a|\,|q| - |p|\,|b|$. And deciding $|a|\,|q| - |p|\,|b| > 0$ also decides $\left|\frac{a}{b}\right| > \left|\frac{p}{q}\right|$. ◀

▶ **Theorem 60.** *Problem 55 is in $\mathsf{P}^{\mathrm{PosSLP}}$.*

Proof. We are given a constant free circuit C of size s which computes f. It is easy to see that $\deg(f) \leq 2^s$. We define $\|f\|_\infty$ to be the largest absolute value of coefficients of f. By induction, it is easy to see that $\|f\|_\infty \leq 2^{2^{2s}}$. Let M be any positive integer such that $M > 4 \cdot 2^s \cdot \|f\|_\infty$. Now we claim:

$$x^d \mid f \iff \left|f\left(\frac{1}{M}\right)\right| < \frac{1}{4M^{d-1}}.$$

Suppose $x^d \mid f$. Then we have $f = f_d x^d + f_{d+1}x^{d+1} + \cdots + f_n x^n$. In this case:

$$f\left(\frac{1}{M}\right) = \frac{1}{M^{d-1}}\left(\frac{f_d}{M} + \frac{f_{d+1}}{M^2} + \cdots + \frac{f_i}{M^{i-d+1}} + \cdots + \frac{f_n}{M^{n-d+1}}\right). \tag{5}$$

In Equation (5), the absolute value of each term $\frac{f_i}{M^{i-d+1}}$ is less than $\frac{1}{4\cdot 2^s}$. Therefore $\left|f\left(\frac{1}{M}\right)\right| < \frac{1}{4M^{d-1}}$.

Now consider the case when $x^d \nmid g$. Let $m < d$ be the least positive integer such that x^m has non-zero coefficient in f. So $f = f_m x^m + g$ with $f_m \neq 0$ and $g = f_{m+1}x^{m+1} + \cdots + f_n x^n$. By using the argument above, we obtain $\left|g\left(\frac{1}{M}\right)\right| < \frac{1}{4M^m}$. Also, $|f_m x^m| \geq \frac{1}{M^m}$. Therefore $\left|f\left(\frac{1}{M}\right)\right| > \frac{3}{4}\frac{1}{M^m} \geq \frac{3}{4}\frac{1}{M^{d-1}} > \frac{1}{4M^{d-1}}$. Hence our claim is true.

Now notice that M has straight complexity at most $3s$. Therefore$f\left(\frac{1}{M}\right)$ has straight complexity (with divisions) at most $4s+1$. Also, $\frac{1}{4M^{d-1}}$ has straight complexity (with divisions) at most $3s+2+2\log d$. Therefore, by using Lemma 59 we can check $\left|f\left(\frac{1}{M}\right)\right| < \frac{1}{4M^{d-1}}$ in $\mathsf{P}^{\text{PosSLP}}$. Therefore Problem 55 is in $\mathsf{P}^{\text{PosSLP}}$. ◀

Theorem 60 and Remark 57 imply that Problem 55 lies in the counting hierarchy CH.

10 Conclusion

Our result on division elimination can be seen as evidence towards the possibility of a positive solution of Problem 1. Though the current techniques may not solve Problem 1, it is interesting to know division elimination (in circuits) is possible without using power series.

It is known that the decision problem of divisibility testing in the high degree regime: whether g (of size s and degree $\exp(s)$) is divisible by a polynomial h (of size s and degree $\exp(s)$) is NP-hard, even when h is a supersparse polynomial [40]. However, its NP-hardness does not rule out the possibility of positive solution of Problem 1.

There are several avenues for extending our study of truncations of power series. Here, we remark that, Theorem 8 implies that, for any prime p, there is a simple algebraic function with degree of its minpoly $= p$, such that the truncation is conditionally hard. But it is not clear whether it is true for composite (because i/k can reduce, when $k \neq p$).

One can also investigate truncation of algebraic power series over characteristic p. [6] showed that n-th coefficient of an algebraic power series over characteristic p can be computed in $O(\log n, p)$-time. One can study truncations of power series with $0-1$ coefficients and relate their hardness with classical assumptions in complexity, eg. truncated Θ-functions [37].

Here are some immediate questions of interest which require rigorous investigation.

1. Can we remove the degree condition on g in Theorem 6?
2. Does Theorem 6 hold in the border sense? Note that, the degree of the approximate circuit can have degree $> d$ and thus homogenization seems necessarily blowing the complexity in d.
3. Can we show that the truncation of *any* "simple" algebraic function (satisfying a minpoly of degree > 2 with bounded coefficients) must be conditionally hard in Theorem 8? In particular, can we show that $(1+9x)^{1/3}$ is conditionally hard?
4. Does Theorem 4 hold in the SOS-complexity regime?

References

1 Alexander Alder. *Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht.* PhD thesis, Zentralstelle der Studentenschaft, 1984.

2 Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Miltersen. On the Complexity of Numerical Analysis. *SIAM Journal on Computing*, 38, January 2006. Preliminary version in the 21^{st} Annual IEEE Conference on Computational Complexity (CCC'06). doi:10.1109/CCC.2006.30.

3 Robert Andrews. Algebraic Hardness Versus Randomness in Low Characteristic. In *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:32. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.37.

4 Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages.* Springer-Verlag, Berlin, Heidelberg, 1988. URL: https://dl.acm.org/doi/book/10.5555/52107.

5 Markus Bläser and Gorav Jindal. On the Complexity of Symmetric Polynomials. In 10^{th} *Innovations in Theoretical Computer Science Conference (ITCS'19)*, volume 124 of *LIPIcs*, pages 47:1–47:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.47.

6 Alin Bostan, Gilles Christol, and Philippe Dumas. Fast computation of the Nth term of an algebraic series over a finite prime field. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation (ISSAC'16)*, pages 119–126, 2016. doi:10.1145/2930889.2930904.

7 Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On Algebraic Branching Programs of Small Width. *J. ACM*, 65(5):1–29, 2018. (Preliminary version in the 32^{nd} Computational Complexity Conference (CCC'17). doi:10.1145/3209663.

8 Peter Bürgisser. The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics*, 4(4):369–396, 2004. arXiv:1812.06828.

9 Peter Bürgisser. On defining integers and proving arithmetic circuit lower bounds. *Computational Complexity*, 18(1):81–103, 2009. doi:10.1007/s00037-009-0260-x.

10 Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.

11 David V Chudnovsky and Gregory V Chudnovsky. On expansion of algebraic functions in power and Puiseux series, I. *Journal of Complexity*, 2(4):271–294, 1986. URL: https://www.sciencedirect.com/science/article/pii/0885064X86900063.

12 Michael Coons. The Transcendence of Series Related to Stern's Diatomic Sequence. *International Journal of Number Theory*, 06, November 2011. doi:10.1142/S1793042110002958.

13 Michael Coons and Peter Borwein. Transcendence of power series for some number theoretic functions. *Proceedings of the American Mathematical Society*, 137, July 2008. doi:10.1090/S0002-9939-08-09737-2.

14 Wellington De Melo and Benar Fux Svaiter. The cost of computing integers. *Proceedings-American Mathematical Society*, 124:1377–1378, 1996. URL: https://www.ams.org/journals/proc/1996-124-05/S0002-9939-96-03173-5/S0002-9939-96-03173-5.pdf.

15 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978. URL: https://www.sciencedirect.com/science/article/abs/pii/0020019078900674.

16 Pranjal Dutta. Real tau-Conjecture for sum-of-squares: A unified approach to lower bound and derandomization. In *16th International Computer Science Symposium in Russia (CSR 2021)*, 2021. URL: https://drive.google.com/file/d/1X8eo9GM4SCNsC2vWjPbUwMX0vff5i2k3/view.

17 Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1152–1165, 2018. URL: https://www.cse.iitk.ac.in/users/nitin/papers/factor-closure.pdf.

18 Pranjal Dutta, Nitin Saxena, and Thomas Thierauf. A Largish Sum-Of-Squares Implies Circuit Hardness and Derandomization. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021. URL: 10.4230/LIPIcs.ITCS.2021.23.

19 Joshua A Grochow et al. Complexity in ideals of polynomials: questions on algebraic complexity of circuits and proofs. *Bulletin of EATCS*, 2(130), 2020. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/607.

20 Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. Boundaries of VP and VNP. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55, pages 34:1–34:14, 2016. URL: https://core.ac.uk/download/pdf/62922137.pdf.

21 Gorav Jindal. *On approximate polynomial identity testing and real root finding*. PhD thesis, Saarland University, 2019. doi:10.22028/D291-29880.

22 Erich Kaltofen. Uniform closure properties of p-computable functions. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 330–337, 1986. doi:10.1145/12130.12163.

23 Erich Kaltofen. Single-factor Hensel lifting and its application to the straight-line complexity of certain polynomials. In *Proceedings of the 19^{th} annual ACM symposium on Theory of computing (STOC'87)*, pages 443–452, 1987. doi:10.1145/28395.28443.

24 Pascal Koiran. Valiant's model and the cost of computing integers. *computational complexity*, 13(3):131–146, 2005.

25 Pascal Koiran. Shallow circuits with high-powered inputs. *Innovations in Computer Science (ICS)*, 2011. URL: https://hal-ens-lyon.archives-ouvertes.fr/ensl-00477023v4/document.

26 Pascal Koiran and Sylvain Perifel. Interpolation in Valiant's theory. *Computational Complexity*, 20(1):1–20, 2011. doi:10.1007/s00037-011-0002-8.

27 Pascal Koiran, Natacha Portier, Sébastien Tavenas, and Stéphan Thomassé. A τ-Conjecture for Newton Polygons. *Foundations of computational mathematics*, 15(1):185–197, 2015. doi:10.1007/s10208-014-9216-x.

28 FV Kuhlmann. On convergent power series, 1996. URL: https://www.mathi.uni-heidelberg.de/~roquette/KONVPOTREIHEN.pdf.

29 Hsiang Kung and Joseph Traub. All Algebraic Functions Can Be Computed Fast. *J. ACM*, 25:245–260, April 1978. doi:10.1145/322063.322068.

30 Dick Lipton and Ken Regan. Factoring and factorials, February 2009. URL: https://rjlipton.wordpress.com/2009/02/23/factoring-and-factorials/.

31 Richard J Lipton. Polynomials with 0-1 coefficients that are hard to evaluate. *SIAM Journal on Computing*, 7(1):61–69, 1978. Preliminary version in the 16^{th} Annual Symposium on Foundations of Computer Science (FOCS 1975). URL: https://epubs.siam.org/doi/abs/10.1137/0207004?journalCode=smjcat.

32 Richard J Lipton. Straight-line complexity and integer factorization. In *International Algorithmic Number Theory Symposium (ANTS 94)*, pages 71–79. Springer, 1994. doi:10.1007/3-540-58691-1_45.

33 Richard J Lipton and Larry J Stockmeyer. Evaluation of polynomials with super-preconditioning. *Journal of Computer and System Sciences*, 16(2):124–139, 1978. URL: https://www.sciencedirect.com/science/article/pii/0022000078900417.

34 Meena Mahajan. Algebraic Complexity Classes. In *Perspectives in Computational Complexity*, pages 51–75. Springer, 2014. doi:10.1007/978-3-319-05446-9_4.

35 Carlos Moreira. On asymptotic estimates for arithmetic cost functions. *Proceedings of the American Mathematical Society*, 125(2):347–353, 1997. URL: https://www.jstor.org/stable/2161660.

36 Ketan D Mulmuley. The GCT program toward the P vs. NP problem. *Communications of the ACM*, 55(6):98–107, 2012. doi:10.1145/2184319.2184341.

37 Danny Nguyen and Igor Pak. Complexity of short generating functions. In *Forum of Mathematics, Sigma*, volume 6. Cambridge University Press, 2018. arXiv:1702.08660.

38 Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

39 Igor Pak. Complexity problems in enumerative combinatorics. In *Proceedings of the International Congress of Mathematicians – Rio de Janeiro 2018. Vol. IV. Invited lectures*, pages 3153–3180. World Sci. Publ., Hackensack, NJ, 2018. doi:10.1142/9789813272880_0176.

40 David A Plaisted. New NP-hard and NP-complete polynomial and integer divisibility problems. *Theoretical Computer Science*, 31(1-2):125–138, 1984. Preliminary in the 17^{th} Annual Symposium on Foundations of Computer Science (FOCS 1976). URL: https://www.sciencedirect.com/science/article/pii/0304397584901300.

41 Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

42 Adi Shamir. Factoring numbers in O (logn) arithmetic steps. *Information Processing Letters*, 8(1):28–31, 1979. URL: https://www.sciencedirect.com/science/article/abs/pii/0020019079900875.

43 Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010. doi:10.1561/0400000039.

44 Michael Shub and Steve Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "NP $\neq$ P?". *Duke Math. J.*, 81(1):47–54 (1996), 1995. A celebration of John F. Nash, Jr. doi:10.1215/S0012-7094-95-08105-8.

45 Volker Strassen. Vermeidung von divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973.

46 L Valiant. Reducibility by algebraic projections in: Logic and algorithmic. In *Symposium in honour of Ernst Specker*, pages 365–380, 1982.

47 Leslie G Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM symposium on Theory of computing*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

48 Joachim Von Zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

49 Wact. Some Accessible Open Problems. Workshop on Algebraic Complexity Theory (WACT 2016). URL: https://www.cs.tau.ac.il/~shpilka/wact2016/concreteOpenProblems/openprobs.pdf.

50 Klaus W Wagner. The complexity of combinatorial problems with succinct input representation. *Acta informatica*, 23(3):325–356, 1986. doi:10.1007/BF00289117.

51 Richard Zippel. Probabilistic Algorithms for Sparse Polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, 1979. doi:10.1007/3-540-09519-5_73.

52 J.von zur Gathen and V. Strassen. Some polynomials that are hard to compute. *Theoretical Computer Science*, 11(3):331–335, 1980. URL: http://www.sciencedirect.com/science/article/pii/0304397580900201.

A Basics in Arithmetic circuit complexity

An *arithmetic circuit* over a field $\mathbb{F}$ is a layered directed acyclic graph that uses field operations $\{+, \times\}$ and computes a polynomial. It can be thought of as an algebraic analog of Boolean circuits. The leaf nodes are labeled with the input variables $x_1, \ldots, x_n$ and constants from $\mathbb{F}$. Other nodes are labeled as addition and multiplication *gates*. The root node outputs the polynomial computed by the circuit. At times, we also use $\div$ gate in the circuit.

For a polynomial f, the size of the smallest circuit computing f is denoted by $L(f)$, it is the *arithmetic circuit complexity* of f. Here, size of an arithmetic circuit is assumed to be the number of nodes (variables included).

In *complexity classes*, we specify an upper bound on these parameters. Valiant's class VP contains the families of n-variate polynomials of degree poly(n) over $\mathbb{F}$, computed by circuits of poly(n)-size. The class VNP can be seen as a non-deterministic analog of the class VP. A family of n-variate polynomials $(f_n)_n$ over $\mathbb{F}$ is in VNP if there exists a family of polynomials $(g_n)_n$ in VP such that for every $\mathbf{x} = (x_1, \ldots, x_n)$ one can write $f_n(\mathbf{x}) = \sum_{w \in \{0,1\}^{t(n)}} g_n(\mathbf{x}, w)$, for some polynomial $t(n)$ which is called the *witness size*. It is straightforward to see that $\mathsf{VP} \subseteq \mathsf{VNP}$ and *conjectured* to be different (Valiant's Hypothesis [47]). *Equivalently,* symbolic permanent$_{n\times n}$ requires $n^{\omega(1)}$ size circuit.

One can define the class VP_0 (respectively, VNP_0) as the analogue of VP (respectively, VNP) in the constant-free regime. For more details see [24, 26, 34, 43, 10].

Coefficient-extraction in arithmetic circuits is easy using interpolation, see the folklore lemma below, for a proof see [43].

▶ **Lemma 61** (Coefficient-Extraction). *Let $L(f) = s$ with $f \in \mathbb{F}[\mathbf{x}]$ and $f = \sum_{0\le i\le d} f_i x_n^i$ with $f_i \in \mathbb{F}[x_1, x_2, \ldots, x_{n-1}]$. Then there is a circuit C of size $O(sd^2)$ computing $f_0, f_1, \ldots, f_d$.*

The next lemma is a homogenization trick, used in [45]. For a proof, see [43, Theorem 2.2].

▶ **Lemma 62** (Homogenization). *If f has an arithmetic circuit of size s, then for any d, there is a circuit of size $O(sd^2)$ computing $Hom_{\le d} f$.*

▶ **Lemma 63.** *Let f be a polynomial $\mathbb{F}[x]$, computed by a size s circuit C. Then, there exists a circuit C' of size $O(sm^2)$ which computes $f, f^{(1)}, f^{(2)}, \ldots, f^{(m)}$.*

Proof. We split every G gate in C to $n+1$ gates $G_0, \ldots, G_m$ in C'. The property we want is that if the gate G is computing the polynomial g in C then G_k computes the polynomial $g^{(k)}$ in C'. Suppose G is a $+$ gate in C with children gates computing the polynomials g_1 and g_2. Now we know that $g^{(k)} = g_1^{(k)} + g_2^{(k)}$. Thus we can easily propagate the derivatives on addition/subtraction gates. If G is a $\times$ gate then using Lemma 66, we know that:

$$(g_1 g_2)^{(k)} = \sum_{i=0}^{k} \binom{k}{i} g_1^{(k-i)} g_2^{(i)}$$

Thus we can computes $g, g^{(1)}, g^{(2)}, \ldots, g^{(m)}$ using additional $O(m^2)$ gates. Therefore C' has $O(sm^2)$ gates. ◀

Polynomial Identity Testing (PIT) is a fundamental question in algebraic complexity. It asks for an algorithm to test the zeroness of a given algebraic circuit via mere query access. It is known that efficient evaluation at random points lead to a randomized polynomial time algorithm for PIT. This is known as *Polynomial Identity Lemma* [38, 15, 51, 41].

▶ **Lemma 64** (Polynomial Identity Lemma). *Let $p(\mathbf{x})$ be an n-variate nonzero polynomial of degree d. Let $S \subseteq \mathbb{F}$ be a finite set. Then,*

$$\Pr_{\boldsymbol{\alpha}\sim S^n} [p(\boldsymbol{\alpha}) = 0] \le d/|S| .$$

Here, $\boldsymbol{\alpha} \in S^n$ is picked independently and uniformly at random.

B Basic mathematical tools

▶ **Lemma 65** (Power series valuation). *Let $g, h \in \mathbb{F}[x]$ such that $g/h \in \mathbb{F}[[x]]$. Let m (respec. n) be the highest power dividing g (respec. h) i.e. $x^m \mid g$ and $x^{m+1} \nmid g$ (respec. for h). Then, $m \ge n$.*

Proof. Suppose, $m < n$. Note that, there exists $0 \ne \alpha \in \mathbb{F}$, such that $h = \alpha\, x^n \cdot (1 + x\, \tilde{h})$, for some $\tilde{h} \in \mathbb{F}[x]$. Similarly, let $g = \beta\, x^m \cdot (1 + x\, \tilde{g})$, for some $\tilde{g} \in \mathbb{F}[x]$ and $\beta \in \mathbb{F}$. Thus,

$$\begin{aligned}\frac{g}{h} &= \frac{\beta}{\alpha} \cdot x^{m-n} \cdot \frac{1 + x\,\tilde{g}}{1 + x\,\tilde{h}} \\ &= \frac{\beta}{\alpha} \cdot x^{m-n} \cdot (1 + x\,\tilde{g}) \cdot (1 + x\,\tilde{h} + (x\,\tilde{h})^2 + \cdots) \\ &\notin \mathbb{F}[[x]]\,, \text{ a contradiction}\,.\end{aligned}$$ ◀

▶ **Lemma 66** (General Leibniz rule). *If f and g are k-time differentiable functions, then*

$$(fg)^{(k)} = \sum_{i=0}^{k} \binom{k}{i} f^{(k-i)} g^{(i)} .$$

▶ **Lemma 67.** *Define $f_d := \sum_{i=0}^{d} x^i$. Then, $S_{\mathbb{F}}(f_d) \le 9 \cdot d^{1/2}$, over any field $\mathbb{F}$.*

Proof of Lemma 67. Fix some $n \in \mathbb{N}$. Note that,

$$f_{n^2-1}(x) \;=\; \left(1 + x + \ldots + x^{n-1}\right) \cdot \left(1 + x^n + \ldots + x^{n(n-1)}\right) .$$

As each factor has n terms, we can write the product as sum of two squares with each polynomial having at most $2n$ terms. Therefore,

$$S_{\mathbb{F}}(f_{n^2-1}(x)) \;\le\; 4n . \tag{6}$$

For general d, let $n \in \mathbb{N}$ be such that $n^2 - 1 \le d < (n+1)^2 - 1$. By definition,

$$f_d(x) \;=\; f_{n^2-1}(x) + x^{n^2} \cdot f_{d-n^2}(x) .$$

Note that, $|f_{d-n^2}(x)|_0 \;\le\; d + 1 - n^2 \le 2n$. Thus, using the trivial upper bound on $S(f)$, we must have

$$S_{\mathbb{F}}(x^{n^2} \cdot f_{d-n^2}(x)) \le 2 \cdot (2n+1) . \tag{7}$$

Combining Equation (6) and Equation (7), we get that $S_{\mathbb{F}}(f_d(x)) \le 8 \cdot \lceil \sqrt{d+1} \rceil + 2$, and the conclusion follows. ◀

C Monic transformation

Given any polynomial $p(\mathbf{x})$ in variables $\mathbf{x} = (x_1, \ldots, x_n)$, there is a standard trick to make it monic in x_n by applying a linear transformation on the variables: for $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n-1}) \in \mathbb{F}^{n-1}$, let

$$\tau_{\boldsymbol{\alpha}} : x_i \mapsto \alpha_i x_n + x_i ,$$

for $i \in [n-1]$, and $x_n \mapsto x_n$. Note that $\deg(\tau_{\boldsymbol{\alpha}}(p)) \le \deg(p)$ [it may *decrease* because of non-trivial cancellations]. It is easy to see that $\tau_{\boldsymbol{\alpha}}$ is an *invertible* map. We show that $\tau_{\boldsymbol{\alpha}}(p)$ is monic in x_n, for a *random* transformation $\tau_{\boldsymbol{\alpha}}$ i.e. when $\boldsymbol{\alpha} \in \mathbb{F}^{n-1}$ chosen randomly. In fact, we show that this map can simultaneously make polynomials monic given that the field $\mathbb{F}$ is sufficiently large.

▶ **Lemma 68** (Monic Transformation). *Let $p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})$ be m-many polynomial of degree d. Let $S \subseteq \mathbb{F}$ be a finite set. For $\boldsymbol{\alpha} \in S^{n-1}$, picked independently and uniformly at random,*

$$\Pr\left[\bigwedge_{i=1}^{m} \tau_{\boldsymbol{\alpha}}(p_i(\mathbf{x}))\; is\ monic\ in\, x_n\right] \;\ge\; 1 - \frac{dm}{|S|} .$$

Proof. Consider the terms of degree d of a non-zero polynomial $p \in \mathbb{F}[\mathbf{x}]$. Define the set

$$T \;:=\; \{\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n) \;\mid\; |\boldsymbol{\beta}|_0 = \sum_i \beta_i = d, \text{ and } \mathrm{coef}_{\mathbf{x}^{\boldsymbol{\beta}}}(p) \neq 0\} .$$

We also denote $\boldsymbol{\beta}' = (\beta_1, \ldots, \beta_{n-1})$, the first $n-1$-coordinates of $\boldsymbol{\beta}$, and similarly $\mathbf{x}' = (x_1, \ldots, x_{n-1})$. Note that, $\tau_{\boldsymbol{\alpha}}(\mathbf{x}^{\boldsymbol{\beta}}) = \boldsymbol{\alpha}^{\boldsymbol{\beta}'} \cdot x_n^d + (\text{lower terms in}\, x_n)$.

Observe that the homogeneous component of degree d in $\tau_{\boldsymbol{\alpha}}(p)$ can be written as $a_{d,p}(\mathbf{x}) = \sum_{\boldsymbol{\beta}\in T} c_{\boldsymbol{\beta}} \cdot \tau_{\boldsymbol{\alpha}}(\mathbf{x}^{\boldsymbol{\beta}})$, for some constants $c_{\boldsymbol{\beta}}$. Trivially, $a_{d,p}$ is a nonzero polynomial, and moreover,

$$a_{d,p}(\boldsymbol{\alpha}) = (\sum_{\boldsymbol{\beta}\in T} c_{\boldsymbol{\beta}} \boldsymbol{\alpha}^{\boldsymbol{\beta}'}) \cdot x_n^d + (\text{lower terms in } x_n).$$

In order to make $\tau_{\boldsymbol{\alpha}}(p)$ monic in x_n, we want $(\sum_{\boldsymbol{\beta}\in T} c_{\boldsymbol{\beta}} \boldsymbol{\alpha}^{\boldsymbol{\beta}'}) \neq 0$. So, define, another polynomial $b_{d,p}(\mathbf{x}') = (\sum_{\boldsymbol{\beta}\in T} c_{\boldsymbol{\beta}} \mathbf{x}'^{\boldsymbol{\beta}'})$. It can have degree atmost d.

As we want each $\tau_{\boldsymbol{\alpha}}(p)$ monic where $p = p_m(\mathbf{x})$, it suffices to find $\boldsymbol{\alpha}$ such that $\prod_{i\in[m]} b_{d,p_i}(\boldsymbol{\alpha}) \neq 0$. Note that, $\deg\left(\prod_{i\in[m]} b_{d,p_i}(\mathbf{x})\right) \leq d \cdot m$. Thus, when we pick $\boldsymbol{\alpha}$ at random, the probability that $\prod_{i\in[m]} b_{d,p_i}(\boldsymbol{\alpha}) = 0$, is at most $\leq dm/|S|$, from Lemma 64. Hence, the conclusion follows. ◀

D Truncation is hard

One can show that truncation (or cost of mod) cannot be *expected* to be logarithmically dependent on the precision (unless permanent is *easy*), reminiscent to [46]. We sketch the proof for the sake of completeness.

▶ **Lemma 69** (Folkore). *Suppose, for any polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ of size s, Hom$_{\leq d} f(\mathbf{x})$ can be computed by circuit of size* $\mathrm{poly}(s, \log d)$*, then* VP = VNP.

Proof. Consider the following polynomial of $n^2 + n$ variables, where we denote $\mathbf{y} = (y_1, \ldots, y_n)$, and $\mathbf{z} = (z_{1,1}, \ldots, z_{n,n})$:

$$g(\mathbf{y}, \mathbf{z}) \ := \ \prod_{i\in[n]} \left(\sum_{j\in[n]} y_j z_{i,j} \right)$$

Observe that coefficient of $y_1 \ldots y_n$ in g is nothing but $\mathrm{perm}(z_{1,1}, \ldots, z_{n,n})$, the permanent polynomial on variables $\mathbf{z}$. Further, each $\mathrm{coef}_{\mathbf{y}^{\boldsymbol{\alpha}}}(g)$ is a *multilinear* polynomial in $\mathbf{z}$, of degree n. Consider a new polynomial f by substituting $y_i = x^{(n+1)^{i-1}}$ (Kronecker substitution). In particular, let

$$f(x, \mathbf{z}) \ := \ g(x, x^{n+1}, x^{(n+1)^2}, \ldots, x^{(n+1)^{n-1}}, \mathbf{z}).$$

As Kronecker substitution gives different weights to different monomials and the maximum degree can be $n \cdot (n+1)^{n-1}$ (i.e. when y_n^n gets substituted), it is easy to deduce that

$$f \ = \ \sum_{k=0}^{n\cdot(n+1)^{n-1}} c_k(\mathbf{z}) \cdot x^k \,.$$

Here, each $c_k(\mathbf{z})$ is a multilinear polynomial of degree n. Moreover, from the above discussion,

$$c_j(z_{1,1}, \ldots, z_{n,n}) \ = \ \mathrm{perm}(z_{1,1}, \ldots, z_{n,n}) \ , \text{where } j \ := \ 1 + (n+1) + \ldots + (n+1)^{n-1} \,.$$

In that case, the degree of $c_j(\mathbf{z}) \cdot x^j$ is $m := j + n = n^{O(n)}$. Thus, we can conclude that $\mathrm{Hom}_{=m}(f) = c_j(\mathbf{z}) \cdot x^j = \mathrm{perm}(\mathbf{z}) \cdot x^j$.

Observe that $L(g) \le \mathrm{poly}(n)$. After Kronecker substitution, the blowup in size in still poly i.e. $L(f) \le \mathrm{poly}(n)$. Hence, assuming the hypothesis, we would get that

$$\mathrm{perm}(\mathbf{z}) \cdot x^j \;=\; \mathrm{Hom}_{=m}(f) \;=\; \mathrm{Hom}_{\le m}(f) \;-\; \mathrm{Hom}_{\le m-1}(f)\,,$$

has $\mathrm{poly}(n)$ size circuit. This implies $\mathrm{perm}(\mathbf{z})$ has $\mathrm{poly}(n)$ size circuit (by substituting $x = 1$), i.e. $\mathsf{VP} = \mathsf{VNP}$. ◀

E Details for Section 3

Here we prove Lemma 16. For completeness, we again state the lemma.

▶ **Lemma 70.** *Suppose $g = \sum_{i \le d_1} g_i x^i$ and $h = x^{d_2} + \sum_{i<d_2} h_i\, x^i$, in $\mathbb{F}[\mathbf{x}]$. Suppose $g = hq + r$, with $r = \sum_{i<d_2} r_i x^i$ and $q = \sum_{i \le d_1-d_2} q_i x^i$. Then, there is a circuit of size $O(d_1\, d_2)$, whose inputs are all h_i, g_i and outputs are all r_i, q_i.*

Proof. We shall denote the desired circuit by C_{d_1,d_2}. So we want:

$$C_{d_1,d_2}(g_0, g_1, \ldots, g_{d_1}, h_0, h_1, \ldots, h_{d_2}) = (r_1, r_2, \ldots, r_{d_2-1}, q_0, q_1, \ldots, q_{d_1-d_2}).$$

If $d_1 < d_2$, we know that $q = 0$. Hence:

$$C_{d_2-1,d_2}(g_0, g_1, \ldots, g_{d_1-1}, h_0, h_1, \ldots, h_{d_1}) = (g_1, g_2, \ldots, g_{d_1-1}).$$

If $d_1 > d_2$, we perform a long division step:

$$g \leftarrow g - h \cdot x^{d_1-d_2} \cdot g_{d_1} = \sum_{i \le d_1-d_2-1} g_i\, x^i \;+\; \sum_{i \ge d_1-d_2}^{d_1-1} \left(g_i - h_{i-(d_1-d_2)}\, g_{d_1}\right) x^i .$$

Note that, we can set $q_{d_1-d_2} = g_{d_1}$. Define:

$$\mathbf{g} \stackrel{\text{def}}{=\!=} (g_0, g_1, \ldots, g_{d_1-d_2-1}, g_{d_1-d_2} - h_0 g_{d_1}, \ldots, g_{d_1-1} - h_{d_2-1} g_{d_1})\,.$$

Then we have:

$$C_{d_1,d_2}(g_0, g_1, \ldots, g_{d_1}, h_0, h_1, \ldots, h_{d_2}) = (C_{d_1-1,d_2}(\mathbf{g}, h_0, h_1, \ldots, h_{d_2}), g_{d_1}). \tag{8}$$

Hence if $S(d_1, d_2)$ is the size of C_{d_1,d_2} then Equation (8) implies that $S(d_1, d_2) = S(d_1 - 1, d_2) + 2d_2$ and $S(d_2 - 1, d_2) = 2d_2 - 1$. Therefore $S(d_1, d_2) \le 2\, d_1\, d_2$. ◀

F Conditional hardness of $\sqrt{1+4x}$

We first show that $\sqrt{1+4x} \in \mathbb{Z}[[x]]$.

▶ **Lemma 71** (Folklore). *We have $\sqrt{1+4x} = \sum_{i \ge 0} \binom{2i}{i} / (2i-1)\, x^i \;\in\; \mathbb{Z}[[x]]$.*

Proof. We know that, $\sqrt{1+4x} \;=\; \sum_{i\ge0} \binom{1/2}{i} (4x)^i$. Now, it is easy to see that:

$$\binom{\frac{1}{2}}{d} \;=\; \frac{\frac{1}{2} \cdot \left(\frac{1}{2}-1\right) \cdot \left(\frac{1}{2}-2\right) \cdot \cdots \cdot \left(\frac{1}{2}-d+1\right)}{d!} \;=\; (-1)^{d-1} \cdot \frac{\binom{2d}{d}}{4^d (2d-1)}.$$

This implies that $\sqrt{1+4x} = \sum_{i\ge0} \binom{2i}{i}/(2i-1)\, x^i$. Further, it is also easy to verify that

$$\binom{2d}{d} \;=\; \left(4\binom{2d-2}{d-1} - \binom{2d}{d}\right) \cdot (2d-1) \;\Longrightarrow\; \binom{2d}{d} / (2d-1) \;\in\; \mathbb{N}\,.$$

Therefore, $\sqrt{1+4x} \in \mathbb{Z}[[x]]$, as desired. ◀

Lemma 71 implies that all the truncations of $\sqrt{1+4x}$ can be computed by division-free circuits.

▶ **Theorem 72.** *If $\tau(\mathrm{trunc}\left(\sqrt{1+4x}, d\right) = O(\log^c d)$, for some constant $c \in \mathbb{N}$, then $(d!)$ is easy. In fact, $\tau(d!) = O(\log^{c+1} d)$.*

Proof. By assumption, we know that $\tau(\mathrm{trunc}\left(\sqrt{1+4x}, d-1\right) = O(\log^c d)$ and $\tau(\mathrm{trunc}\left(\sqrt{1+4x}, d\right) = O(\log^c d)$. By using Lemma 71, we see that:

$$\mathrm{trunc}\left(\sqrt{1+4x}, d\right) - \mathrm{trunc}\left(\sqrt{1+4x}, d-1\right) = (-1)^{d-1} x^d \frac{\binom{2d}{d}}{2d-1}.$$

Hence, $\tau((-1)^{d-1}x^d \cdot \binom{2d}{d}/(2d-1) = O(\log^c d)$. Therefore $\left((-1)^{n-1}\binom{2d}{d}/(2d-1)\right)$ has complexity $O(\log^c d)$ by substituting $x = 1$. This also implies that $\binom{2d}{d}$ has complexity $O(\log^c d)$. Invoking Lemma 29, we conclude. ◀

▶ **Corollary 73.** *If $(d!)$ has complexity $\omega(\mathrm{poly}(\log d))$, then $\tau(\mathrm{trunc}\left(\sqrt{1+4x}, d\right) = \omega(\mathrm{poly}(\log d))$.*

G Integral power series: Details for Section 6

We will use some number-theoretic tool to show that the candidate power series is integral. So, before delving into that, we go through some preliminary tools being used.

▶ Fact 74 (Folklore). Product of any k consecutive positive integers is divisible by $k!$.

▶ **Definition 75** (p-adic valuation). *Let p be a prime and $n \in \mathbb{Z}$. We denote p-adic valuation of n as $\nu_p(n)$ to be the* highest exponent *such that $p^{\nu_p(n)} \mid n$. Formally, $\nu_p : \mathbb{Z} \longrightarrow \mathbb{N}$ defined by*

$$\nu_p(n) = \max\{v \in \mathbb{N} : p^v \mid n\}.$$

Note that, by definition, $\nu_p(\mathrm{rad}(n)) = 1$ if $p \mid n$, and 0 otherwise.

▶ **Theorem 76** (Legendre's formula). *For a prime p and $n \in \mathbb{N}$, $\nu_p(n) = \sum_{j=1}^{\infty} \lfloor n/p^j \rfloor$.*

Now, we prove integrality of a power series which is our candidate algebraic function for Theorem 8. It suffices to prove the integrality of $(1+k^2x)^{1/k}$, which we prove below.

▶ **Theorem 77** (Restatement of Theorem 31, Integral power series). *Let $k \in \mathbb{N}$. Define $f_k(x) := \left(1 + k^2 \cdot x\right)^{1/k}$. Then, $f_k(x) \in \mathbb{Z}[[x]]$.*

Proof. By binomial expansion, $f_k \in \mathbb{Q}[[x]]$. Let $f_k(x) = \sum_{d \geq 0} a_d \cdot x^d$. We'll prove by *strong induction* that indeed the coefficients are integers.

Obviously $a_0 = 1$, and assume that for $m \in \mathbb{N}$ we have proved that $a_\ell \in \mathbb{Z}$ for $0 \leq \ell < m$. The coefficient at x^m in $\left(\sum_{d=0}^{\infty} a_d x^d\right)^k = \left(1 + \sum_{d=1}^{\infty} a_d x^d\right)^k$ is equal to $k \cdot a_m$ plus a bunch of terms that we know are integer by the induction hypothesis; hence $k \cdot a_m = b \in \mathbb{Z}$. But by the binomial series formula we have

$$a_m = k^{2m} \cdot \binom{1/k}{m} = \frac{k^{2m} \cdot \prod_{j=0}^{m-1}(1/k - j)}{m!} = \frac{k^m \cdot \prod_{j=0}^{m-1}(1 - kj)}{m!}.$$

It suffices to prove that $k \mid b$. If we can show that $\nu_p(b) \geq \nu_p(k)$ for every prime p dividing k, this would certainly imply that $k \mid b$. So, fix a prime $p \mid k$. Note that

$$b \;=\; k \cdot a_m \;= X/m!\,, \text{ where } X \;:=\; k^{m+1} \cdot \prod_{j=0}^{m-1} (1-kj)\,.$$

As, $p \mid k$, we must have $\prod_{j=0}^{m-1}(1-kj) \equiv 1 \bmod p$. Thus, $\nu_p(X) = \nu_p\left(k^{m+1}\right) = (m+1)\nu_p(k)$. And by Theorem 76, $\nu_p(m!) = \sum_{j=1}^{\infty} \lfloor m/p^j \rfloor < \sum_{j=1}^{\infty} m/p^j = m/p - 1 \leq m$. Thus,

$$\nu_p(b) \;=\; \nu_p(X) - \nu_p(m!) \;\geq\; (m+1)\nu_p(k) - m \geq \nu_p(k)\,,$$

as we wanted. Putting it together gives $a_m \in \mathbb{Z}$ proving the inductive step. Hence, the conclusion follows. ◀

H From hardness of algebraic functions to hardness of permanent in constant-free regime

Here, we sketch why one of the truncations being hard implies permanent does not have small constant-free circuits (implying $\mathsf{VP}_0 \neq \mathsf{VNP}_0$). The proof is reminiscent to [9]. We point out the main components. We denote Perm_n as the permanent polynomial of a $n \times n$ symbolic matrix.

▶ **Theorem 78** (Hardness of permanent). *Let us fix $i, k \in \mathbb{N}$ such that $i < k$. Further, assume that, $L\left(\mathrm{trunc}\left((1+k^2\,x)^{i/k}\right), d\right) = \omega(\mathrm{poly}(\log d))$, then $\tau(Perm_n) = \omega(\mathrm{poly}(\log n))$.*

▶ Remark 79. One can also prove a conditional implication referring to the original Valiant hypothesis $\mathsf{VP}_{\mathbb{C}} \neq \mathsf{VNP}_{\mathbb{C}}$, assuming GRH (Generalized Riemann Hypothesis). This has also been pointed out in [9, Corollary 4.2]. This basically follows from the fact that under GRH and assuming $\mathsf{VP} = \mathsf{VNP}$, then $\mathsf{CH} \subseteq \mathsf{P}/\mathrm{poly}$.

Before going into the proof sketch, we define CH-*definable* sequences. The *counting hierarchy* is denoted by CH [50]. The class of poly-size circuits can be expressed by the nonuniform *advice class* P/poly.

Let $q(n)$ be a polynomial. Let $a = (a(n,\ell))_{n \in \mathbb{N},\ \ell \leq q(n)}$ be a sequence of integers such that $a(n,\ell)$ has *exponential bitsize*, i.e., $|a(n,\ell)| \leq 2^{n^c}$ for all k and some constant c. We think of n, ℓ as being represented in binary using $O(\log n)$ bits.

With the sequence, we associate a language that determines the bits of $a(n,\ell)$ in binary,

$$\begin{aligned} \mathrm{Sgn}(a) \;&=\; \{(n,\ell) \mid a(n,\ell) \geq 0\}, \\ \mathrm{Bit}(a) \;&=\; \{(n,\ell,j,b) \mid \text{ the } j\text{-th bit of } a(n,\ell) \text{ equals } b\}\,. \end{aligned}$$

▶ **Definition 80** ([9, Definition 3.2]). *The sequence $a = (a(n,\ell))_{n,\ell}$ of integers of exponential bitsize is* CH-definable *if* $\mathrm{Sgn}(a) \in \mathsf{CH}$ *and* $\mathrm{Bit}(a) \in \mathsf{CH}$.

The sequences of integers that are definable in CH are *closed* under iterated addition, iterated multiplication, and integer division [9, Theorem 3.10]. Koiran et al. [26, Theorem 2.14] used the binary version of the same theorem.

▶ **Theorem 81** ([9, 26]).

(i) *Let $q(n)$ be a polynomial and suppose $(a(n,\ell))_{n\in\mathbb{N},\ell\le q(n)}$ is* CH*-definable. Then the sum- and product-sequences $b(n)$ and $c(n)$ are* CH*-definable, where*

$$b(n) = \sum_{\ell=0}^{q(n)} a(n,\ell) \qquad \text{and} \qquad c(n) = \prod_{\ell=0}^{q(n)} a(n,\ell)\,.$$

(ii) *Suppose $(s(n))_{n\in\mathbb{N}}$ and $(t(n))_{n\in\mathbb{N}}$ are definable in* CH *and $t(n)>0$ for all n. Then the sequence of quotients $\lfloor s(n)/t(n)\rfloor_{n\in\mathbb{N}}$ is definable in* CH*.*

Now, we state the most important theorem proven in [9, Theorem 4.1] from which Theorem 78 will follow almost trivially.

▶ **Theorem 82.** *Let q be a polynomially bounded function and $(b(n,\ell))_{n\in\mathbb{N},\ell\le q(n)}$ and $(d(n))_{n\in\mathbb{N}}$ are definable in* CH*. Let*

$$f_n = \sum_{\ell=0}^{q(n)} b(n,\ell)x^\ell \in \mathbb{Z}[x]\,,\; g_n = f_n/d(n) \in \mathbb{Q}[x]\,.$$

If $\tau(Perm_n) = \mathrm{poly}(\log n)$, then $L_{\mathbb{Q}}(g_n) = \mathrm{poly}(\log n)$.

Now, we are ready to prove Theorem 78.

Proof sketch of Theorem 78. Let, $(1+k^2\cdot x)^{i/k} := \sum_{j\ge 0} a_{i,j}\,x^j \in \mathbb{Z}[[x]]$. By binomial expansion, we have

$$a_{i,j} = k^{2j}\cdot\binom{i/k}{j} = k^j/j!\cdot\prod_{\ell=0}^{j-1}(i-k\ell)\,.$$

As k is a constant, $\prod_{\ell=0}^{j-1}(i-k\ell), j!, k^j$ are all trivially definable in CH, by Theorem 81. Further, by Theorem 77, $a_{i,j}\in\mathbb{Z}$ implying $(a_{i,j})$ CH-definable, again by Theorem 81.

The rest directly follows from Theorem 82. Note that, if $\tau(\mathrm{Perm}_n) = \mathrm{poly}(\log n)$, then from the above argument, truncation of the power series upto n i.e. $f_n = \sum_{j=0}^{n} a_{i,j}x^j$ must be *easy*, as the coeffecients are CH-definable. This directly contradicts our assumption that the truncation is hard. Hence, permanent cannot have polynomial size constant-free circuits. ◀

I Algorithm

On the following page, we write the algorithm for the first part of Theorem 8.

Algorithm 1 Integer factorization assuming the truncations of $(1+k^2x)^{i/k}$ being easy for each i.

```
Input: A composite positive integer $n$.
Output: A non-trivial factor of $n$.
Define $N(d,k) := \frac{k^{(k-2)d}(dk)!}{(d!)^k}$.
$m \leftarrow k$.
while true do
    Compute $\gcd(N(m,k),n)$.
    if $\gcd(N(m,k),n) = 1$ then
        $m \leftarrow mk$.
    else if $\gcd(N(m,k),n) = n$ then
        $t \leftarrow m$.                        ▷ This $m$ is the desired $t$.
        break
    else
        return $\gcd(N(m,k),n)$      ▷ Here $\gcd(N(m,k),n)$ is a non-trivial factor of $n$.
    end if
end while
              ▷ At this step, all the primes dividing $n$ are in the interval $[t+1, tk]$.
$a \leftarrow 1$.
$b \leftarrow t$.
while true do
    if $b - a \leq 1$ then
        $s \leftarrow a$.                        ▷ This $a$ is the desired $s$.
        break
    end if
    $c \leftarrow \lceil (a+b)/2 \rceil$.
    Compute $\gcd(N(c,k),n)$.
    if $\gcd(N(c,k),n) = 1$ then
        $a \leftarrow c$.
    else if $\gcd(N(c,k),n) = n$ then
        $b \leftarrow c$.
    else
        return $\gcd(N(c,k),n)$      ▷ Here $\gcd(N(c,k),n)$ is a non-trivial factor of $n$.
    end if
end while
              ▷ At this step, all the primes dividing $n$ are in the interval $[sk+1, (s+1)k]$.
for $i = sk+1$ to $(s+1)k$ do
    if $i$ divides $n$ then
        return $i$                         ▷ Here $i$ is a non-trivial factor of $n$.
    end if
end for
```

SOS Lower Bound for Exact Planted Clique

Shuo Pang ✉ 🏠
Mathematics Department, University of Chicago, IL, USA

Abstract

We prove a SOS degree lower bound for the planted clique problem on the Erdös-Rényi random graph $G(n, 1/2)$. The bound we get is degree $d = \Omega(\epsilon^2 \log n/\log\log n)$ for clique size $\omega = n^{1/2-\epsilon}$, which is almost tight. This improves the result of [5] for the "soft" version of the problem, where the family of the equality-axioms generated by $x_1 + ... + x_n = \omega$ is relaxed to one inequality $x_1 + ... + x_n \geq \omega$.

As a technical by-product, we also "naturalize" certain techniques that were developed and used for the relaxed problem. This includes a new way to define the pseudo-expectation, and a more robust method to solve out the coarse diagonalization of the moment matrix.

2012 ACM Subject Classification Theory of computation → Proof complexity; Mathematics of computing → Random graphs; Theory of computation → Semidefinite programming

Keywords and phrases Sum-of-Squares, planted clique, random graphs, average-case lower bound

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.26

Related Version *ECCC Technical Report*: `https://eccc.weizmann.ac.il/report/2021/070`

Acknowledgements I am very grateful to Aaron Potechin for the introduction of the problem and the encouraging communications, and to Alexander Razborov for the advice and help on improving the quality of the paper. My thanks also go to the anonymous reviewers for their constructive criticism of the presentation.

1 Introduction

1.1 The problem and the proof system

Whether one can find a max-clique in a random graph $G \sim G(n, 1/2)$ efficiently and be correct with high probability has been a long-standing open problem in computational complexity since [19]. In [18, 22], a relaxed formulation as the *planted clique problem* was introduced: if we further plant a random clique of size $\omega \gg \log n$ to G, can it be efficiently recovered? Information-theoretically this is possible, since w.h.p. the largest clique in G has size $(2 + o(1)) \log n$. While computationally, the average-case hardness of this problem is still widely believed even after it has been intensively studied and has inspired research directions in an extremely wide range of fields (just to mention a few: cryptography [2], learning theory [8], mathematical finance [3], computational biology [28]). So far, the best known polynomial-time algorithm is for $\omega = \Omega(\sqrt{n})$ [1], which is a so-called spectral algorithm (see e.g. [17]).

The sum-of-squares (SOS) hierarchy [30, 27, 23] is a stronger family of semidefinite programming (SDP) algorithms which, roughly speaking, is SDP on the extended set of variables $\{x_{i(1)}...x_{i(d)} \mid i_1, ..., i_d \in [n]\}$ according to the degree parameter d, and it can be significantly more powerful than spectral algorithms and traditional SDP (see e.g. [4, 17]). Recent years have witnessed rapid development on SOS-based algorithms which turn out to provide a characterization of a wide class of algorithmic techniques – for a list of evidence, we refer the reader to the survey [6] and the introduction of [17]. The SOS proof system is the natural proof-theoretic counterpart of these algorithms, also known as the *Positivstellensatz* system [14]: it works with polynomials over $\mathbb{R}$, and given polynomial equalities (axioms) $f_1(x) = 0, ..., f_k(x) = 0$ on $x = (x_1, ..., x_n)$, a proof (that is, a refutation of the existence of a solution) is

$$-1 = \sum_{i=1}^{k} f_i q_i + \sum_j r_j^2 \quad \text{in } \mathbb{R}[x_1, ..., x_n]$$

where $q_1, ..., q_m$ and $r_1, ...$ are arbitrary polynomials on $x_1, ..., x_n$ over $\mathbb{R}$. Under certain conditions, in particular when all variables are boolean ($x_i^2 = x_i$), such an refutation always exists if the axioms are contradictory. The *degree-d SOS proof system* is this plus a degree limitation

$$\max_{i,j}\{\deg(f_i) + \deg(q_i),\ 2\deg(r_j)\} \leq d.$$

For more about the relation between the SOS proofs and SDP algorithms, see e.g. [26, 29]. The average-case hardness of the clique problem has a very simple form in proof complexity: for $G \sim G(n, 1/2)$, can the proof system efficiently refute the existence of a size-ω ($\gg \log n$) clique w.h.p.? Note the system cannot just say "No" but must search for a certificate – a proof. A lower bound here would automatically give the hardness on any class of algorithms based on the proof system. Given that the decision version of the spectral algorithm of [1] corresponds to a degree-2 SOS proof, a SOS degree lower bound would bring us a much better understanding of the hardness of the problem. The standard formulation is the following.

▶ **Definition 1.1.** *Given an n-vertex simple graph G and a number ω, the **Clique Problem** for degree-d SOS proof system has the following **axioms**.*

$$\begin{aligned} &\text{(Boolean)} \quad && x_i^2 = x_i \quad \forall i \in [n] \\ &\text{(Clique)} \quad && x_i x_j = 0 \quad \forall \{i, j\} \textit{ non-edge} \\ &\text{(Size)} \quad && x_1 + ... + x_n = \omega \end{aligned} \tag{1.1}$$

To confirm no ω-clique exists is to give a SOS refutation of the above. The SOS system has the so-called duality: to show degree lower bound it suffices to consider *pseudo-expectation* and the resulting *moment matrix*[1]. With boolean variables (which is our case), this can be demonstrated on multi-linear polynomials. Let $\mathcal{X}^{\leq a} = \{x_S \mid S \subseteq [n],\ |S| \leq a\}$ for any a.

▶ **Definition 1.2.** *A **degree-d pseudo-expectation** for the Clique Problem on G is a map $\widetilde{E} : \mathcal{X}^d \to \mathbb{R}$ satisfying the following four **constraints** when extended by $\mathbb{R}$-linearity.*

$$\text{(Default)} \quad \widetilde{E}x_\emptyset = 1 \tag{1.2}$$

$$\text{(Clique)} \quad \widetilde{E}x_S = 0, \quad \forall S : |S| \leq d,\ G|_S \textit{ non-clique} \tag{1.3}$$

$$\text{(Size)} \quad \widetilde{E}\Big((x_1 + ... + x_n)x_S\Big) = \omega \cdot \widetilde{E}x_S \quad \forall S :\ |S| \leq d - 1 \tag{1.4}$$

*where in (1.4), $x_A \cdot x_B := x_{A\cup B}$. For the last constraint, define the **moment matrix** M to be the $\binom{[n]}{\leq d/2} \times \binom{[n]}{\leq d/2}$ matrix[2] with expression $M(A, B) = \widetilde{E}x_{A\cup B}$ for all $|A|, |B| \leq d/2$, then:*

$$\text{(PSDness)} \quad M \textit{ is positive semi-definite.} \tag{1.5}$$

It is not hard to see that if a degree-d pseudo-expectation exists then there is no degree-d SOS refutation.

[1] We use the name for simplicity. More cautiously, it should be called the *pseudo-moment matrix*.

[2] d is always assumed to be even.

A relaxation of the problem was studied in [5]: decide whether there exists $\widetilde{E}$ as in Definition 1.2 except by one change – replace Size Constraints by one weaker inequality $\widetilde{E}(x_1 + ... + x_n) \geq \omega$. Henceforth, we call the Clique Problem (Def. 1.1) **Exact Clique** and this relaxation **Non-Exact Clique**.[3] We will study their average-case hardness over $G \sim G(n, 1/2)$.

How to deal with the exact problem is a subtle but important open problem. On the problem itself, lower bounds on the "weak" formulation indeed gave the important algorithmic message – an integrality gap for many SOS-based optimization algorithms – but still, they do not rule out the possibility that SOS can efficiently refute $x_1 + ... + x_n = k$ for each individual large k, and the distinction between "weak" and "strong" formulations also involves how one thinks *the* SOS SDP optimization problem should be formulated.

Perhaps more importantly, it is about the limit of existing methods for proving average-case SOS lower bounds. Current techniques from the so-called *pseudo-calibration heuristic* [5] tend to deal successfully with "soft" constraints (i.e. inequalities, or usually just one bound on a single pseudo-expectation value) while being poor at handling "hard" constraints (i.e. equalities). Finding techniques to deal with the latter is thus in need. Progress toward this goal is made in [20] for random CSPs, where the number of hard constraints is at most two[4]. Their method is to break such constraint(s) into local ones and satisfy each using real, independent distributions. For "inherently more rigid" problems like Exact Clique (whose hard constraints are "almost everywhere"), however, it seems unlikely a similar strategy could work.

Lastly, there are concrete applications of lower bounds on Exact Clique. Such a lower bound can give by reduction lower bounds for other problems, e.g. for the approximated Nash-Welfare, and potentially for the coloring problem and stochastic block models [20, 21].

1.2 Previous work

For upper bounds, if $\omega = \Omega(\sqrt{n})$ then degree-2 SOS can refute Exact Clique with high probability [12]. On the other hand, if $\omega > d \geq 2.1 \log n$, a degree-$d$ SOS refutation for Exact Clique is not hard to see; since we have not been able to find it in the literature, we include it as Observation 1.3 below.

For lower bounds, for Exact Clique, [13] showed that the weaker system *d-round Lovasz-Schrijver* cannot refute it when $\omega = O(\sqrt{n/2^d})$; [25] proved degree-d lower bound on SOS for $\omega = \widetilde{O}(n^{1/d})$, and this bound on ω was improved to $\widetilde{O}(n^{1/3})$ for $d = 4$ [10] and $\widetilde{O}(n^{\frac{1}{\lfloor d/2 \rfloor + 1}})$ for general d [15]. For Non-Exact Clique, [5] proved the almost tight lower bound $d = \Omega(\epsilon^2 \log n)$ for $\omega = n^{1/2-\epsilon}$, $\epsilon > 0$ arbitrary (could depend on n).

▶ **Observation 1.3** (Upper bound for Exact Clique if $\omega > d = 2.1 \log n$)**.** *Note* $(x_1 + ... + x_n)^d = \omega^d$ *modulo the Size Axiom. The LHS can be multi-linearly homogenized to degree-d by* $x_S = \frac{1}{\omega - |S|} \sum_{i \notin S} x_{S \cup \{i\}}$ *by this axiom again, after which w.h.p. all terms are 0 by Clique Axioms, as there is no size-*$2.1 \log n$ *clique in* $G \sim G(n, 1/2)$ *w.h.p.. This gives the contradiction* $0 = 1$*. Note this proof is actually in the weaker Nullstellensatz system (for definition see e.g. [7]).*

[3] There is no "planted clique" in the problem's formulation, but traditionally, the problem is still called the planted clique problems due to the algorithmic motivation behind.

[4] One on the objective value of the CSP, and/or one on the Hamming weight of x.

1.3 Results of the paper

Our main result is the following.

▶ **Theorem 1.4.** *Let $\epsilon > 0$ be any parameter, $\omega = n^{1/2-\epsilon}$. W.p. $> 1 - n^{-4\log n}$ over $G \sim G(n, \frac{1}{2})$, any SOS refutation of Exact Clique requires degree at least $\epsilon' \log n/\log\log n$, where $\epsilon' = \min\{\epsilon^2, \frac{1}{40^2}\}/2000$.*

We also have the following result. It does not allow to improve the lower bound but provides a new, hopefully simplifying, perspective on certain techniques that were used for the non-exact problem.

▶ **Theorem 1.5** (Informal). *For the Non-Exact Clique problem,*

(1) *There is a way to define the correct pseudo-expectation from simple incidence algebra on the vertex-set;*

(2) *For the resulting moment matrix M, there is a weakened version of the quadratic equation $M = NN^\top$ whose solvability is given by, and actually equivalent to, a general graph-decomposition fact from which a "first-approximate" diagonalization of M can be deduced.*

2 Key technical ideas

The two results use almost completely different ideas, so we treat them separately in the proof overview:

- Theorem 1.4: Section 2.1 to 2.4.
- Theorem 1.5: Section 2.5.

The presentation of this section is structured for mathematical clarity. On the other hand, the following picture may provide a clearer bird's-eye view, where "$\cdots$" means the corresponding section(s) in the text:

Pseudo-expectation design: A common idea (in below)
→ Non-exact case (2.5 first half $\cdots$ 3.1)
→ Exact case (2.1 $\cdots$ 3.2).
Proving PSDness: Recursive factorization refresh (5.1, 5.3)
→ Lower bound proof (2.1 to 2.4 $\cdots$ 6).

And a "naturalizing" result that can be read independently:

How to deduce the "coarse" diagonalization (2.5 second half $\cdots$ 5.2).

Let's start with a common idea. Suppose we deal with degree-d SOS, $\omega = n^{1/2-\epsilon}$ where $\epsilon > 0$ is small. To construct pseudo-expectations on size $\le d$-subsets of $[n]$, as is usual in complexity theory, we take a parameter $\tau \gg d$ (think of $d \ll \tau \ll \log n$) and make the construction for all size $\le \tau$-subsets first, in hope to later have a good control on its behavior on size $\le d$ subsets. This idea is most clearly demonstrated in the nonexact case (Section 3.1.2) and is also inherited to the exact case, as we will see next (equation (2.1)).

2.1 The exact pseudo-expectation

We define the pseudo-expectation for Exact Clique now. To satisfy Size Constraints (1.4), a natural way is to generate $\widetilde{E}$ in a top-down fashion: fix $\widetilde{E}x_S$ for all $|S| = d$ first, denoted as the vector $\widetilde{E}_d x$, then recursively set

$$\widetilde{E}x_S \leftarrow \frac{1}{\omega - |S|} \sum_{i \notin S} \widetilde{E}x_{S \cup \{i\}} \quad \forall |S| < d.$$

The Clique Constraints (1.3) can be satisfied if $\widetilde{E}_d x_S$ factors through the clique function on S. Inspired by the non-exact case (Lemma 3.5), we use Fourier characters and consider

$$\widetilde{E}x_S = \sum_{T : |V(T) \cup S| \leq \tau} F(|V(T) \cup S|) \cdot \chi_T \quad \forall S : \ |S| = d \tag{2.1}$$

for some function F. We call F a ***d*-generating function**.[5] Thus

$$\widetilde{E}x_S = \frac{1}{\binom{w-d+u}{u}} \sum_{T : |V(T) \cup S| \leq \tau} \chi_T \cdot \Bigg[\sum_{c=0}^{u} \binom{|V(T) \cup S| - d + u}{c} \binom{n - |V(T) \cup S|}{u - c} \cdot F(|V(T) \cup S| + u - c) \Bigg]$$

where $u := d - |S|$, for all S with $|S| \leq d$.

One key novelty we bring is the following choice

$$F(x) = \frac{(x + 8\tau^2)!}{(8\tau^2)!} \cdot (\frac{\omega}{n})^x. \tag{2.2}$$

With this F, the resulting moment matrix, denoted by $\widetilde{M}$, is:

$$\widetilde{M}(A, B) = \sum_{T : |V(T) \cup A \cup B| \leq \tau} \widetilde{M}(A, B; T) \chi_T \quad \forall A, B : \ |A|, |B| \leq d/2$$

where $\widetilde{M}(A, B; T) =$

$$\frac{1}{\binom{\omega-d+u}{u}} \Bigg[\sum_{c=0}^{u} \binom{|V(T) \cup A \cup B| - (d - u)}{c} \binom{n - |V(T) \cup A \cup B|}{u - c} \cdot \underbrace{\frac{(|V(T) \cup A \cup B| + u - c + 8\tau^2)!}{(8\tau^2)!} \cdot (\frac{\omega}{n})^{|(V(T) \cup A \cup B)| + u - c}}_{F\left(|V(T) \cup A \cup B| + u - c\right)} \Bigg], \tag{2.3}$$

where $u = d - |A \cup B|$.

This seemingly mysterious choice of F is ultimately for proving PSDness of $\widetilde{M}$, which can be seen after a series of technical transformations (Remark 2.10, 3.12). It will be very interesting to know if there is *a priori* an explanation of it. See Remark 3.9, 6.14 for why the simpler, "traditional" choices from the literature, which simulate some plant-distributions, seem cannot work here.

[5] To be distinguished from the usual generating functions for sequences.

2.2 An Hadamard decomposition and Euler transform

For the Exact Clique problem, by a standard SOS homogeneity reduction (Lemma 4.1) it suffices to prove PSDness of the $\binom{[n]}{d/2} \times \binom{[n]}{d/2}$ principal minor of $\widetilde{M}$. We denote this minor by M.

One unpleasant feature of M is that in its expression (2.3), the parameter $u = |A \cap B|$ appears in a deeply nested way. To make a PSDness analysis on M (in particular, get a clue of how to diagonalize it), we resolve this intricacy by two steps. First,

$$M = \sum_{c=0}^{\frac{d}{2}} m_c \circ M_c \tag{2.4}$$

where m_c, M_c are matrices s.t. for all $|I|, |J| = d/2$,

$$m_c(I, J) = \frac{1}{\binom{\omega-d+u}{u}} \omega^{u-c} \quad \text{where } u \text{ denotes } |I \cap J|; \tag{2.5}$$

$$M_c(I, J) = \begin{cases} \sum\limits_{T: |V(T) \cup I \cup J| \le \tau} \chi_T \cdot M_c(|I \cap J|, |V(T) \cup I \cup J|), & \text{if } |I \cap J| \ge c; \\ 0, & \text{o.w.} \end{cases} \tag{2.6}$$

whose coefficients are

$$M_c(u, a) = \binom{a-(d-u)}{c}\binom{n-a}{u-c} n^{-(u-c)} \frac{(a+u-c+8\tau^2)!}{(8\tau^2)!} (\frac{\omega}{n})^a,$$

where $u = |I \cap J|$, $a = |V(T) \cup I \cup J|$. We will analyze m_c, M_c's separately.

The "harder" part is M_c. To further remove the dependence on $|I \cap J|$ in $M_c(I, J)$, our second step is to consider a decomposition

$$M_c = \sum_{R \in \binom{[n]}{\le \frac{d}{2}}} M_c^R \tag{2.7}$$

where for each R the matrix M_c^R is supported on rows and columns whose index contains R. To derive the expression of M_c^R, we use **Euler transform**: if $x(\cdot), y(\cdot)$ are two sequences defined on $\mathbb{N}$ s.t. $x(m) = \sum_{l=0}^{m} \binom{m}{l} y(l)$ for all m, then $x(\cdot)$ is called the *Euler transform* of $y(\cdot)$, and the inverse transform is given by $y(m) = \sum_{l=0}^{m} (-1)^{m-l} \binom{m}{l} x(l)$.

Apply the inverse Euler transform to $M_c(u, a)$ in the above[6] on u (fixing c, a), we get:

$$Y_c(r, a) = \begin{cases} \sum\limits_{l=c}^{r} (-1)^{r-l} \binom{r}{l} \binom{a+l-d}{c} \binom{n-a}{l-c} n^{-(l-c)} \frac{(a+l-c+8\tau^2)!}{(8\tau^2)!} & \text{, if } r \ge c; \\ 0 & \text{, o.w.} \end{cases} \tag{2.8}$$

In summary, the following lemma can be proved.

▶ **Lemma 2.1** (ΣΠ-decomposition of M).

$$M = \sum_{c=0}^{\frac{d}{2}} m_c \circ \left(\sum_{R \in \binom{[n]}{\le d/2}} M_c^R \right) = \sum_{R \in \binom{[n]}{\le d/2}} \left(\sum_{c=0}^{|R|} m_c \circ M_c^R \right) \tag{2.9}$$

where each m_c is by (2.5)*, and each M_c^R has the following expression.*

[6] A subtle but important point is that $M_c(u, a)$ is partial (i.e. defined only when $u \ge c,\ a - (d-u) \ge c$), and we need to extend it to $(u, a) \in \mathbb{N}^2$ – see Def. 6.5.

1. *$M_c^R = 0$ if $|R| < c$;*
2. *If $R \not\subseteq I \cap J$, $M_c^R(I,J) = 0$;*
3. *If $|R| \geq c$ and $R \subseteq I \cap J$, then*

$$M_c^R(I,J) = \sum_{T: |V(T) \cup I \cup J| \leq \tau} M_c^R(I,J;T)\chi_T$$

where, if denote $a = |V(T) \cup I \cup J|$,

$$M_c^R(I,J;T) = (\frac{\omega}{n})^a \cdot Y_c(|R|, a) \quad (\textit{defined by } (2.8)). \tag{2.10}$$

4. *For all $0 \leq c \leq r \leq d/2$ and $0 \leq a \leq \tau$, $|Y_c(r,a)| < \tau^{5\tau}$.*

Intuition for analysis. The intuition behind decomposition (2.9) is that, the first factor m_c is decreases in c and m_0 is very positive; while for every fixed R, M_0^R is positive and other M_c^R's ($c > 0$) are not too large. This is expounded by the following two lemmas.

▶ **Lemma 2.2.** *For each $c = 0, ..., d/2$,*

$$m_0 = \omega m_1 = ... = \omega^{\frac{d}{2}} m_{\frac{d}{2}} \succ \frac{d}{2\omega}\mathrm{Id}. \tag{2.11}$$

▶ **Lemma 2.3** (Main Lemma)**.** *In decomposition (2.9), w.p. $> 1 - n^{-5\log n}$ the following hold. For all $R \in \binom{[n]}{\leq d/2}$, let $P^R = \{I \in \binom{[n]}{d/2} \mid R \subseteq I\}$, the following holds.*

(1)

$$M_0^R \succeq n^{-d}\mathrm{diag}(\widetilde{\mathrm{Cl}})_{P^R \times P^R}; \tag{2.12}$$

(2)

$$\pm\omega^{-c}M_c^R \preceq n^{-c/6} \cdot M_0^R, \quad \forall 0 < c \leq |R|. \tag{2.13}$$

These two lemmas immediately imply that $M(G) \succeq n^{-d-1}\mathrm{diag}(\widetilde{\mathrm{Cl}}(G))_{\binom{[n]}{d/2} \times \binom{[n]}{d/2}}$ w.h.p., and Theorem 1.4 is an easy corollary of this (Cor. 6.2, 6.10).

The proof of Lemma 2.2 is relatively easy using *Johnson schemes* (see Lemma 6.4). Below we show how to prove the Main Lemma.

2.3 Recursive factorization: an extension

Fix any c, R ($|R| \geq c$). To prove the Main Lemma, an important step is to derive an approximate diagonalization of M_c^R, where we will use the *recursive factorization* technique from [5]. This technique will be refreshed, formalized and extended properly for our use in Section 5.3.

For now, we give a **first-approximate factorization** of M_c^R then apply this technique to get a refined diagonalization by Lemma 2.6.

The next definition in full (Definition 6.11) mentions many terms about a graph-theoretic structure; we omit the details here.

▶ **Definition 2.4** (Side factors)**.** *Fix $R \in \binom{[n]}{\leq \frac{d}{2}}$. For $i = 0, 1, ..., \tau$ let $L^{R,i}$ be the matrix of dimension $\binom{[n]}{\frac{d}{2}} \times \binom{[n]}{\leq \frac{d}{2}}$ defined by equation (6.20) (the exact content is not important for now). Call $\widetilde{L^R} = (L^{R,0}, ..., L^{R,\tau})$ the **left factor**, and $(\widetilde{L^R})^\top$ the **right factor**.*

We use these factors to give a PSD factorization in the form $M^R = \widetilde{L^R}\,(-)\left(\widetilde{L^R}\right)^\top$. The starting point is a coarse, "first approximate" factorization. In the definition below, T_m simply means an edge-set and $\mathrm{mSep}_{A,B}(T_m)$ is the set of all *minimal separators* of vertex-sets A, B (Def. 4.10). Let D^τ be the diagonal matrix $\operatorname{diag}\left((\frac{\omega}{n})^{\frac{|A|}{2}}\right) \otimes \mathrm{Id}_{\{0,\dots,\tau\}\times\{0,\dots,\tau\}}$.

▶ **Definition 2.5.** *For any $R \in \binom{[n]}{\leq d/2}$ define the index set*

$$S^R = \{(A,i) \in \binom{[n]}{\leq d/2} \times \{0,...,\tau\} \mid A \supseteq R,\ |A| + i \geq \frac{d}{2}\}.$$

For $c = 0, ..., |R|$, define $Q^R_{c,0}$ to be the $\{0,...,\tau\} \times \{0,...,\tau\}$-blocked matrix, each block of dimension $\binom{[n]}{\leq d/2} \times \binom{[n]}{\leq d/2}$: it is supported on $S^R \times S^R$, expressed by $Q^R_{c,0}\Big((A,i),(B,j)\Big) =$

$$\sum_{\substack{T_m: |V(T_m)\cup A\cup B| \leq \tau \\ A,B \in \mathrm{mSep}_{A,B}(T_m)}} (\frac{\omega}{n})^{|V(T_m)\cup A\cup B| - \frac{|A|+|B|}{2}} \cdot \underbrace{Y_c\big(|R|,\ |V(T_m)\cup A\cup B| + (i+j)\big)}_{\text{defined by (2.8)}} \cdot \chi_{T_m} \quad (2.14)$$

*Then we call $\widetilde{L^R} \cdot \left(D^\tau \cdot Q^R_{c,0} \cdot D^\tau\right) \cdot \left(\widetilde{L^R}\right)^\top$ the **first approximate factorization** of M^R_c.*

▶ **Lemma 2.6** (Recursive approximate factorization; informal). *For any $R \in \binom{[n]}{\leq d/2}$ and $0 \leq c \leq |R|$, we have the following decomposition.*

$$M^R_c = \widetilde{L^R} \cdot \left[D^\tau \Big(Q^R_{c,0} - Q^R_{c,1} + ... \pm Q^R_{c,d}\Big) D^\tau\right] \cdot \left(\widetilde{L^R}\right)^\top + \mathcal{E}^R_c. \quad (2.15)$$

Here, all $Q^R_{c,k}$'s ($k = 0, 1, ..., d$) are supported within $S^R \times S^R$ with expression

$$Q^R_{c,k}\Big((A,i),(B,j)\Big) = \sum_{T_m: |V(T_m)\cup A\cup B| \leq \tau} q^R_{c,k}(\mathcal{R}_m, i, j) \cdot \chi_{T_m}$$

where $\mathcal{R}_m$ denotes the triple $(A, B; T_m)$, and the coefficients $q^R_{c,k}(\cdot, i, j)$'s depend only on the "shape" of $\mathcal{R}_m$, satisfying

$$|q^R_{c,k}(\mathcal{R}_m, i, j)| \leq \tau^{5\tau} \cdot (\frac{\omega}{n^{1-\epsilon}})^{s-p+k/3} \quad \forall (i,j) \quad (2.16)$$

where $s = \frac{|A|+|B|}{2}$, p is the max number of vertex-disjoint paths from A to B in $\mathcal{R}_m$.

Moreover, the "error" $\mathcal{E}^R_c(G)$ is supported within rows and columns that contains R and is clique in G, and w.p. $> 1 - n^{-9\log n}$, $\left\|\mathcal{E}^R_c\right\| < n^{-\epsilon\tau/2}$.

▶ Remark 2.7. In this factorization, the middle matrices Q's have a "tensored-dimension" with $(\tau + 1)$, i.e. it is a $(\tau + 1) \times (\tau + 1)$-blocked matrix, each block of dimension $\binom{[n]}{\leq d/2} \times \binom{[n]}{\leq d/2}$. This reflects a key difference (at least technically) between Exact Clique and the non-exact case; see Remark 6.14.

2.4 Proving PSDness: encounter with Hankel matrices

With Lemma 2.2 and the recursive factorization lemma 2.6 at hand, the following is the key step towards the Main Lemma.

▶ **Lemma 2.8.** *W.p.* $> 1 - n^{-8\log n}$ *over* G*, the following holds.*

(1) $\forall R \in \binom{[n]}{\le d/2}$,

$$Q^R_{0,0} - Q^R_{0,1} + \ldots \pm Q^R_{0,\frac{d}{2}} \;\succeq\; \tau^{-7\tau} \cdot \mathrm{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R \times S^R}$$

where recall $S^R = \{(A, i) \in \binom{[n]}{\le d/2} \times \{0, \ldots, \tau\} \mid A \supseteq R, |A| + i \ge \frac{d}{2}\}$.

(2) $\forall R$, $0 < c \le |R|$

$$\pm\omega^{-c}\left(Q^R_{c,0} - Q^R_{c,1} + \ldots \pm Q^R_{c,\frac{d}{2}}\right) \;\preceq\; n^{-c/4} \cdot \mathrm{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R \times S^R}.$$

To prove this lemma, modulo somewhat standard steps (three Lemmas 6.34, 6.37, 6.38) the final technical challenge is:

Show the positiveness of $\mathbb{E}[Q^R_{0,0}]$ (Corollary 6.30).

We describe below how this is done. After simplification, the real task is to analyze the positiveness of the following matrix[7]:

$$\sum_{l=0}^{r} (-1)^{r-l} \frac{\binom{r}{l}}{l!} \cdot H_{\tau,\ l+8\tau^2} \quad \text{for any } 0 \le r \le d/2 \tag{2.17}$$

where $\{H_{m,t}\}$ is the family of $(m+1) \times (m+1)$-matrices

$$H_{m,t}(i, j) = (i + j + t)! \quad \forall 0 \le i, j \le m.$$

This is a special family of the so-called *Hankel matrices* whose (i, j)th element depends only on $i + j$. General Hankel matrices seem to arise naturally in moment problems but they are notoriously wild-behaving in many aspects (see e.g. [31]). Fortunately enough, for the special family here we can manage to get a relatively fine understanding; we term this family **factorial Hankel matrices**. The key observation is that they have a concrete *recursive diagonalization* (Proposition 6.27), resulting in the following property.

▶ **Proposition 2.9.** *If parameters* m, t, r *satisfy*

$$t + 1 > 8 \cdot \max\{r^2, m\}, \tag{2.18}$$

then $H_{m,t+1} \succeq 2r^2 H_{m,t}$.

▶ Remark 2.10. The condition (2.18) in the above proposition is the reason of the "$8\tau^2$" in the numerator of F, (2.2).

With this proposition, it is relatively easy to complete the proof of the Lemma 2.8, hence the Main Lemma. This completes the proof overview of Theorem 1.4.

2.5 Ideas for Theorem 1.5

In this subsection, we demonstrate how to "naturalize" certain techniques that were used for the lower bounds of Non-Exact Clique.

[7] The subscripts are not exactly as in the problem but suffice to demonstrate the spirit.

On defining the pseudo-expectation. (Section 3.1) Previously, the pseudo-expectation is obtained via the so-called *pseudo-calibration* method. We show how to define the same $\widetilde{E}$ in very different terms via the incidence algebra on the vertex-set, which can also be regarded as a simple refinement of the construction in [13].

The ζ*-matrix* on $[n]$ is the $2^{[n]} \times 2^{[n]}$ 0-1 matrix with $\zeta(A,B) = 1$ iff $A \subseteq B$. We observe that ζ reveals the basic linear structure of the true expectation on cliques in the case of a single planted clique, and we use ζ to define $\widetilde{E}$. That is, we define a *degree-*τ approximate-distribution vector $p_\tau(G)$ first – it approximates the real planted-clique distribution, with a standard twist so as to be supported on cliques in G (3.8) – then take the vector $\zeta_{d,\tau} \cdot p_\tau(G)$ as $\widetilde{E}x$ (Def. 3.3). Here, $(\cdot)_\tau$ means to truncate the matrix or vector to indices whose size $\leq \tau$. In this way, $\widetilde{E}$ inherits the linear structure posed by ζ too.

On deducing the first-approximate diagonalization. (Section 5) We deduce a "coarse" diagonalization of the resulting moment matrix from $\widetilde{E}$ in above. The deduction has two steps: 1. Analyze the expectation of the matrix; 2. The (imaginary) diagonalization of the matrix is in essence a quadratic equation, which we weaken to a proper "modular" version and solve the latter. We call step 2 the **mod-order analysis** (Section 5.2), whose underlying idea is inspired by and similar to the more broad dimension-analysis in physical sciences: weaken the equation to its most significant part in a well-defined way (Def. 5.5). One ingredient towards defining the weakening is the norm information on certain pseudo-random matrices (the *graphical matrices*).

The resulting weakened equation has a nice structure to work with (Lem. 5.6, Cor. A.2). Using standard techniques for studying algebraic equations – actually a simple *polarization* (Appendix A.2) – we can deduce a solvability condition for the polarized equation, which translates to the existence of a general graph-theoretic structure (equation (A.19) and Fact A.1). The "coarse" diagonalization is then formulated based on this structure.

To demonstrate this equation in more detail, it suffices to concentrate on the $\binom{[n]}{d/2} \times \binom{[n]}{d/2}$-minor of the moment matrix, denoted by M':

$$M'(I,J) = \sum_{T:|V(T)\cup I\cup J|\leq\tau} (\frac{\omega}{n})^{|V(T)\cup I\cup J|}\chi_T, \quad \forall I,J: \ |I| = |J| = d/2.$$

Step 1: expectation. By using *Johnson schemes* as in [25], we get an explicit decomposition $\mathbb{E}[M'] = CC^\top$ where C is $\binom{[n]}{d/2} \times \binom{[n]}{\leq d/2}$, and actually with a fine understanding of the spectrum of $\mathbb{E}[M']$.

Step 2: mod-order analysis. Given $\mathbb{E}[M'] = CC^\top$ from Step 1, ideally we hope to solve the quadratic matrix equation

$$M' = NN^\top \tag{2.19}$$

in N with $\mathbb{E}[N] = C$, and N extending C by non-trivial Fourier characters. Two observations about (2.19) follow.

(1) Order in $\frac{\omega}{n}$. Entries of M' all have a clear order in $\frac{\omega}{n}$. Like in fixed-parameter problems, we treat $\frac{\omega}{n}$ as a distinguished structural parameter and try to solve the correct power of $\frac{\omega}{n}$ in N first.

(2) Norm-match. A closer look into $CC^\top$ shows

$$\|C_rC_r^\top\| \approx \binom{d/2}{r} \cdot (\frac{\omega}{n})^{d-r}n^{d/2-r}, \quad r = 0, ..., d/2, \tag{2.20}$$

where assume $C = (C_0, ..., C_{d/2})$, each C_r having column dimension $\binom{[n]}{r}$. Assume $N = (N_0, ..., N_{d/2})$. Then we expect $N_r N_r^\top$ to concentrate around $C_r C_r^\top$ for each r, and so expect the norm of the non-constant part of $N_r N_r^\top$ to be bounded by (2.20). Under this expectation, the known tight norm bounds on related matrix pieces would tell us, for each possible appearing term in N, the least order of $\frac{\omega}{n}$ in its coefficient.

With these observations, we can weaken equation (2.19) to a simple "modular version" that is more informative about the (imaginary) solution N. Namely, abstract $(\frac{\omega}{n})$ as a fresh variable α and work in ring $\mathbb{R}[\alpha, \{\chi_T\}]$, consider

$$(M' \mod \text{high order}) = (N \mod \text{high order}) \cdot (N^\top \mod \text{high order}) \tag{2.21}$$

where "order" means power of α (think of α as an "infinitesimal"). We call (2.21) the *mod-order equation* and its analysis the *mod-order analysis*. For details see Definition 5.5.

We feel that this approach leads us more naturally to the realization of using the graph-theoretic structure beyond guesses, and the simple general idea behind the mod-order analysis might hopefully find other applications.

2.6 Structure of the paper

In Section 3 we define the pseudo-expectations and show Theorem 1.5(1). In Section 4 we recall some fundamental tools for analysis. In Section 5 we refresh the technique of recursive factorization and show Theorem 1.5(2). With all preparations in place, in Section 6 we prove the main Theorem 1.4. The paper is concluded in Section 7 with open problems.

Notation. I, J, A, B, S will be used to denote vertex-sets, and T for edge-sets. $E(S) := \binom{S}{2}$. G denotes a simple graph on the vertex-set $[n]$. "$T \subseteq E([n])$" will be omitted in summation when there is no confusion. Finally, we use $y(n) = O(x(n))$ to mean that there is some absolute constant c s.t. $y(n) \leq cx(n)$ for all n.

Parameter regime. Throughout the paper,

$$\begin{aligned}
\epsilon &= \text{any positive parameter (wlog } \epsilon < \frac{1}{40});\\
\omega &= n^{1/2-4\epsilon};\\
\tau &= \frac{\epsilon}{200} \log n / \log\log n;\\
d &= \frac{\epsilon}{100}\tau.
\end{aligned}$$

3 Pseudo-expectations

In this section, we define the pseudo-expectations. As a warm-up we start with the non-exact problem, then move on to the exact case.

3.1 Non-exact case: a new perspective

Given a graph G we can think of a degree-d pseudo-expectation as assigning a number $\widetilde{Ex}_S$ to each subset $S \subseteq [n]$ of size $\leq d$, so that the resulting vector $\widetilde{Ex}$ *looks indistinguishable* to the expectation resulted from the case when a random-ω clique is planted, from the view of degree-d SOS.

As explained at the beginning of Section 2, to make up such an assignment we first go beyond to slightly larger subsets of size τ. We define an "approximate distribution" on size $\leq \tau$-cliques in G then use it to generate pseudo-expectation on all size $\leq d$-subsets.

3.1.1 ζ-function and Möbius inversion

Given n-vertex graph G, let $p(G) \in \mathbb{R}^{2^{[n]}}$ be the max-clique-indicator vector, then

$$q(G) := \zeta \cdot p(G)$$

is a vector supported exactly on all cliques in G, where ζ is the $2^{[n]} \times 2^{[n]}$ matrix

$$\zeta(A, B) = 1 \text{ iff } A \subseteq B, \quad \forall A, B \subseteq [n]. \tag{3.1}$$

In particular, if G itself is a single clique then $q(G)$ is the clique-indicator. We will use $\zeta_{a,b}$ to denote the submatrix of ζ on rows $\binom{[n]}{\leq a}$ and columns $\binom{[n]}{\leq b}$, and use similar notation on all related vectors.

Consider the plant-situation where G is indeed a single random clique. Suppose its distribution is represented by a *plant-distribution* vector $p_{\text{plant}} \in \mathbb{R}^{2^{[n]}}$. Let the *output-expectation* q_{out} be indicator-vector of cliques in G in expectation. Then

$$q_{\text{out}} = \zeta \cdot p_{\text{plant}}. \tag{3.2}$$

We call such a pair $(p_{\text{plant}}, q_{\text{out}})$ a **plant-setting**.

▶ **Definition 3.1** (Two plant-settings). *The exact plant-setting* (p_0, q_0) *is:*

$$p_0(S) = \frac{1}{\binom{n}{\omega}} \text{ if } |S| = \omega \quad \text{and } 0 \text{ otherwise}, \tag{3.3}$$

and

$$q_0(S) = (\zeta p_0)(S) = \frac{\binom{n-|S|}{\omega-|S|}}{\binom{n}{\omega}}. \tag{3.4}$$

I.e. in this setting a random size-ω subset is chosen to be the planted clique.

The independent plant-setting (p_1, q_1) *is:*

$$p_1(S) = (\frac{\omega}{n})^{|S|}(1 - \frac{\omega}{n})^{n-|S|} \quad \forall S \subseteq [n], \tag{3.5}$$

and

$$q_1(S) = (\zeta p_1)(S) = (\frac{\omega}{n})^{|S|}. \tag{3.6}$$

I.e. any vertex is included in the planted clique w.p. $\frac{\omega}{n}$ independently.

Thus the matrix ζ reveals the basic linear relations between $(p_{\text{plant}}, q_{\text{out}})$. It is upper-triangular (with row- and column-indices ordered in a size-ascending way), invertible, and the inverse is the *Möbius inversion* matrix:

$$\zeta^{-1}(A, B) = (-1)^{|B \setminus A|} \text{ if } A \subseteq B, \text{ and } 0 \text{ otherwise}.$$

Note $(\zeta_{a,a})^{-1} = (\zeta^{-1})_{a,a}$ for all $a \leq n$. Moreover, if let the pseudo-expectation be defined as $\widetilde{E}x = p \in \mathbb{R}^{2^{[n]}}$ for some vector p, then the "full" $2^{[n]} \times 2^{[n]}$ moment matrix is

$$M_{SOS} = \zeta \text{diag}(p) \zeta^{\top}. \tag{3.7}$$

In particular, if p is a nonnegative vector then M_{SOS} is immediately PSD.

3.1.2 The non-exact pseudo-expectation

Idea. Given any G, we will first construct a *degree-τ* "approximate plant-distribution" $p_\tau(G)$, which simulates the plant-distribution (Def. 3.1) in the sense that they give similar output-expectations. We also require $p_\tau(G)$ to be supported on size $\le \tau$-cliques in G. Then we can take $\widetilde{E}x = \zeta_{d,\tau} \cdot p_\tau(G)$ so that the result inherits the linear structure posed by ζ.

What is this $p_\tau(G)$? From the view of approximation it seems taking $\zeta^{-1}_{\tau,\tau}(q_1)_\tau$ would suffice, while to make it supported on cliques, same as in [13] we add a clique-indicator factor:

$$p_\tau(G)(S) = \left(2^{|\binom{S}{2}|}\mathrm{Cl}_S(G) \cdot \zeta^{-1}_{\tau,\tau}(q_1)_\tau\right)(S) \quad \forall S \subseteq [n] \text{ of size } \le \tau \tag{3.8}$$

where $\mathrm{Cl}_S(\cdot)$ is the clique indicator function and $2^{|\binom{S}{2}|}$ is for re-normalization.

▶ **Definition 3.2.** *$\forall S \subseteq [n]$, the **normalized clique-indicator** is function*

$$\widetilde{\mathrm{Cl}}_S(G) := 2^{|\binom{S}{2}|}\mathrm{Cl}_S(G). \tag{3.9}$$

$\widetilde{\mathrm{Cl}}(G)$ denotes the (column) vector of them over a family of S's, which will always be clear from the context.

▶ **Definition 3.3.** *The **non-exact pseudo-expectation** is*

$$\widetilde{E}_{\text{nonexact}} = \zeta_{d,\tau} \cdot p_\tau(G) = \zeta_{d,\tau} \cdot (\widetilde{\mathrm{Cl}}(G) \circ \zeta^{-1}_{\tau,\tau}) \cdot (q_1)_\tau \quad \in \mathbb{R}^{\binom{[n]}{\le d}} \tag{3.10}$$

where "$\circ$" is the Hadamard product[8]*.*

In short, $\widetilde{E}_{\text{nonexact}}$ refined the construction in [13] by one step: factor through size-τ subsets (in the *only* non-trivial way) so that the size-d output inherits linear relations posed by ζ.

The resulting moment matrix is

$$M_{\text{nonexact}}(G) = \zeta_{d/2,\tau} \cdot \mathrm{diag}\left(p_\tau(G)\right) \cdot \left(\zeta_{d/2,\tau}\right)^\top, \tag{3.11}$$

similarly as (3.7).

▶ Remark 3.4. $\widetilde{E}_{\text{nonexact}}$ looks like a true expectation on cliques in G, namely, if $p_\tau(G)$ were nonnegative then the PSDness of $M_{\text{nonexact}}(G)$ would be immediate. Alas, this is not true by computation[9]. That the PSDness could still possibly hold is because $\zeta_{d/2,\tau}$ in (3.11) is degenerate.

▶ **Lemma 3.5** (Theorem 1.5(1)). *For all $S \subseteq [n]$ s.t. $|S| \le d$,*

$$\widetilde{E}_{\text{nonexact}} x_S = \sum_{T:|V(T)\cup S| \le \tau} \left(\frac{\omega}{n}\right)^{|V(T)\cup S|} \chi_T. \tag{3.12}$$

Proof. Note $\widetilde{\mathrm{Cl}}_S = \sum_{T \subseteq E(S)} \chi_T$ for all S. Now for S, S' with appropriate size bound,

$$\left(\widetilde{\mathrm{Cl}} \circ \zeta^{-1}_{\tau,\tau}\right)(S, S') = \begin{cases} \sum_{T \in E(S)} \chi_T \cdot (-1)^{|S' \setminus S|}, & \text{if } S \subseteq S' \\ 0, & \text{o.w.} \end{cases};$$

[8] In general $(M_1 \circ M_2) \cdot M_3 \neq M_1 \circ (M_2 \cdot M_3)$, but they are equal if M_1 is a column vector.

[9] One intuition, suggested by a reviewer, is that any true expectation on cliques has objective value $\sum_{i=1}^n x_i = O(\log n)$ w.h.p.. Now if $p_\tau(G)$ were nonnegative then it would be almost a distribution since $\widetilde{E}_{\text{nonexact}}(x_\phi) \approx 1$ (which is not too hard to check by (3.12)), but its objective value $n^{\frac{1}{2}-\epsilon}$ is too big.

$$\begin{aligned}\left(\zeta_{d,\tau}\cdot(\widetilde{\mathrm{Cl}}\circ\zeta^{-1}_{\tau,\tau})\right)(S,S') &= \sum_{S'':S\subseteq S''\subseteq S'}\left(\sum_{T\subseteq E(S'')}\chi_T\cdot(-1)^{|S'\setminus S''|}\right)\\ &= \sum_{T:V(T)\cup S\subseteq S'}\chi_T\cdot\left(\sum_{S'':V(T)\cup S\subseteq S''\subseteq S'}(-1)^{|S'\setminus S''|}\right)\\ &= \sum_{T:V(T)\cup S\subseteq S'}\chi_T\cdot\delta_{S'=V(T)\cup S} = \sum_{T:V(T)\cup S=S'}\chi_T.\end{aligned}$$

Therefore, $\widetilde{E}_{\mathrm{nonexact}}x_S =$

$$\begin{aligned}\left(\zeta_{d,\tau}\cdot(\widetilde{\mathrm{Cl}}\circ\zeta^{-1}_{\tau,\tau})(q_1)_\tau\right)(S) &= \sum_{S':|S'|\leq\tau}\left(\sum_{T:V(T)\cup S=S'}\chi_T\cdot(\frac{\omega}{n})^{|S'|}\right)\\ &= \sum_{T:|V(T)\cup S|\leq\tau}\chi_T\cdot(\frac{\omega}{n})^{|V(T)\cup S|}\end{aligned}$$

for all S with $|S|\leq d$. ◀

3.2 The exact case

In this subsection, we give a generic way to generate possible pseudo-expectations that satisfy Size Constraints (1.4). The idea is to define $\widetilde{E}x_S$ in a top-down fashion: fix $\widetilde{E}x_S$ for all $|S|=d$ first, then recursively set

$$\widetilde{E}x_S \leftarrow \frac{1}{\omega-|S|}\sum_{i\notin S}\widetilde{E}x_{S\cup\{i\}} \tag{3.13}$$

for smaller-sized S's. If denote by $\widetilde{E}_d x$ the vector of the assignments for S's s.t. $|S|=d$, then this amounts to multiplying $\widetilde{E}_d x$ by the following matrix.

▶ **Definition 3.6.** *The d-**filtration matrix** $\mathrm{Fil}_{d,=d}$, of dimension $\binom{[n]}{\leq d}\times\binom{[n]}{d}$, is:*

$$\mathrm{Fil}_{d,=d}(A,B) = \begin{cases}\binom{\omega-|A|}{d-|A|}^{-1}, & \textit{if } A\subseteq B \textit{ (where } |B|=d\textit{);}\\ 0, & \textit{otherwise.}\end{cases} \tag{3.14}$$

▶ **Definition 3.7.** *Given vector $\widetilde{E}_d x$ which assigns a value to each d-subset $S\subseteq[n]$, the **exact pseudo-expectation generated by $\widetilde{E}_d x$** is*

$$\widetilde{E}x := \mathrm{Fil}_{d,=d}\cdot\widetilde{E}_d x. \tag{3.15}$$

▶ **Lemma 3.8.** *The pseudo-expectation in Definition 3.7 satisfies the Size Constraints* (1.4)*, regardless of the choice of $\widetilde{E}_d x$.*

Proof. For any $S\in\binom{[n]}{<d}$, take a vector $v_S\in\mathbb{R}^{\binom{[n]}{\leq d}}$

$$v_S(S') = \begin{cases}\omega-|S|, & \text{if } S'=S;\\ -1, & \text{if } S'\supseteq S \text{ and } |S'\setminus S|=1;\\ 0, & \text{otherwise}\end{cases}$$

then it suffices to show $v_S^\top\mathrm{Fil}_{d,=d}=0$. But this is a direct check. ◀

The $\widetilde{E}$ generated like so should further satisfy:

1. Clique Constraints (1.3);
2. PSDness Constraint (1.5);
3. Default Constraint (1.2) (so far we only have $\omega \cdot \widetilde{E}x_\emptyset = \widetilde{E}x_1 + ... + \widetilde{E}x_n$).

Item 3 is not a problem as long as $\widetilde{E}x_\emptyset > 0$, since we can always rescale everything by $(\widetilde{E}x_\emptyset)^{-1}$ without affecting other constraints.

▶ Remark 3.9 (Example). The following construction seems natural. Combining Def. 3.7 with the perspective from Section 3.1.2, we can take (3.10) with the exact plant-setting (p_0, q_0), followed by multiplying $\mathrm{Fil}_{d,=d}$:

$$\widetilde{E}_{\text{example}} x_S = \mathrm{Fil}_{d,=d} \cdot \left(\zeta_{d,\tau} \cdot (\widetilde{\mathrm{Cl}}(G) \circ \zeta^{-1}_{\tau,\tau}) \cdot (q_0)_\tau \right).$$

Actually, it can be easily checked that it satisfies Clique Constraints; it also has a nice expression in Fourier characters. By some computation which we omit here, modulo provably negligible error the resulting matrix is

$$M_{\text{example}}(I,J) = \sum_{\substack{T: \\ |V(T)\setminus(I\cup J)| \le \tau - d}} \chi_T \cdot \frac{\binom{n-|V(T)\cup I\cup J|}{\omega-|V(T)\cup I\cup J|}}{\binom{n}{\omega}}.$$

The only problem, however, is that we don't know how to prove the PSDness. Despite a transparent similarity to the previous expression (3.12), a similar proof breaks down seriously here, and the main reason is the loss of nice arithmetic structure when changing from function $(\frac{\omega}{n})^x$ to $\frac{\binom{n-x}{\omega-x}}{\binom{n}{\omega}}$. See also Remark 6.14.

3.3 The exact pseudo-expectation

Now we pinpoint an exact pseudo-expectation in Definition 3.7. With the idea stated in detail in the overview (Section 2.1), we give the construction directly.

We take the pseudo-expectation for $|S| = d$ in the form

$$\widetilde{E}x_S = \sum_{T: |V(T)\cup S| \le \tau} \chi_T \cdot F(|V(T) \cup S|)$$

for some function F. F is called a ***d*-generating function**. Then for general $|S| \le d$, (3.14) gives:

$$\begin{aligned}\widetilde{E}x_S = \frac{1}{\binom{w-d+u}{u}} \sum_{T: |V(T)\cup S| \le \tau} \chi_T \cdot \Bigg[\sum_{c=0}^{u} \binom{|V(T)\cup S| - d + u}{c} \binom{n - |V(T)\cup S|}{u-c} \\ \cdot F(|V(T)\cup S| + u - c) \Bigg]\end{aligned} \tag{3.16}$$

where we have let $u := d - |S|$.

▶ **Lemma 3.10.** *Any exact pseudo-expectation generated by* (3.16) *satisfies the Clique and Size Constraints* (1.3),(1.4).

Proof. For Clique Constraints, note (3.16) only depends on $|V(T) \cup S|$, so by grouping terms $\widetilde{E}x_S = \sum_{T:|V(T)\cup I\cup J| \le \tau} M(I,J;T)\chi_T$ factors through $\widetilde{\mathrm{Cl}}_{I\cup J} = \sum_{T \subseteq E(I\cup J)} \chi_T$. I.e., $M(I,J)(G) = 0$ if $\widetilde{\mathrm{Cl}}_{I\cup J}(G) = 0$.

It satisfies Size Constraints by Lemma 3.8. ◀

Now we pinpoint a choice of the d-generating function.

▶ **Definition 3.11** (Exact d-generating function)**.**

$$F(x) := \frac{(x+8\tau^2)!}{(8\tau^2)!} \cdot (\frac{\omega}{n})^x.$$

▶ Remark 3.12. As is said in the proof overview, the design of F, especially its first factor, is technical and the ultimate goal is to make the resulting M positive. The numerator $(x+8\tau^2)!$ will be used in Proposition 6.28, where the term $8\tau^2$ can be replaced by larger polynomials in τ. The $(8\tau^2)!$ in denominator is added just for convenience (see Remark 3.14).

▶ **Definition 3.13.** *The* ***exact moment matrix*** $\widetilde{M}$ *is*

$$\widetilde{M}(A,B) = \sum_{T:|V(T)\cup A\cup B|\le\tau} \widetilde{M}(A,B;T)\chi_T \quad \forall |A|,|B| \le d/2$$

where $\widetilde{M}(A,B;T) =$

$$\frac{1}{\binom{\omega-d+u}{u}} \Bigg[\sum_{c=0}^{u} \binom{|V(T)\cup A\cup B| - (d-u)}{c} \binom{n-|V(T)\cup A\cup B|}{u-c} \\ \cdot \underbrace{\frac{\big(|V(T)\cup A\cup B| + u - c + 8\tau^2\big)!}{(8\tau^2)!} \cdot (\frac{\omega}{n})^{|(V(T)\cup A\cup B)|+u-c}}_{f\big(|V(T)\cup A\cup B|+u-c\big)}\Bigg] \tag{3.17}$$

and where $u = d - |A\cup B|$.

▶ Remark 3.14. In (3.17), the “most significant” factor is $(\frac{\omega}{n})^{|V(T)\cup A\cup B|} \cdot \omega^{-c}$, if notice $\frac{\binom{n-|V(T)\cup A\cup B|}{u-c}}{\binom{\omega-d+u}{u}}\omega^u n^{-(u-c)}$ has 0th-order in ω, n. One thing to keep in mind is that other factors like $\frac{(|V(T)\cup A\cup B|+u-c+8\tau^2)!}{(8\tau^2)!}$ are qualitatively smaller than ω, within our parameter regime.

4 Preparations

In this section, we prepare some necessary tools for studying the matrices.

4.1 Homogenization for Exact Clique

With the Size Constraints (1.4) satisfied, any moment matrix can be reduced to its $\binom{[n]}{d/2}$-principal minor, which is slightly more convenient to work with. The following homogeneity trick is standard in the SOS literature.

Given any degree-d moment matrix $M_{dSOS}(G)$ that satisfies the Size Constraints (1.4), let $M(G)$ be its principal minor on $\binom{[n]}{d/2} \times \binom{[n]}{d/2}$.

▶ **Lemma 4.1.** $M_{dSOS}(G)$ *is PSD* $\Leftrightarrow$ $M(G)$ *is PSD.*

Proof. The $\Rightarrow$ part is trivial. Now suppose M_{dSOS} is not PSD, then

$$\exists a \in \mathbb{R}^{\binom{[n]}{\le d/2}} \quad a^\top M_{dSOS} a = -1. \tag{4.1}$$

With the presence of boolean constraints (i.e. define $\widetilde{E}(x_i^2 \cdot p) := \widetilde{E}(x_i \cdot p)$ for all i and all polynomial p of degree $\leq d-2$), this is equivalent to

$$\widetilde{E}(g^2) = -1 \tag{4.2}$$

where $g = a^\top x = \sum_{|S| \leq d/2} a_S x_S$ is multi-linear. Now substitute every x_S ($|S| < d$) in g by the corresponding linear combination of $\{x_{S'} \mid |S'| = d\}$ from (3.13). This does not affect the value of (4.2) since $\widetilde{E}$ satisfies the equality constraints. We get

$$\widetilde{E}(g_1^2) = -1 \tag{4.3}$$

for some multi-linear, degree-$d/2$ homogeneous g_1. Now translate (4.3) back (assume $g_1 = b^T x$, $x = (x_S)_{|S|=d/2}$) to $b^\top M b = -1$, we see that M is not PSD. ◀

4.2 Concentration bound on polynomials

The following is standard.

▶ **Lemma 4.2.** *Suppose $a < \log n$, and p is a polynomial*

$$p = \sum_{T:\ |V(T)|=a} c(T)\chi_T \quad c_T \in \mathbb{R}$$

and $C > 0$ is a number s.t. $|c(T)| \leq C$ for all T. Then W.p. $1 - n^{-10\log n}$ over G,

$$|p(G)| < C \cdot n^{a/2} 2^{a^2} n^{4 \log\log n}. \tag{4.4}$$

Proof. Power-estimation. For all $k \in \mathbb{N}$, (we can think of $a < k = o(n/a)$)

$$p^{2k} = \sum_{T_1,\ldots,T_{2k}:\ |V(T_i)|=a} c(T_1)\ldots c(T_{2k})\chi_{T_1} \cdot \ldots \cdot \chi_{T_{2k}} \tag{4.5}$$

Take expectation of (4.5). Each $\mathbb{E}[\chi_{T_1}\ldots\chi_{T_{2k}}(G)] \neq 0$ (i.e. equals 1) iff every edge appears even times in $T_1, \ldots, T_{2k}$, which implies $|V(T_1 \cup \ldots \cup T_{2k})| \leq \frac{1}{2} \cdot 2ka = ka$. There are at most $ka\binom{n}{ka} < n^{ka}$ many choices of such $V(T_1 \cup \ldots \cup T_{2k})$. For each choice, there are in turn at most $\binom{ka}{a} \cdot 2^{\binom{a}{2}} < (ka)^a \cdot 2^{a^2/2}$ many ways to choose each T_i. Therefore,

$$\mathbb{E}[p^{2k}] \leq C^{2k} \cdot n^{ka} \left((ka)^a 2^{a^2/2}\right)^{2k} := N^{2k} \quad \text{where} \quad N = Cn^{a/2} \cdot (ka)^a \cdot 2^{a^2/2}.$$

By Markov inequality, $\Pr\left[p^{2k} > (2N)^{2k}\right] < 2^{-2k}$. Take $k := 10\log^2 n$, we get that w.p. $> 1 - n^{-10\log n}$,

$$|p(G)| < 2N < C \cdot n^{a/2} 2^{a^2} n^{4 \log\log n}$$

for all large enough n. ◀

4.3 Norm concentration of pseudo-random matrices

Now we state a concentration bound on pseudo-random matrices which, like in almost all previous work on the subject, will be a fundamental tool for us.

The pseudo-random matrices refer to the *graphical matrices* ([24]). Intuitively, such a matrix collects Fourier characters of all embeddings of a fixed "shape". Definition 4.3, 4.5 below are implicit in [24, 25, 16] and is termed explicitly in [5].

▶ **Definition 4.3.** *A* ***ribbon*** $\mathcal{R}$ *is a (ordered) triple* $(A, B; T)$ *where* A, B *are vertex-sets and* T *is an edge set.* A, B *are called the left and right vertex set of* $\mathcal{R}$*. The* ***size*** *of* $\mathcal{R}$ *is*

$$|V(\mathcal{R})| = |V(T) \cup A \cup B|.$$

By definition, a ribbon $\mathcal{R} = (A, B; T)$ as a graph always has no isolated vertex outside of $A \cup B$.

▶ **Definition 4.4.** *We say* $\mathcal{R} = (A, B; T)$ *is* ***left-generated*** *if every vertex in* $V(\mathcal{R})$ *is either in* B *or can be reached by paths*[10] *from* A *without touching* B*. Being* ***right-generated*** *is symmetrically defined.*

▶ **Definition 4.5.** *For ribbon* $(A, B; T)$*, if further* $A \cup B$ *is totally-ordered, it is called a* ***shape****. Denote a shape by* $\mathcal{U} = (A, B; T)$*. As before,* $V(\mathcal{U}) = A \cup B \cup V(T)$*, and its* ***size*** *is* $|V(\mathcal{U})|$*.*

When fixing an underlying vertex-set $[n]$, a ribbon $\mathcal{R}$ within vertex set $[n]$ can always be regarded as shapes, with the induced ordering on vertices. So in this setting, we may speak of the shape of $\mathcal{R}$ and interchangeably use $\mathcal{R}$ to denote shapes.

▶ **Definition 4.6.** *A real-valued function* f *defined on a set of ribbons within vertex-set* $[n]$ *is called* ***symmetric with respect to shapes****, if whenever* $\mathcal{R}$ *and* $\mathcal{R}'$ *are of the same shape then* $f(\mathcal{R}) = f(\mathcal{R}')$*.*

▶ **Definition 4.7** ([24])**.** *Fix an* n*, and a shape* $\mathcal{U} = (A, B; T)$ *Define the* ***graphical matrix*** *of shape* $\mathcal{U}$ *to be the following* $2^{[n]} \times 2^{[n]}$*-matrix* $M_{\mathcal{U}}$*. Call a map* $\phi : V(\mathcal{U}) \to [n]$ proper *if* ϕ *is injective and respects the order on* $A \cup B$*, then*

$$\forall I, J \subseteq [n], \quad M_{\mathcal{U}}(I, J) = \sum_{\substack{T:\ \exists proper\ \phi\ s.t.\\ \phi(A)=I, \phi(B)=J, \phi(T)=T'}} \chi_{T'}$$

($= 0$ *if no such* ϕ *exists). Here,* ϕ *on* T *means the natural induced map on edges.*

▶ **Theorem 4.8** (Norm bounds on $M_{\mathcal{U}}$ [24, 5])**.** *For any shape* $\mathcal{U} = (A, B; T)$ *of size* $t < \log n$*, w.p.* $> 1 - n^{-10 \log n}$ *over* G*,*

$$\|M_{\mathcal{U}}(G)\| \le n^{\frac{t-p}{2}} \cdot 2^{O(t)} \cdot (\log n)^{O(t+p-2r)} \tag{4.6}$$

where $r = |A \cap B|$*,* p *is the maximum number of vertex-disjoint paths between* (A, B) *in* $\mathcal{U}$*. Moreover, this bound is tight up to polylog*(n)*-factors, for all* $M_{\mathcal{U}}$ *with the described parameters ([24], Thm 38]).*

Moreover, under the same notation, if further denote $s = \frac{|A|+|B|}{2}$ *then*

$$\|M_{\mathcal{U}}(G)\| \le n^{\frac{t-p}{2}} \cdot 2^{O(t)} \cdot (\log n)^{O(t-s)}. \tag{4.7}$$

Theorem 4.8 is proved by a careful estimation of the trace-power $\mathbb{E}[\mathrm{tr}(M_{\mathcal{U}}^{2k})]$ (for some $k > 0$), which we omit here. Its "moreover" part follows from (4.6) since $t \ge |A \cup B| = 2s - r$, $p \le s$, so

$$t + p - 2r \le t + s - 2(2s - t) = 3(t - s).$$

▶ Remark 4.9. Theorem 4.8 and its proof is a far-reaching generalization of that of the concentration bounds on polynomials, Lemma 4.2. Namely, if take special shapes in the form $\mathcal{U} = (A, A; T)$, then the corresponding matrix $M_{\mathcal{U}}$ is diagonal, so estimating its norm is equivalent to estimating absolute values of the diagonals which are polynomials.

[10] We always stick to the convention of including degenerate paths (one-point path).

4.4 Some general notions on graphs

We finish our preparation with some general graph-theoretic notions.

▶ **Definition 4.10** (Vertex-separator). *Given graph H and two vertex-subsets $A, B \subseteq V(H)$, call $S \subseteq V(H)$ an (A, B)-**vertex-separator** if any path*[11] *from A to B in H must pass through S. Let*

$$s_{A,B}(H) := \min\{|S| \mid S \textit{ is an } (A, B)\textit{-vertex-separator}\}.$$

*A vertex-separator achieving this minimum is a **min-separator**. $\mathrm{mSep}_{A,B}(H)$ denotes the set of all min-separators.*

This definition naturally applies to ribbons $\mathcal{R} = (A, B; T)$, by using the graph H as on $V(T) \cup A \cup B$ with edge-set T. In that case we can write the corresponding size and set of the min-separators as

$$s_{A,B}(T), \quad \mathrm{mSep}_{A,B}(T) \textit{ or } \mathrm{mSep}(\mathcal{R}).$$

Menger's theorem. For any finite graph H, $s_{A,B}(H)$ equals to the maximum number of vertex-disjoint paths from A to B in H.

▶ **Definition 4.11.** *For ribbon $\mathcal{R} = (A, B; T)$, let us define its **reduced size** to be*

$$e_{A,B}(T) := |V(T) \cup A \cup B| - s_{A,B}(T). \tag{4.8}$$

The reduced size is double of the exponent in n in the bound of Theorem 4.8, hence is the controlling parameter of the norm of the graphical matrix.

5 Non-exact case PSDness: a refresh

In this section, we review and refresh the proof techniques for the non-exact problem. In Section 5.1 and 5.2, we show Theorem 1.5(2) via the so-called *mod-order analysis*, which gives a conceptually different approach to the techniques. In Section 5.3, we formalize the recursive factorization in a convenient language and extend it properly for later use.

Declaration. Section 5.2 is only for Theorem 1.5(2). The reader can safely skip it if she wants to proceed directly to the proof of Theorem 1.4.

Notation. Thoughout Section 5, M' denotes the $\binom{[n]}{\frac{d}{2}} \times \binom{[n]}{\frac{d}{2}}$-minor[12] of the non-exact moment matrix.

$$M'(I, J) = \sum_{T:|V(T)\cup I\cup J|\le\tau} (\frac{\omega}{n})^{|V(T)\cup I\cup J|}\chi_T \quad \forall I, J \in \binom{[n]}{d/2}. \tag{5.1}$$

Goal of Section 5. Diagonalize M' approximately, such that the difference matrix is negligible (w.h.p. when plugging G).

[11] Same as in the previous footnote. In particular, every vertex-separator contains $A \cap B$.

[12] Strictly speaking, PSDness of this minor is not sufficient as we do not have a homogeneity reduction in non-exact case. Nevertheless, it suffices to demonstrate the idea.

5.1 Step 1: Diagonalization of $\mathbb{E}[M']$

▶ **Proposition 5.1.** *$\mathbb{E}[M'] = CC^\top$, where C is the $\binom{[n]}{d/2} \times \binom{[n]}{\le d/2}$-matrix*

$$C = (\zeta^\top)_{d/2,\le d/2} \cdot \operatorname{diag}\left(\sqrt{t(|A|)}\right)_{A\in\binom{[n]}{\le d/2}} \tag{5.2}$$

and $t(r) = (1 - O(\frac{d\omega}{n})) \cdot (\frac{\omega}{n})^{d-r}$ for all $r = 0, ..., d/2$.

This can be shown by a similar calculation as in [25], as below.

▶ **Definition 5.2** (See e.g. [9]). *Fix parameters n, k. A **Johnson scheme** $\mathfrak{J}$ is an $\binom{[n]}{k} \times \binom{[n]}{k}$-matrix that satisfies $\mathfrak{J}(I,J) = \mathfrak{J}(I',J')$ whenever $|I \cap J| = |I' \cap J'|$.*

It can be checked that (fix n, k) all Johnson schemes are symmetric matrices and form a commutative $\mathbb{R}$-algebra, so they are simultaneously diagonalizable. In below we fix n and $k = d/2$. An obvious $\mathbb{R}$-basis for Johnson schemes is $D_0, ..., D_{d/2}$ where

$$D_r(I,J) = \begin{cases} 1, & \text{if } |I \cap J| = r \\ 0, & \text{o.w.} \end{cases} \qquad \forall I, J \in \binom{S}{d/2}. \tag{5.3}$$

Another basis which we denote by $\mathfrak{J}_0, ..., \mathfrak{J}_{d/2}$ is

$$\mathfrak{J}_r(I,J) = \binom{|I \cap J|}{r}, \quad \forall I, J \in \binom{[n]}{\frac{d}{2}}. \tag{5.4}$$

$\mathfrak{J}_0, ..., \mathfrak{J}_{d/2}$ are PSD matrices since

$$\mathfrak{J}_r = \sum_{A \subseteq [n], |A| = r} u_A u_A^\top \quad \text{where} \quad u_A \in \mathbb{R}^{\binom{[n]}{k}},\ u_A(B) = 1_{A \subseteq B}. \tag{5.5}$$

Clearly $\mathfrak{J}_{d/2} = \mathrm{Id}$. More generally, we have:

▶ **Fact 5.3** (See e.g. (4.29) in [9]). *The Johnson schemes (for $(n, d/2)$) have shared eigenspace-decomposition $\mathbb{R}^{\binom{[n]}{d/2}} = V_0 \oplus ... \oplus V_{d/2}$, and*

$$\mathfrak{J}_r = \bigoplus_{i=0}^{\frac{d}{2}} \lambda_r(i) \cdot \Pi_i \quad \text{for } r = 0, ..., d/2$$

where Π_i is the orthogonal projection to V_i w.r.t. the Euclidean inner product, and the eigenvalues are

$$\lambda_r(i) = \binom{\frac{d}{2} - i}{r - i}\binom{n - \frac{d}{2} - i}{\frac{d}{2} - r}, \quad 0 \le i \le \frac{d}{2}.$$

▶ **Lemma 5.4.** *$\mathbb{E}[M'] = \sum_{r=0}^{d/2} t(r)\mathfrak{J}_r$ where each $t(r) = (1 - O(\frac{d\omega}{n})) \cdot (\frac{\omega}{n})^{d-r}$.*

Proof. By definition, $\mathbb{E}[M'] = \sum_{r=0}^{d/2} (\frac{\omega}{n})^{d-r} D_r$. Note each D_r decomposes as

$$D_r = \sum_{r'=r}^{d/2} (-1)^{r'-r} \binom{r'}{r} \cdot \mathfrak{J}_{r'} \tag{5.6}$$

since $RHS(I,J) = \sum_{r'=r}^{d/2} (-1)^{r'-r}\binom{r'}{r}\binom{|I\cap J|}{r'} = \sum_{r'=r}^{|I\cap J|} (-1)^{r'-r}\binom{|I\cap J|}{r}\binom{|I\cap J|-r}{r'-r} = \binom{|I\cap J|}{r} \cdot 1_{|I\cap J|=r} = 1_{|I\cap J|=r}$. So together,

$$\begin{aligned}
\mathbb{E}[M'] &= \sum_{r=0}^{d/2} (\frac{\omega}{n})^{d-r} \left(\sum_{r'=r}^{d/2} (-1)^{r'-r} \binom{r'}{r} \mathfrak{J}_{r'} \right) \\
&= \sum_{r'=0}^{d/2} \mathfrak{J}_{r'} \cdot \left(\sum_{r=0}^{r'} (\frac{\omega}{n})^{d-r} (-1)^{r'-r} \binom{r'}{r} \right) \qquad (5.7)\\
&= \sum_{r'=0}^{d/2} \mathfrak{J}_{r'} \cdot (\frac{\omega}{n})^{d-r'} (1 - \frac{\omega}{n})^{r'}
\end{aligned}$$

which proves the lemma. ◀

By Lemma 5.4 and (5.5), if let $t(r) = (\frac{\omega}{n})^{d-r'}[1-\frac{\omega}{n}]^{r'}$ then

$$\mathbb{E}[M'] = \sum_{A:|A|\le d/2} t(|A|) u_A u_A^\top = (\zeta^\top)_{d/2,\le d/2} \cdot \operatorname{diag}\Big(t(|A|)\Big) \cdot \zeta_{\le d/2, d/2} = CC^\top,$$

where used that the matrix $(\zeta^\top)_{d/2,\le d/2}$ has columns $\{u_A \mid |A| \le d/2\}$. This proves Proposition 5.1.

5.2 Step 2: Mod-order analysis toward "coarse" diagonalization

Given $\mathbb{E}[M'] = CC^\top$, ideally we hope to continue to solve for

$$M' = NN^\top \qquad (5.8)$$

with $\mathbb{E}[N] = C$, and N extending C by non-trivial Fourier characters. Also, we restrict ourselves to symmetric solutions w.r.t. shapes (Def. 4.6).

Toward this goal, we define and study a relaxed equation first (Definition 5.5). Let us start with its motivation.

(1) Order in $\frac{\omega}{n}$. Entries of M' all have a clear order in $\frac{\omega}{n}$. Like in fixed-parameter problems, we treat $\frac{\omega}{n}$ as a distinguished structural parameter and try to solve the correct power of $\frac{\omega}{n}$ in terms in N.

(2) Norm-match. Let's have a closer look into

$$\mathbb{E}[M'] = CC^\top = \sum_{r=0}^{d/2} (1 - O(\frac{d\omega}{n})) \cdot (\frac{\omega}{n})^{d-r} \mathfrak{J}_r.$$

By fact 5.3, each $\mathfrak{J}_r$ b has norm $\binom{d/2}{r} \cdot n^{d/2-r}$ so

$$\|C_r C_r^\top\| \approx \binom{d/2}{r} \cdot (\frac{\omega}{n})^{d-r} n^{d/2-r}, \quad r = 0, ..., d/2. \qquad (5.9)$$

We expect $N_r(N_r)^\top$ to concentrate around $C_r(C_r)^\top$, so the norm of the "random" part, i.e. matrix of nontrivial Fourier characters in $N_r(N_r)^\top$, is expected to be bounded by (5.9). The tight bound from Theorem 4.8 tells how this may happen, which we review below.

It will be convenient to use a scaling of variables: let

$$L = (L_0, ..., L_{\frac{d}{2}}) = (N_r \cdot (\frac{\omega}{n})^{\frac{-|A|}{2}})_{0 \le r \le \frac{d}{2}},$$

then

$$M' = L \cdot \mathrm{diag}\left((\frac{\omega}{n})^{|A|}\right) \cdot L^\top \quad \text{with} \quad \mathbb{E}[L] = (\ C_r \cdot (\frac{\omega}{n})^{-r/2}\ )_{r=0,1,\ldots,d/2.} \tag{5.10}$$

Now suppose

$$L_r(I,A) = \sum_{\text{small } T} \beta_{I,A}(T)\chi_T, \quad A \in \binom{[n]}{r}$$

where assume as in (1), an order of $\frac{\omega}{n}$ can be separated:

$$\beta_{I,A}(T) = \underbrace{(\frac{\omega}{n})^x}_{\text{main-order term}} \cdot (\text{ factor} \ll \frac{n}{\omega} \text{ and} \gg \frac{\omega}{n}\ ). \tag{5.11}$$

Fix I, A, T, we are looking for the condition on x in order to have the expected norm control on $L_r(\frac{\omega}{n})^r(L_r)^\top$. Ignore for a moment the cross-terms, such a single graphical matrix square in $L_r(\frac{\omega}{n})^r L_r^\top$ is

$$(\frac{\omega}{n})^{2x} R_{(I,A;T)} \cdot (\frac{\omega}{n})^r \cdot R_{(I,A;T)}^\top$$

which has norm[13]

$$\lessapprox (\frac{\omega}{n})^{2x+r} \cdot n^{\mathrm{e}_{I,A}(T)} \cdot 2^{O(|V(T)\cup I \cup A|)} \cdot (\log n)^{>0}$$

by Theorem 4.8. Here recall $e_{I,A}(T) = |V(T) \cup I \cup A| - s_{I,A}(T) (\geq |I| - |A| = \frac{d}{2} - r)$. Compare this with (5.9), we need $(\frac{\omega}{n})^{2x} n^{e_{I,A}(T)} < \binom{d/2}{r}(\frac{\omega}{\sqrt{n}})^{d/2-r}$. If think of 2^d as qualitatively smaller than any positive constant power of ω, n, the natural bound to put is $x \geq e_{I,A}(T)$ which actually is the limit requirement when $\frac{\log \omega}{\log n} \to \frac{1}{2}$. Suggested by this, we will set the restriction $x \geq e_{I,A}(T)$ right from the start in the relaxed equation.

The above motivation leads to the following definition. Take a ring $\mathbb{A}$ by adding fresh variables α and χ_T's to $\mathbb{R}$, where T ranges over subsets of $\binom{[n]}{2}$ and they only satisfy relations $\{\chi_{T'} \cdot \chi_{T''} = \chi_T : T' \oplus T'' = T\}$.

▶ **Definition 5.5.** *The* ***mod-order equation*** *is*

$$L_\alpha \cdot \mathrm{diag}\left(\alpha^{|A|}\right) \cdot (L_\alpha)^\top = M_\alpha \qquad \text{mod } (*) \tag{5.12}$$

on the $\binom{[n]}{d/2} \times \binom{[n]}{\leq d/2}$ *matrix variable* L_α *in ring* $\mathbb{A}$*, where*

$$M_\alpha(I,J) := \sum_{T:|V(T)\cup I \cup J| \leq \tau} \alpha^{|V(T)\cup I\cup J|}\chi_T,$$

and mod $(*)$ *is the* ***modularity****, which means position-wise mod the ideal*

$$\left(\{\alpha^{|V(T)\cup I \cup J|+1}\chi_T\},\ \{\chi_T : |V(T) \cup I \cup J| > \tau\}\right).$$

Moreover, if denote $L_\alpha(I,A) = \sum_T \beta_{I,A}(T)\chi_T$ *where* $\beta_{I,A}(T) \in \mathbb{R}[\alpha]$*, then*[14]

$$\alpha^{e_{I,A}(T)} \mid \beta_{I,A}(T) \quad \forall I, A, T. \tag{5.13}$$

We are interested in solutions that are ***symmetric****, i.e.* $\beta_{I,A}(T') = \beta_{J,B}(T'')$ *whenever* $(I,A;T')$*,* $(J,B;T'')$ *are of the same shape.*

[13] Here the matrix is naturally truncated from $2^{[n]} \times 2^{[n]}$, which doesn't change anything since the original matrix is always 0 elsewhere.

[14] Recall $e_{I,A}(T')$ is the reduced size $|V(T') \cup I \cup A| - s_{I,A}(T')$ (Def. 4.11).

The following is the key observation. Its proof demonstrates how to make deductions from the mod-order equations efficiently, and is presented in Appendix A.1.

▶ **Lemma 5.6** (Order match). *If a product $\alpha^{|A|} \cdot \beta_{I,A}(T') \cdot \beta_{J,A}(T'')$ from the LHS of* (5.12) *is nonzero mod* $(*)$, *then both of the following hold:*

$$A \text{ is a min-separator for both } (I, A; T'),\ (J, A; T''); \tag{5.14}$$

$$(V(T') \cup I \cup A)\ \cap\ (V(T'') \cup J \cup A) = A. \tag{5.15}$$

Moreover, (5.14), (5.15) *imply that*

$$A \text{ is a min-separator of } (I, J; T) \text{ (where } T = T' \oplus T''\text{)}; \tag{5.16}$$

$$|V(T') \cup I \cup A|,\ |V(T'') \cup J \cup A| \le \tau. \tag{5.17}$$

By this lemma, in an imagined solution we can assume $\beta_{I,A}(T') \neq 0$ only when it satisfies its part in conditions (5.14), (5.17).

Using this information, plus a further technique of *polarization*, we can deduce the following Proposition 5.8 which is the main takeaway of the analysis here. A graph-theoretic fact (the "in particular" part below) appears exactly as the solvability condition. For deductions see Appendix A.2.

▶ **Fact 5.7** ([11]). *For any ribbon $(I, J; T)$, the set of all min-separators, $\mathrm{mSep}_{I,J}(T)$, has a natural poset structure: min-separators $A_1 \le A_2$ iff A_1 separates $(I, A_2; T)$, or equivalently as can be checked, iff A_2 separates $(J, A_1; T)$. The set is actually a **lattice** under this partial-ordering: $\forall A_1, A_2 \in \mathrm{mSep}_{I,J}(T)$ their join and meet exist. In particular, there exist unique **minimum** and **maximum**.*

Denote the minimum by $S_l(I, J; T)$ and the maximum by $S_r(I, J; T)$, which is the "leftmost" and "rightmost" min-separator, respectively.

▶ **Proposition 5.8** (Mod-order diagonalization). *Let*

$$L_\alpha(I, A) := \sum_{\substack{T':\ |V(T') \cup I \cup A| \le \tau \\ A = S_l(I, A; T') \\ T' \cap E(A) = \emptyset \\ (I, A; T') \text{ left-generated (Def. 4.4)}}} \alpha^{e_{I,A}(T')} \chi_{T'},$$

$$Q_{0,\alpha}(A, B) := \sum_{\substack{T_m:\ |T \cup A \cup B| \le \tau \\ A, B \in \mathrm{mSep}_{A,B}(T_m)}} \alpha^{e_{A,B}(T_m)} \chi_{T_m}$$

(T_m to indicate "middle"). Then

$$L_\alpha \cdot [\mathrm{diag}\left(\alpha^{\frac{|A|}{2}}\right) \cdot Q_{0,\alpha} \cdot \mathrm{diag}\left(\alpha^{\frac{|A|}{2}}\right)] \cdot L_\alpha^\top = M_\alpha \qquad \text{mod } (*) \tag{5.18}$$

where recall $(*)$ means ideal $(\{\alpha^{|V(T) \cup I \cup J| + 1} \chi_T\},\ \{\chi_T : |V(T) \cup I \cup J| > \tau\})$ position-wise on each (I, J).

Equation (5.18) is slightly weaker than a solution to (5.12) but is sufficient for all use, as we are only concerned with PSDness. In particular, it gives the first-approximate diagonalization of the matrix M', recast as Definition 5.9 below. This shows Theorem 1.5(2).

5.3 Recursive factorization

In this subsection, we give a formalization and extension of the recursive factorization technique, which is used to refine the coarse diagonalization from Step 2 above. We give some new notions that are convenient and extendable to matrix products (Def. 5.13, 5.15), along with some simplification (Lem. 5.25) and refinement (Prop. 5.24) for later use.

First, the coarse diagonalization (5.18) can be recast in $\mathbb{R}[\{\chi_T\}]$-matrices as below.

▶ **Definition 5.9.** *Let L be the $\binom{[n]}{\frac{d}{2}} \times \binom{[n]}{\leq \frac{d}{2}}$-matrix*

$$L(I,A) := \sum_{\substack{T'\colon\ |V(T')\cup I\cup A|\leq\tau\\ A=S_l(I,A;T')\\ T'\cap E(A)=\emptyset\\ (I,A;T')\text{ left-generated}}} (\frac{\omega}{n})^{|V(T')\cup I\cup A|-|A|}\chi_{T'}, \tag{5.19}$$

and Q_0 be the $\binom{[n]}{\leq\frac{d}{2}} \times \binom{[n]}{\leq \frac{d}{2}}$-matrix

$$Q_0(A,B) := \sum_{\substack{T_m:|T_m\cup A\cup B|\leq\tau\\ A,B\in\mathrm{mSep}_{A,B}(T_m)}} (\frac{\omega}{n})^{|V(T_m)\cup A\cup B|}\chi_{T_m}. \tag{5.20}$$

Finally, let

$$D := \operatorname{diag}\left((\frac{\omega}{n})^{\frac{|A|}{2}}\right)_{A\in\binom{[n]}{\leq d/2}}. \tag{5.21}$$

*We call $L(DQ_0)L^\top$ the **first-approximate diagonalization** of M'.*

Despite of its name ("approximate"), the difference

$$M' - L(DQ_0D)L^\top \tag{5.22}$$

is, however, far from negligible. This is where the recursive factorization will be applied, and in the end it will give

$$M' = L\cdot[D\cdot(Q_0 - Q_1 + Q_2 ... \pm Q_{d/2})\cdot D]\cdot L^\top + \mathcal{E} \tag{5.23}$$

for some negligible error-matrix $\mathcal{E}$.

▶ Remark 5.10. Use of D is superficial in (5.22), (5.23); we keep it so that the middle matrices Q_i are better-positioned. The LD here corresponds to the "L" matrix in [5].

Let us start with some necessary notions.

5.3.1 More notion on graphs

▶ **Definition 5.11** ([5] Def. 6.5[15]). *For any ribbon $\mathcal{R} = (I,J;T)$, its **canonical decomposition** is a ribbon-triple*

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) = ((I,A;T_l),(A,B;T_m),(B,J;T_r))$$

[15] Similar notions actually appeared implicitly in the mod-order analysis (cf. condition (5.14), (5.15)), while here they appear in a more "canonical" left-, middle-, right- form.

determined uniquely by the following. $A = S_l(I, J; T)$, $B = S_r(I, J; T)$. $V(\mathcal{R}_l)$ *is* A *unioned with the set of vertices reachable by paths from* I *in* T *without touching* A*, and* $T_l = T|_{V(\mathcal{R}_l)} \backslash E(A)$*. Similarly,* $V(\mathcal{R}_r)$ *is* B *unioned with the set of the vertices reachable from* J *in* T *without touching* B*, and* $T_r = T|_{V(\mathcal{R}_r)} \backslash E(B)$*. Finally,* $T_m = T \backslash (T' \sqcup T'')$*.*

R_l, R_m, R_r *are called the* ***left-, middle-, right- ribbon*** *of* $\mathcal{R}$*, respectively.*

▶ Remark 5.12 (Properties of the canonical decomposition). A few properties follow from the definition of the canonical decomposition of $\mathcal{R} = (I, J; T)$.

$$A = S_l(I, A; T_l),\ \ B = S_r(B, J; T_r)$$

(so they are unique separator of $\mathcal{R}_l, \mathcal{R}_r$, respectively);

$$T_l \cap E(A) = \emptyset = T_r \cap E[A];$$

$$\mathcal{R}_l \text{ is left-generated}, \quad \mathcal{R}_r \text{ is right-generated} \quad \text{(Def. 4.4)};$$

$$A, B \in \text{ mSep}_{A,B}(T_m) \quad (\text{so } |A| = |B|).$$

The above four are about each of $\mathcal{R}_l$, $\mathcal{R}_m$, $\mathcal{R}_m$ (the “inner” conditions). Moreover, there is the intersection property on pairs of them (the “outer” conditions)[16]:

$$V(\mathcal{R}_l) \cap V(\mathcal{R}_m) \subseteq A,\ V(\mathcal{R}_m) \cap V(\mathcal{R}_r) \subseteq B,\ V(\mathcal{R}_l) \cap V(\mathcal{R}_r) \subseteq A \cap B$$

which implies

$$e(\mathcal{R}_l) + |V(\mathcal{R}_m)| + e(\mathcal{R}_r) = |V(\mathcal{R})|. \tag{5.24}$$

The canonical decomposition can be *reversely* described as follows.

▶ **Definition 5.13** (Inner and outer canonicality). *For a triple of ribbons in the form*

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) = \Big((I, A; T_l), (A, B; T_m), (B, J; T_r)\Big)$$

(T_l, T_m, T_r *are arbitrary subsets of an edge-set), their* ***ribbon-sum*** *is ribbon*

$$(I, J; T) \quad \textit{where } T = T_l \oplus T_m \oplus T_r.$$

The triple is called ***inner-canonical****, if they satisfy the “inner” conditions:*

$$\begin{aligned} &A = S_l(I, A; T_l), \quad B = S_r(B, J; T_r),\\ &T_l \cap E(A) = \emptyset = T_r \cap E[A],\\ &\mathcal{R}_l \textit{ left-generated}, \quad \mathcal{R}_r \textit{ right-generated},\\ &A, B \in \text{ mSep}_{A,B}(T_m). \end{aligned} \tag{5.25}$$

The triple is ***outer-canonical*** *if they satisfy the “outer” condition:*

$$V(\mathcal{R}_l) \cap V(\mathcal{R}_m) \subseteq A,\ V(\mathcal{R}_m) \cap V(\mathcal{R}_r) \subseteq B,\ V(\mathcal{R}_l) \cap V(\mathcal{R}_r) \subseteq A \cap B. \tag{5.26}$$

The triple is a ***canonical triple*** *if it is both inner- and outer- canonical.*

[16] cf. conditions (5.14), (5.15)

▶ **Proposition 5.14.** *Canonical triples are 1-1 correspondent to their ribbon-sum, via the canonical decomposition.*

Proof. This follows by an immediate check from the definition. ◀

We further extend the notions to matrix products. Recall $\mathbb{R}[\{\chi_T\}]$ is the ring from adding fresh variables χ_T's into $\mathbb{R}$ for every $T \subseteq \binom{[n]}{2}$ (fixing an n), with relations $\{\chi_{T'} \cdot \chi_{T''} = \chi_T \mid T' \oplus T'' = T\}$.

▶ **Definition 5.15** (Approximate form). *Suppose matrices X, Y have rows and columns indexed by subsets of $[n]$ with entries in $\mathbb{R}[\{\chi_T\}]$; and in every entry, each character regarded as a ribbon on distinguished sets (row, column) has ribbon size $\leq \tau$. Suppose X, Y have dimensions s.t. $XYX^\top$ is defined.*

Every nonzero triple product (without collecting like-terms) in

$$XYX^\top \tag{5.27}$$

thus has form

$$\underbrace{X(I, A; T_l)Y(A, B; T_m)X(J, B; T_r)}_{\text{nonzero in } \mathbb{R}}\chi_{T_l} \cdot \chi_{T_m} \cdot \chi_{T_r}, \tag{5.28}$$

and can be identified with a ribbon-triple in the natural way, with

$$X(I, A; T_l)Y(A, B; T_m)X(J, B; T_r)\chi_{T_l \oplus T_m \oplus T_r} \quad \in \mathbb{R}[\{\chi_T\}]$$

its ***resulting term****. We say* (5.28) *is an* ***outer-canonical product*** *if the ribbon-triple is outer-canonical, and it* ***exceeds degree*** *if $|V(T) \cup I \cup J| > \tau$.*

The ***approximation form*** *of $XYX^\top$ is:*

$$XYX^\top = \left(XYX^\top\right)_{\text{can}} + (XYX^\top)_{\text{non-can}} + \mathcal{E}_{\text{deg}}, \tag{5.29}$$

or equivalently,

$$\left(XYX^\top\right)_{\text{can}} = XYX^\top - (XYX^\top)_{\text{non-can}} - \mathcal{E}_{\text{deg}},$$

where $\left(XYX^\top\right)_{\text{out-can}}$ is the matrix collecting all terms of outer-canonical products that do not exceed degree, $(XYX^\top)_{\text{non-can}}$ collecting all terms of non-outer-canonical products, and $\mathcal{E}_{\text{deg}}$ collecting all rest terms.

▶ Remark 5.16. With this language, Proposition 5.14 gives an *a posteriori* explanation of the coarse diagonalization (Def. 5.9): $M' = [L(DQ_0D)L^\top]_{\text{can}}$.

5.3.2 Recursive factorization: the machinery

We start with the following, which is Definition 5.9 restated in the current language.

▶ **Definition 5.17** (First-approximate factorization of M').

$$M' = L(DQ_0D)L^\top - [L(DQ_0D)L^\top]_{\text{non-can}} - \mathcal{E}_{1;\text{deg}} \tag{5.30}$$

where $\mathcal{E}_{1;\text{deg}}$ is by Def. 5.15, applied to the product $L(DQ_0D)L^\top$, where the index "1" is added for later convenience. $L(DQ_0D)L^\top$ is celled the ***first-approximate factorization*** *of M'.*

The high-degree error $\mathcal{E}_{1;\text{deg}}$ is actually negligible in norm[17] (we will prove the analogous statement in the exact case); the main task is to analyze the "main error", $[L(DQ_0D)L^\top]_{\text{non-can}}$. For this, the key point of the whole technique is

$[L(DQ_0D)L^\top]_{\text{non-can}}$ itself factors through $L, L^\top$ approximately, too.

I.e. $\exists Q_1$ s.t.

$$[L(DQ_0D)L^\top]_{\text{non-can}} = [L(DQ_1D)L^\top]_{\text{can}} + \mathcal{E}'_{1;\text{negl}}. \tag{5.31}$$

for some negligible $\mathcal{E}'_{1;\text{negl}}$. And we can repeat this for $[L(DQ_1D)L^\top]_{\text{non-can}}$ and so on. To describe the factorization (5.31), a generalized notion is useful.

▶ **Definition 5.18** ([5], Def. 6.9[18]). *A **generalized ribbon** is a usual ribbon together with a new set of isolated vertices. In symbol, it is denoted as $\mathcal{R}^* = (A, B; T^*)$ where*

$$T^* = T \sqcup \mathcal{I},$$

*T an edge-set, $\mathcal{I}$ a vertex set disjoint from $V(T) \cup A \cup B$, called the **isolated vertex-set of $\mathcal{R}^*$**, denoted as $\mathcal{I}(\mathcal{R}^*)$. $V(\mathcal{R}^*) = V(T) \cup A \cup B \cup \mathcal{I}$. A usual ribbon is also a generalized ribbon with $\mathcal{I} = \emptyset$. $(A, B; T)$ is called the (unique) largest ribbon in $\mathcal{R}$.*

▶ Remark 5.19. $\mathcal{I}(\mathcal{R}^*)$ could be different from the isolated set of the underlying graph, as it excludes vertices in $A \cup B$.

▶ **Definition 5.20.** *A **side-inner-canonical triple** is*

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) = ((I, A; T_l),\ (A, B; T_m),\ (B, J; T_r))$$

where $\mathcal{R}_l$, $\mathcal{R}_r$ are ribbons satisfying the inner-canonical conditions on their part (the first three of (5.25)), while $\mathcal{R}_m$ is just a ribbon.

The following operation is the technical core of recursive factorizations.

▶ **Definition 5.21** (Separating factorization; Def. 6.10 of [5]). *Given an side-inner-canonical tripe*

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) = ((I, A; T_l),\ (A, B; T_m),\ (B, J; T_r)),$$

*denote $T = T_l \oplus T_m \oplus T_r$, and denote by Z the multi-set of "unexpected intersections" i.e. multi-set of vertices from $(\mathcal{R}_l \cap \mathcal{R}_m) - A$, $(\mathcal{R}_m \cap \mathcal{R}_r) - B$, $(\mathcal{R}_l \cap \mathcal{R}_r) - (A \cap B)$. Call $z(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) = |Z|$ the **intersection size** of the triple. It can be checked that*

$$|V(\mathcal{R}_l) \cup V(\mathcal{R}_m) \cup V(\mathcal{R}_r)| = |V(\mathcal{R}_l)| + |V(\mathcal{R}_m)| + |V(\mathcal{R}_r)| - |A| - |B| - z. \tag{5.32}$$

We further separate this triple into an "outer-canonical" one, as follows.

Define S'_l to be the leftmost min-separator of $(I, A \cup (Z \cap V(\mathcal{R}_l)); T_l)$, and similarly S'_r the right-most min-separator of $(B \cup (Z \cap V(\mathcal{R}_r)), J; T_r)$. Note $S'_l, S'_r \subseteq V(T) \cup I \cup J$ from definition.

[17] Matrices considered all have support on clique-rows and clique-columns, given G.
[18] It was called *improper ribbon*, but we feel the name here is perhaps more proper.

*Define ribbon $\mathcal{R}_l' = (I, S_l'; T_l')$, whose vertex set $V(\mathcal{R}_l')$ is S_l' unioned with the set of vertices in $\mathcal{R}_l$ reachable from I by paths in T_l without touching S_l', and T_l' is $T_l \backslash E(S_l')$ restricted to $V(\mathcal{R}_l')$. Ribbon $\mathcal{R}_r'$ is symmetrically defined. In particular, $T_l' \cap T_r' = \emptyset$. $\mathcal{R}_m^*$ is the **generalized** ribbon $(S_l', S_r'; T_m^*)$ where*

$$T_m^* = T \backslash (T_l' \sqcup T_r') \ \sqcup \ \mathcal{I}(\mathcal{R}_m^*),$$

$\mathcal{I}(\mathcal{R}_m^*)$ collecting all the rest isolated vertices:

$$\mathcal{I}(\mathcal{R}_m^*) = V(\mathcal{R}_l) \cup V(\mathcal{R}_m) \cup V(\mathcal{R}_r) \ - \ V(T) \cup I \cup J. \tag{5.33}$$

*The resulting $(\mathcal{R}_l', \mathcal{R}_m^*, \mathcal{R}_r')$ is called the **separating factorization** of ribbon triple $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$, which we denote as*

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) \to (\mathcal{R}_l', \mathcal{R}_m^*, \mathcal{R}_r'). \tag{5.34}$$

▶ Remark 5.22 (Properties of separating factorization). Some natural properties follow. Let $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) \to (\mathcal{R}_l', \mathcal{R}_m^*, \mathcal{R}_r')$ in the same notation as above.

(1) The resulting triple $(\mathcal{R}_l', R_m^*, \mathcal{R}_r')$ is side-inner-canonical and outer-canonical (i.e. their pair-wise vertex intersections are within the corresponding S_l', S_r' and $S_l' \cap S_r'$). So the corresponding ribbon triple (from replacing R_m^* with its largest ribbon) is canonical and is disjoint from $\mathcal{I}(\mathcal{R}_m^*)$.

(2) $\mathcal{R}_l' \subseteq \mathcal{R}_l$, and S_l' separates $(V(\mathcal{R}_l'),\ V(\mathcal{R}_l) - V(\mathcal{R}_l'))$ in $\mathcal{R}_l$. In particular, we can talk about the part of $\mathcal{R}_l$ to the right of S_l', which is disjoint from R_l' and actually can be easily checked to be in $\mathcal{R}_m^*$. Similar fact holds for $\mathcal{R}_r$.

(3) Since S_l' separates (I, A) in $\mathcal{R}_l$, and A is the unique min-separator of $\mathcal{R}_l$, there are $|A|$ many vertex-disjoint paths from A to S_l' in $\mathcal{R}_l$. Similarly for $\mathcal{R}_r$.

▶ **Lemma 5.23** (Lemma 6.14, 7.14 of [5]). *Suppose $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) \to (\mathcal{R}_l', \mathcal{R}_m^*, \mathcal{R}_r')$. In the same notation as in Definition 5.21,*

(1) *$|S_l'| + |S_r'| \geq |A| + |B| + 1$;*

(2) [19] *If further denote $s = \frac{|A|+|B|}{2}$, p' the maximum number of vertex-disjoint paths from S_l' to S_r' in $\mathcal{R}_m^*$, and p the maximum number of vertex-disjoint paths from A to B in $\mathcal{R}_m$, then*

$$2(s' - s) + (p - p') + |\mathcal{I}(\mathcal{R}_m^*)| \leq z(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r).$$

Proof.

(1) By definition there must be some unexpected pair-wise intersection between $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$. In either of the three cases of breaking (5.26), $\exists v \in Z$ that is in $V(\mathcal{R}_l) - A$ or in $V(\mathcal{R}_r) - B$. WLOG suppose the first happens. Then $S_l' \neq A$ since v can be reached from I without passing A by the left-generated condition on $\mathcal{R}_l$. Similarly, if $|S_l'| = |A|$ then it is A as A is the unique min-separator separating (I, A), so this is impossible. Thus $S_l' > A$.

(2) We refer the reader to its proof in the original paper. ◀

Now we apply the above machinery to the target, $L(DQ_0D)L^\top$.

[19] Recall in our setting $\mathcal{R}_m$ is always a ribbon, without any isolated vertex.

5.3.3 Apply the machinery

Conceptually, the separating factorization tells us how to "cancel" the terms in $[L(DQ_0D)L^\top]_{\text{non-can}}$ using $L, L^\top$. Namely, in $L(DQ_0D)L^\top$, any product from $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$ (Def.5.15) that is non-outer-canonical results in a term in $[L(DQ_0D)L^\top]_{\text{non-can}}$ at (I, J), and we can cancel it by the product from its separating factorization $(\mathcal{R}'_l, \mathcal{R}^*_m, \mathcal{R}'_r)$: take R'_l at position (I, S'_l) in L, R'_r at position (S'_r, J) in $L^\top$, and the largest ribbon of $\mathcal{R}^*_m$ at (S'_l, S'_r) in a new middle matrix DQ_1D. I.e., we cancel it by $-[L(DQ_1D)L^\top]_{\text{can}}$.

Of course, there are other triples whose separating factorization result in the same ($\mathcal{R}'_l$, largest ribbon of $\mathcal{R}^*_m$, $\mathcal{R}'_r$) so we need to collect them all in DQ_1D. More seriously, the (I, S'_l)th entry of L is actually a sum of different R'_ls, so we need to make sure that this cancellation works for them simultaneously in multiplication.

The following is what insures the simultaneous cancellation can work. It is stated in a refined version that is more than needed here (i.e. we further distinguish different (i, j) parameters), but this will be needed in the exact case (Lemma 6.20).

▶ **Proposition 5.24** (Solvability condition, cf. Claim 6.12 in [5]). *Fix (I, J, S'_l, S'_r), and a generalized ribbon $\mathcal{R}^*_m$ on (S'_l, S'_r). Let $(\mathcal{R}'_l, \mathcal{R}'_r)$ be inner-canonical left and right ribbons with distinguished sets $(I, S'_l), (S'_r, J)$ respectively, as in Definition 5.13. Let $(\mathcal{R}''_l, \mathcal{R}''_r)$ be another such ribbon pair, with the same reduced size*

$$e(\mathcal{R}'_l) = e(\mathcal{R}''_l),\ \ e(\mathcal{R}'_r) = e(\mathcal{R}''_r).$$

(Or the same size, equivalently.) Then for every fixed tuple (i, j, z) the following holds: there is an 1-1 matching between ribbon-triples

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)\ \text{s.t.}\ \begin{cases}(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) \to (\mathcal{R}'_l, \mathcal{R}^*_m, \mathcal{R}'_r),\\ (e(\mathcal{R}_l),\ e(\mathcal{R}_r),\ z(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)) = (i, j, z).\end{cases} \tag{5.35}$$

and

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)\ \text{s.t.}\ \begin{cases}(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) \to (\mathcal{R}''_l, \mathcal{R}^*_m, \mathcal{R}''_r),\\ (e(\mathcal{R}_l),\ e(\mathcal{R}_r),\ z(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)) = (i, j, z).\end{cases} \tag{5.36}$$

Moreover, this matching fixes every middle $\mathcal{R}_m$.

Proof. We give a reversible map from the set of (5.35) onto the set of (5.36). Take a $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$ from (5.35). By Remark 5.22 (2), the part of $\mathcal{R}_l$ to the right of S'_l is in $\mathcal{R}^*_m$ hence is disjoint from both R'_l and R''_l. Similarly for $\mathcal{R}'_r$, $\mathcal{R}_r$. Now take the map

$$(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r) \mapsto (\phi(\mathcal{R}_l), \mathcal{R}_m, \phi(\mathcal{R}_r))$$

where $\phi(\mathcal{R}_l)$ replace $\mathcal{R}'_l$ to $\mathcal{R}''_l$ within $\mathcal{R}_l$, and $\phi(\mathcal{R}_r)$ replaces $\mathcal{R}'_r$ to $\mathcal{R}''_r$ within $\mathcal{R}_r$. Clearly $\mathcal{R}^*_m$, thus $\mathcal{R}_m$, is unchanged. Also, as $\mathcal{R}'_l$, $\mathcal{R}''_l$ have the same size by assumption, by the disjointness above this replacement operation keeps the size of $\mathcal{R}_l$. Moreover, $\mathcal{R}_l$, $\phi(\mathcal{R}_l)$ have the same right distinguished set which is the unique min-separator of both, so $e(\mathcal{R}_l) = e(\phi(R_l))$. Similarly for $\mathcal{R}_r, \phi(\mathcal{R}_r)$, so the parameter (i, j) is unchanged by ϕ. The intersection parameter z is unchanged too, since the changed part is disjoint from $Z(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$. Finally, the inverse map is given the same way by changing the role of $(\mathcal{R}'_l, \mathcal{R}'_r)$ and $(\mathcal{R}''_l, \mathcal{R}''_r)$. ◀

The following lemma will be repeatedly used.

▶ **Lemma 5.25** (One round of factorization). *Let L be as (5.19), and Q be any $\binom{[n]}{\le d/2}\times\binom{[n]}{\le d/2}$-matrix with entries*

$$Q(A,B)=\sum_{T_m:\ |V(T_m)\cup A\cup B|\le\tau}(\frac{\omega}{n})^{|V(\mathcal{R}_m)|}q(\mathcal{R}_m)\cdot\chi_{T_m}\tag{5.37}$$

where $\mathcal{R}_m$ denotes $(A,B;T_m)$, and $q(\cdot)$ is a function symmetric w.r.t. shapes.

Define matrix $Q',\mathcal{E}'_{\text{negl}}$ as follows so that

$$(LQL^\top)_{\text{non-can}}=(LQ'L^\top)_{\text{can}}+\mathcal{E}'_{\text{negl}}\tag{5.38}$$

holds. First, let

$$Q'(A,B)=\sum_{T_m:\ |V(T_m)\cup A\cup B|\le\tau}(\frac{\omega}{n})^{|V(\mathcal{R}_m)|}q'(\mathcal{R}_m)\cdot\chi_{T_m}\tag{5.39}$$

where $q'(\mathcal{R}_m)$ is as follows. Fix any $\mathcal{R}_m=(A,B;T_m)$ and let $t=|V(\mathcal{R}_m)|\le\tau$, $s=\frac{|A|+|B|}{2}$. For every generalized ribbon $\mathcal{R}_m^*$ that contains $\mathcal{R}_m$ as its largest ribbon and $|V(\mathcal{R}_m^*)|\le\tau$, ***fix*** *a ribbon pair $(\mathcal{R}'_l,\mathcal{R}'_r)$ s.t. $(\mathcal{R}'_l,\mathcal{R}_m^*,\mathcal{R}'_r)$ is the separating factorization for some ribbon triple with $|V(\mathcal{R}'_l)|,|V(\mathcal{R}'_r)|\le\tau$ (if there is none, exclude this $\mathcal{R}_m^*$ in the summation below). Then let*

$$\begin{aligned}q'(\mathcal{R}_m)&=\sum_{\substack{\mathcal{R}_m^*:\ \text{gen. ribbon on }(A,B)\\ |V(\mathcal{R}_m^*)|\le\tau\\ \text{largest ribbon is }\mathcal{R}_m}}(\frac{\omega}{n})^{|\mathcal{I}(\mathcal{R}_m^*)|}\cdot q''(\mathcal{R}_m^*),\quad \textit{where}\\ q''(\mathcal{R}_m^*)&=\sum_{1\le z\le d/2}\ \sum_{\substack{\mathcal{P}=(\mathcal{R}_l,\mathcal{R},\mathcal{R}_r):\ \text{side-inn. can.}\\ \mathcal{P}\to(\mathcal{R}'_l,\mathcal{R}_m^*,\mathcal{R}'_r)\text{ for the fixed }\mathcal{R}'_l,\mathcal{R}'_r\\ z(\mathcal{P})=z}}(\frac{\omega}{n})^z\cdot q(\mathcal{R}).\end{aligned}\tag{5.40}$$

Note $q'(\mathcal{R}_m)$ doesn't depend on the choice $(\mathcal{R}'_l,\mathcal{R}'_r)$ by Proposition 5.24, and $q'(\cdot)$ is also symmetric w.r.t. shapes. Now define $\mathcal{E}'_{\text{negl}}$ s.t. (5.38) holds.

Then the conclusions are:

(1) *W.p. $>1-n^{-9\log n}$ over G, $\left\|\mathcal{E}'_{\text{negl}}\right\|\le\max\{q(A,B;T)\}\cdot n^{-\epsilon\tau}$;*

(2) *If there is a number C for which*

$$\forall\mathcal{R}_m\quad |q(\mathcal{R}_m)|\le C\cdot(\frac{\omega}{n^{1-\epsilon}})^{s-p}\tag{5.41}$$

where p denotes the maximum number of vertex-disjoint paths between A,B in $\mathcal{R}_m$.[20] *Then*

$$\forall\mathcal{R}_m\quad |q'(\mathcal{R}_m)|\le C\cdot(\frac{\omega}{n^{1-\epsilon}})^{s-p+1/3}.$$

Proof. We compare $[LQ'L^\top]_{\text{can}}$ with $[LQL^\top]_{\text{non-can}}$ as step (0), then prove (1), (2).

(0). For any fixed (I,J), recall $[LQL^\top]_{\text{non-can}}(I,J)$ is

$$\sum_{\substack{(\mathcal{R}_l,\mathcal{R}_m,\mathcal{R}_r):\ \text{side. inn. can.}\\ \text{non-outer-can.}\\ \text{all three have size }\le\tau}}(\frac{\omega}{n})^{|V(\mathcal{R}_l)|+|V(\mathcal{R}_m)|+|V(\mathcal{R}_r)|-|A|-|B|}q(\mathcal{R}_m)\chi_{T_l\oplus T_m\oplus T_r}\tag{5.42}$$

where we denoted the distinguished sets of $\mathcal{R}_m$ by (A,B) when $\mathcal{R}_m$ is given. For each $(\mathcal{R}_l,\mathcal{R}_m,\mathcal{R}_r)$ in it, there is a unique $(\mathcal{R}'_l,\mathcal{R}_m^*,\mathcal{R}'_r)$ that is its separating factorization: $(\mathcal{R}_l,\mathcal{R}_m,\mathcal{R}_r)\to(\mathcal{R}'_l,\mathcal{R}_m^*,\mathcal{R}'_r)$. There are two cases.

[20] This is also $s_{A,B}(T_m)$ by Menger's theorem; we use p here for appliance with applying Lemma 5.23(2).

First case: $|V(\mathcal{R}_m^*)| \le \tau$. In this case, there is the corresponding term

$$(\frac{\omega}{n})^{|V(\mathcal{R}_l')|+|V(\mathcal{R}_m')|+V(\mathcal{R}_r')|-|S_l'|-|S_r'|} \cdot (\frac{\omega}{n})^{z+|\mathcal{I}(\mathcal{R}_m^*)|} \cdot q(\mathcal{R}_m')\chi_{T_l' \oplus T_m^* \oplus T_r'} \tag{5.43}$$

in $(LQ'L^\top)_{\text{can}}(I,J)$, where $\mathcal{R}_m'$ denotes the largest ribbon of $\mathcal{R}_m^*$ and $\chi_{T_m^*}$ means the character from $\mathcal{R}_m'$, and $z \ge 1$ is the intersection size of $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$. Recall for the separating factorization, $T_l' \oplus T_m^* \oplus T_r' = T_l \oplus T_m \oplus T_r$ and

$$\begin{aligned} |V(\mathcal{R}_l) \cup V(\mathcal{R}_m) \cup V(\mathcal{R}_r)| &= |V(\mathcal{R}_l')| + |V(\mathcal{R}_m^*)| + |V(\mathcal{R}_r')| - |S_l'| - |S_r'| \\ &= |V(\mathcal{R}_l)| + |V(\mathcal{R}_m)| + |V(\mathcal{R}_r)| - |A| - |B| - z \end{aligned}$$

Also, $|V(R_m^*)| = |V(\mathcal{R}_m')| + |\mathcal{I}(\mathcal{R}_m^*)|$. Together we have that the coefficient in (5.43) equals the one in (5.42) from $(\mathcal{R}_l', \mathcal{R}_m^*, \mathcal{R}_r')$.

Conversely, by definition of Q' and (5.40) and Prop. 5.24 every outer-canonical product in $LQ'L^\top$ corresponds uniquely to a side inner-canonical triple $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$ in the above case. Therefore, $\mathcal{E}_{\text{negl}}'$ by definition collects all terms in the next case.

Second case: $|V(\mathcal{R}_m^*)| > \tau$. By the above explanation, $\mathcal{E}_{\text{negl}}'(I,J) =$

$$\sum_{\substack{(\mathcal{R}_l,\mathcal{R}_m,\mathcal{R}_r)\text{: side. inn. can.} \\ \text{non-outer-can.} \\ \text{all three has size } \le \tau \\ \text{resulting } |V(\mathcal{R}_m^*)| > \tau}} (\frac{\omega}{n})^{|V(\mathcal{R}_l)|+|V(\mathcal{R}_m)|+|V(\mathcal{R}_r)|-|A|-|B|} q(\mathcal{R}_m)\chi_{T_l \oplus T_m \oplus T_r}. \tag{5.44}$$

where we omit writing the obvious condition that $\mathcal{R}_l$ ($\mathcal{R}_r$) has its left (right) vertex set as I (J).

(1) Take a triple $(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)$ in (5.44). Recall

$$\begin{aligned} |V(\mathcal{R}_l)| + |V(\mathcal{R}_m)| + |V(\mathcal{R}_r)| - |A| - |B| &= |V(\mathcal{R}_l) \cup V(\mathcal{R}_m) \cup V(\mathcal{R}_r)| + z \\ &= |V(T) \cup I \cup J| + |\mathcal{I}(\mathcal{R}_m^*)| + z. \end{aligned}$$

Also $|\mathcal{I}(\mathcal{R}_m^*)| \le z + d/2$ as a quick corollary of Lemma 5.23[21]. Fix an $T = T_l \oplus T_m \oplus T_r$ and $a > \tau - |V(T) \cup I \cup J|$, we upper bound the number of triples in (5.44) resulting in $(\frac{\omega}{n})^{|V(T)\cup I \cup J|+a} \cdot \chi_T$ (ignoring $q(\mathcal{R}_m)$ for the moment): to create such a triple, we need to choose a set as $\mathcal{I}(\mathcal{R}_m^*)$ of size $\le a/2 + d/4$ since a is intended to be $|\mathcal{I}(\mathcal{R}_m^*)| + z$ so $a \ge 2\mathcal{I}(\mathcal{R}^*) - d/2$; then to decide the triple over the fixed vertex set there are $< 3^{3\tau} \cdot 2^{3\binom{\tau}{2}}$ many ways. Together, the coefficient of χ_T in (5.44) has absolute value smaller than the following: let $B_0 = \max\{q(\cdot)\}$,

$$\begin{aligned} & B_0 \cdot (\frac{\omega}{n})^{|V(T)\cup I \cup J|+a} \cdot n^{(a+d)/2} 2^{2\tau^2} \\ = {} & B_0 (\frac{\omega}{n^{1-2\epsilon}})^{|V(T)\cup I\cup J|} \left(n^{-2\epsilon}\right)^{|V(T)\cup I \cup J|} \cdot (\frac{\omega}{\sqrt{n}})^a \cdot n^{d/2} 2^{2\tau^2} \\ \le {} & B_0 (n^{-1/2})^{|V(T)\cup I \cup J|} \cdot n^{-2\epsilon(|V(T)\cup I\cup J|+a)} n^{d/2} 2^{2\tau^2} \qquad (\omega \le n^{1/2-4\epsilon}) \\ \le {} & B_0 (n^{-1/2})^{|V(T)\cup I\cup J|} \cdot n^{-1.5\epsilon\tau} \end{aligned}$$

the last step by $|V(T) \cup I \cup J| + a > \tau$ by the case condition and that $d < \epsilon\tau/10$, $2^{2\tau} < n^{\epsilon/10}$. Also, all χ_T appearing in (5.44) has $|V(T)| \le 3\tau$. So by Lemma 4.2, for fixed (I,J), w.p. $> 1 - n^{-10\log n}$

$$|\mathcal{E}_{\text{negl}}'(I,J)| < \sum_{a=0}^{3\tau} B_0 n^{-a/2} n^{-1.5\epsilon\tau} \cdot n^{a/2} n^{4 \log\log n} 2^{a^2} < n^{-1.4\epsilon\tau}.$$

By union bound over $|\{(I,J)\}| < n^d$, w.p. $> 1 - n^{-9\log n}$ $\left\|\mathcal{E}_{\text{negl}}'\right\| < n^d \cdot n^{-1.4\epsilon\tau} < n^{-\epsilon\tau}$.

[21] Actually it can be shown that $|\mathcal{I}(\mathcal{R}_m^*)| \le z$ but we don't need this.

(2) Fix an $\mathcal{R}_m$. By (5.40),

$$q'(\mathcal{R}_m) = \sum_{\substack{z,\mathcal{R}_m^*: \\ \text{largest ribbon } =\mathcal{R}_m}} (\frac{\omega}{n})^{|\mathcal{I}(\mathcal{R}_m^*)|+z} \sum_{\substack{\mathcal{P}=(\mathcal{R}_l,\mathcal{R},\mathcal{R}_r)\text{: side-inn. can.} \\ \mathcal{P}\to(\mathcal{R}_l',\mathcal{R}_m^*,\mathcal{R}_r') \text{ for the fixed } \mathcal{R}_l',\mathcal{R}_r' \\ z(\mathcal{P})=z}} q(\mathcal{R}).$$

For a fixed $\mathcal{R}_m^*$, there are no more than $8^{z\tau} < n^{\epsilon z}$ many triples in the second summation (recall $\mathcal{R}_l', \mathcal{R}_r'$ is fixed), as after fixing whether each vertex appears in each of the three ribbons and fixing $A, B \subseteq \mathcal{R}_m^*$ as distinguished sets of $\mathcal{R}$, we only need to assign possible edges that appear in more than once in the original triple, and it can be checked that such an edge must has at least one end in the already fixed (multi-set) Z of size $\leq z$. Further, by Lemma 5.23(2) and condition (5.41), the second summation in above in absolute value is

$$\begin{aligned} \leq n^{\epsilon z}(\frac{\omega}{n})^{z+|\mathcal{I}(\mathcal{R}_m^*)|}|q(\mathcal{R})| &\leq (\frac{\omega}{n^{1-\epsilon}})^{2(s'-s)+(p-p')+2|\mathcal{I}(\mathcal{R}_m^*)|} \cdot C(\frac{\omega}{n^{1-\epsilon}})^{s-p} \\ &\leq C \cdot (\frac{\omega}{n})^{2|\mathcal{I}(\mathcal{R}_m^*)|} \cdot (\frac{\omega}{n^{1-\epsilon}})^{s'-p'+1/2} \end{aligned}$$

where (s, p) denotes the corresponding parameter for each $\mathcal{R}$ and (s', p') for $\mathcal{R}_m$, and the last step uses $s' - s \geq 1/2$ from Lemma 5.23(1). Finally, in the outer sum, for fixed i_0 there are $< n^{i_0}$ many ways to choose $\mathcal{R}_m^*$ s.t. $|\mathcal{I}(\mathcal{R}_m^*)| = i_0$, and $1 \leq z \leq 3\tau$. So together,

$$|q'(\mathcal{R}_m)| \leq 3\tau \sum_{i_0=0}^{d/2} C \cdot n^{i_0}(\frac{\omega}{n})^{2i_0} \cdot (\frac{\omega}{n^{1-\epsilon}})^{s'-p'+1/2} \leq C \cdot (\frac{\omega}{n^{1-\epsilon}})^{s'-p'+1/3}. \qquad \blacktriangleleft$$

Now we can apply Lemma 5.25 to $[L(DQ_0D)L^\top]_{\text{non-can}}$: in (5.30) let $Q \leftarrow (DQ_0D)$, we get

$$[L(DQ_0D)L^\top]_{\text{non-can}} = [L(DQ_1D)L^\top]_{\text{can}} + \mathcal{E}'_{1;\text{negl}}$$

for some Q_1 and $\mathcal{E}'_{1;\text{negl}}$. Then we can repeat this on $[L(DQ_1Q)L^\top]_{\text{non-can}}$ and so on, to get a final **recursive approximate factorization** of M:

$$\begin{aligned} M' = L\Big(D(Q_0 - Q_1 + Q_2 - ... \pm Q_d)D\Big)L^\top - \Big(&\mathcal{E}_{1;\text{deg}} - ... \pm \mathcal{E}_{1+d;\text{deg}}\Big) \\ + \Big(&\mathcal{E}'_{1;\text{negl}} + ... + \mathcal{E}'_{d;\text{negl}}\Big). \end{aligned} \tag{5.45}$$

Here it implicitly used the following.

▶ **Proposition 5.26** ([5] Claim 6.15). $Q_{d+1} = 0$.

Proof. First we show by induction: $\forall k$, in Q_k every appearing ribbon $R_m = (A, B; T_m)$ has $|A| + |B| \geq k$. Case $k = 0$ is trivial. From k to $k + 1$, by Lemma 5.25 every $\mathcal{R}_m' = (A', B'; T_m')$ in Q_{k+1} is the largest ribbon of some $\mathcal{R}_m^*$ in the separating factorization of some non-outer-canonical triple in $L(DQ_kD)L^\top$. Suppose that triple has the middle part $\mathcal{R}_m = (A, B; T_m)$. Then by the inductive hypothesis $|A| + |B| \geq k$, and by Lemma 5.23(1) $|A'| + |B'| \geq |A| + |B| + 1 \geq k + 1$, and the induction is completed. For $k = 1 + d$, no ribbon can satisfy this while having both distinguished sets in $\binom{[n]}{d/2}$. ◀

We have completed the recursive factorization technique for later use.

▶ Remark 5.27. PSDness of M' would follow from (5.45) by a few last steps[22]. This part is standard, and similar arguments will be given for the exact case (Section 6) so we omit it here.

6 PSDness of the exact pseudo-expectation

Notation. Henceforth M **exclusively** refers to the $d/2$-homogeneous minor of the moment matrix $\widetilde{M}$ in Definition 3.13.

The main theorem of this section is the following.

▶ **Theorem 6.1.** *W.p.* $> 1 - n^{-5\log n}$, $M(G) \succeq n^{-d-1}\mathrm{diag}\left(\widetilde{\mathrm{Cl}}(G)\right)_{\binom{[n]}{d/2}\times\binom{[n]}{d/2}}$.

▶ **Corollary 6.2.** *W.p.* $> 1 - n^{-5\log n}$, $\widetilde{E}x_\emptyset > 0$.

Proof. By construction (3.13), $\widetilde{E}x_\emptyset = \frac{\binom{\omega-d/2}{d-d/2}}{\binom{\omega}{d}\binom{d}{d/2}} \sum_{S:|S|=d/2} \widetilde{E}x_S = \frac{\binom{\omega-d/2}{d-d/2}}{\binom{\omega}{d}\binom{d}{d/2}}\mathrm{Tr}(M)$, and by Theorem 6.1 this is positive with high probability. ◀

Theorem 1.4 is a quick corollary of Theorem 6.1: for our pseudo-expectation from Definition 3.13, its moment matrix is PSD by Theorem 6.1 and Lemma 4.1; it satisfies the Default Constraint by Corollary 6.2 and the discussion above Remark 3.9; and it satisfies the Clique and Size Constraints by Lemma 3.8. The degree-d lower bound follows.

The rest of Section 6 is for proving Theorem 6.1. We first reduce it to the main lemma (Lemma 6.9) in the next subsection, then prove that lemma.

6.1 An Hadamard product and Euler transform

For proving Theorem 6.1, we want to factor the matrix M into an $XYX^\top$ form as in the non-exact case. The first problem is that, unlike in the non-exact situation, here in the expression of $M(I, J)$ (Def. 3.13), the appearance of the parameter

$$u = |I \cap J|$$

makes a similar factorization of terms unlikely[23]. As a first step towards resolving this issue, in this subsection, we express M in a $\Sigma\Pi$-form (6.15) where in each leaf matrix, the dependence on u is removed. In later subsections, we will factor each such leaf matrix.

6.1.1 Hadamard product

By definition (3.17), in $M(I, J)$ the coefficient before χ_T can be re-written as

$$M(I,J;T) = \sum_{c=0}^{u}\Bigg[\frac{1}{\binom{\omega-d+u}{u}}\omega^{u-c}\cdot \underbrace{\left(\binom{a-(d-u)}{c}\binom{n-a}{u-c}n^{-(u-c)}\frac{(a+u-c+8\tau^2)!}{(8\tau^2)!}\left(\frac{\omega}{n}\right)^a\right)}_{:=M_c(u,a)}\Bigg] \tag{6.1}$$

[22] As noted previously, this is not yet the PSDness of the moment matrix as we do not have the homogeneous reduction in non-exact case. A full proof is just similar, though.

[23] It doesn't appear in the non-exact case (5.1) at all.

where again $u = |I \cap J|, a = |V(T) \cup I \cup J|$. This means M is a sum of Hadamard products

$$M = \sum_{c=0}^{\frac{d}{2}} m_c \circ M_c \tag{6.2}$$

where m_c, M_c are matrices: for all $|I|, |J| = d/2$,

$$m_c(I,J) = \frac{1}{\binom{\omega-d+u}{u}}\omega^{u-c} \quad u = |I \cap J| \tag{6.3}$$

$$M_c(I,J) = \begin{cases} \sum\limits_{T:|V(T)\cup I\cup J|\le\tau} \chi_T \cdot M_c(|I\cap J|, |V(T)\cup I\cup J|) & \text{, if } |I\cap J| \ge c; \\ 0 & \text{, o.w.} \end{cases} \tag{6.4}$$

▶ Remark 6.3. It is important to note that we defined m_c to be supported on all (I,J), while let $M_c(I,J) = 0$ if $|I \cap J| < c$, so (6.2) still holds. The use of this is in Lemma 6.4 below.

The intuition behind decomposition (6.2) is that the second factor M_c is "close" to each other for varying c, while the first factor m_c is qualitatively decreasing in c. This, if true, would make it possible for us to concentrate on showing the PSDness in the main case $c = 0$.

The next lemma proves the second half of the above intuition. The other half will be stated more precisely as the Main Lemma 6.9.

▶ **Lemma 6.4.** *For each* $c = 0, ..., d/2$,

$$m_c = \omega^{-c}\sum_{k=0}^{d/2} b_k \cdot \mathfrak{J}_k$$

where $\mathfrak{J}_k$*'s are the Johnson basis* (5.4), $b_k/k! \in [\frac{d}{2\omega}, 1 + \frac{2dk}{\omega}]$. *In particular,*

$$m_0 = \omega m_1 = ... = \omega^{\frac{d}{2}} m_{\frac{d}{2}} \succ \frac{1}{\omega}\mathrm{Id}. \tag{6.5}$$

Proof. By definition, $m_c = \omega^{-c}\sum\limits_{l=0}^{d/2}\frac{\omega^l}{\binom{\omega-d+l}{l}}D_l$, where matrices D_l $(l = 0, ..., d/2)$ are the simple basis of Johnson schemes (5.3). By basis-change (5.6), $m_c = \omega^{-c}\sum\limits_{k=0}^{d/2}\mathfrak{J}_k \cdot k!\left(\sum\limits_{l=0}^{k}(-1)^{k-l}\cdot\underbrace{\left[\frac{\omega}{\omega-(d-l)}\cdot\ldots\cdot\frac{\omega}{\omega-(d-1)}\cdot\frac{1}{(k-l)!}\right]}_{:=f_k(l),\text{ which is }1/k!\text{ if }l=0}\right)$. For fixed k, $f_k(l)$ is increasing in l so $\sum\limits_{l=0}^{k}(-1)^{k-l}f_k(l) \ge f_k(k) - f_k(k-1) > \frac{d/2}{\omega}\cdot(1+\frac{d/2}{\omega})^{k-1} \ge \frac{d}{2\omega}$. Note for $k = d/2$, $\mathfrak{J}_{d/2} = \mathrm{Id}$ so we get (6.5). ◀

6.1.2 Euler transform

Fixing c, now we look into the second factor M_c in (6.2). For fixed $(I, J; T)$, again denote $u = |I \cap J|$, $a = |V(T) \cup I \cup J|$. By (6.1)

$$M_c(u,a) = \binom{a-(d-u)}{c}\binom{n-a}{u-c}n^{-(u-c)}\frac{(a+u-c+8\tau^2)!}{(8\tau^2)!}(\frac{\omega}{n})^a \tag{6.6}$$

is the coefficient of χ_T in $M_c(I,J)$ for $c \le u$, which is a partial function.

▶ **Definition 6.5** (Extended $M_c(u,a)$). *For fixed $c \geq 0$, the function $M_c(u,a)$ in (6.6) is partial, defined for $(u,a) \in \mathbb{N}^2$ s.t.*

$$u \geq c,\ u + a \geq d + c.$$

It can be naturally ***extended to*** $\mathbb{N}^2$ *by letting*

$$\binom{n-a}{u-c} = 0 \quad \text{if } u < c, \tag{6.7}$$

and using the usual convention on binomial coefficients

$$\binom{-m}{k} = (-1)^k \cdot \binom{m+k-1}{k} \quad \forall 0 < m, 0 \leq k; \tag{6.8}$$

$$\binom{m}{k} = 0 \quad \forall 0 \leq m < k \tag{6.9}$$

on the expression $M_c(u,a)$ (6.6). We will still use $M_c(u,a)$ to mean this extended function.

In particular, $\binom{m}{0} = 1$ for all $m \in \mathbb{Z}$; if $0 \leq a - (d-u) < c$ then $M_c(u,a) = 0$ since $\binom{a-(d-u)}{c} = 0$.

To further remove the dependence on $u = |I \cap J|$, consider a decomposition

$$M_c = \sum_{R \in \binom{[n]}{\leq \frac{d}{2}}} M_c^R \tag{6.10}$$

where for each $R \in \binom{[n]}{\leq \frac{d}{2}}$ the matrix M_c^R is supported on rows and columns whose index contains R. More explicitly, for any $(I,J;T)$ let $a = |V(T) \cup I \cup J|$, suppose

$$M_c^R(I,J) := \begin{cases} (\frac{\omega}{n})^a \sum\limits_{T:|V(T)\cup I \cup J| \leq \tau} Y_c(|R|,a) \cdot \chi_T & \text{, if } R \subseteq I, J; \\ 0 & \text{, o.w.} \end{cases} \tag{6.11}$$

for some function $Y_c(u,a)$ to be chosen, then comparing for every tuple $(I,J;T)$ we see that equation (6.10) is equivalent to that for any fixed c, a:

$$\sum_{r=0}^{u} \binom{u}{r} Y_c(r,a) (\frac{\omega}{n})^a = M_c(u,a). \tag{6.12}$$

This suggests to take $Y_c(u,a) \cdot (\frac{\omega}{n})^a$ to be the *inverse Euler transform* (w.r.t. variable u) of the **extended** function $M_c(u,a)$.

▶ **Fact 6.6.**[24] *If $x(m), y(m)$ are two sequences defined on $\mathbb{N}$ s.t.*

$$\forall m \quad x(m) = \sum_{l=0}^{m} \binom{m}{l} y(l),$$

then $x(m)$ is called the ***Euler transform*** *of $y(m)$, whose inverse transform is*

$$\forall m \quad y(m) = \sum_{l=0}^{m} (-1)^{m-l} \binom{m}{l} x(l).$$

[24] The fact itself can be seen as an application of ζ-matrix and its inverse.

▶ **Definition 6.7** (Coefficients in M_c^R). *For every fixed c, define*

$$Y_c(r,a) = \begin{cases} \sum\limits_{l=c}^{r} (-1)^{r-l} \binom{r}{l} \binom{a+l-d}{c} \binom{n-a}{l-c} n^{-(l-c)} \frac{(a+l-c+8\tau^2)!}{(8\tau^2)!} & \text{, if } r \geq c; \\ 0 & \text{, o.w.} \end{cases} \tag{6.13}$$

Then as a clear-up summary, we get:

▶ **Lemma 6.8** (The Hadamard-product decomposition of M).

$$M = \sum_{c=0}^{\frac{d}{2}} m_c \circ \left(\sum_{R: R \in \binom{[n]}{\leq d/2}} M_c^R \right) \tag{6.14}$$

$$= \sum_{R \in \binom{[n]}{\leq d/2}} \underbrace{\left(\sum_{c=0}^{|R|} m_c \circ M_c^R \right)}_{:= M^R} \tag{6.15}$$

where each m_c is as in Lemma 6.4 and each M_c^R has the following expression.

1. $M_c^R = 0$ *if* $|R| < c$;
2. *If* $R \not\subseteq I \cap J$, $M_c^R(I,J) = 0$;
3. *If* $|R| \geq c$ *and* $R \subseteq I \cap J$,

   $$M_c^R(I,J) = \sum_{T: |V(T) \cup I \cup J| \leq \tau} M_c^R(I,J;T) \chi_T$$

   where, if denote $a = |V(T) \cup I \cup J|$,

   $$M_c^R(I,J;T) =$$
   $$(\frac{\omega}{n})^a \underbrace{\sum_{l=c}^{|R|} (-1)^{|R|-l} \binom{|R|}{l} \binom{a+l-d}{c} \binom{n-a}{l-c} n^{-(l-c)} \frac{(a+l-c+8\tau^2)!}{(8\tau^2)!}}_{Y_c(|R|,a),\ (6.13)}. \tag{6.16}$$

4. *For all* $0 \leq c \leq r \leq d/2$ *and* $0 \leq a \leq \tau$,

   $$|Y_c(r,a)| < \tau^{5\tau}.$$

Proof. (1), (2), (3) is definition. To check (6.14) i.e. $M_c = \sum_R M_c^R$, we check for every $(I,J;T)$ where $|I| = |J| = d/2$, $|V(T) \cup I \cup J| \leq \tau$. Let $u = |I \cap J|$, $a = |V(T) \cup I \cup J|$, then note $a - (d-u) \geq 0$, and

$$\sum_{R:} M_c^R(I,J;T) = \sum_{R: R \subseteq I \cap J} M_c^R(I,J;T) = (\frac{\omega}{n})^a \sum_{r=0}^{|I \cap J|} \binom{|I \cap J|}{r} Y_c(r,a).$$

By the Euler transform and (6.12), the RHS equals the extended $M_c(u,a)$. Thus, we only need to see $M_c(u,a) = 0$ if further $u < c$ or $a - (d-u) < c$ (in particular, in such cases $c > 0$), and this is by (6.7), (6.9).

For (4),

$$|Y_c(u,a)| = \left| \sum_{l=c}^{r} (-1)^{r-l} \binom{r}{l} \binom{a+l-d}{c} [\binom{n-a}{l-c} n^{-(l-c)}] \frac{(a+l-c+8\tau^2)!}{(8\tau^2)!} \right|$$
$$< r \cdot 2^r \cdot (2\tau)^r \cdot 1 \cdot (9\tau^2)^{2\tau} < \tau^{5\tau}$$

where note $r \leq d/2 \ll \tau$ in our parameter regime. ◀

▶ **Lemma 6.9 (Main Lemma).** *In the decomposition* (6.15), *w.p.* $> 1-n^{-5\log n}$ *the following hold. For all* $R \in \binom{[n]}{\le d/2}$*, let* $P^R = \{I \in \binom{[n]}{d/2} \mid R \subseteq I\}$*,*

(1)

$$M_0^R \succeq n^{-d}\mathrm{diag}(\widetilde{\mathrm{Cl}})_{P^R \times P^R}; \tag{6.17}$$

(2)

$$\pm\omega^{-c} M_c^R \preceq n^{-c/6} \cdot M_0^R, \quad \forall 0 < c \le |R|. \tag{6.18}$$

▶ **Corollary 6.10** (Theorem 6.1)**.** *W.p.* $> 1 - n^{-5\log n}$ *over* G*,*

$$M(G) \succeq n^{-d-1}\mathrm{diag}(\widetilde{\mathrm{Cl}}(G))_{\binom{[n]}{d/2}\times\binom{[n]}{d/2}}.$$

Proof. For each R, by definition $M^R = \sum_{c=0}^{|R|} m_c \circ M_c^R$. Suppose the situation in Lemma 6.9 happens, which has probability $> 1 - n^{-5\log n}$. Since Hadamard product with a PSD matrix presevres PSDness (the Schur product theorem),

$$\begin{aligned}
\sum_{c=1}^{|R|} m_c \circ M_c^R &\preceq \sum_{c=1}^{|R|} m_c \circ \left(\omega^c n^{-c/6} \cdot M_0^R\right) && \text{(Lemma 6.9(2))}\\
&= \left(\sum_{c=1}^{|R|} n^{-c/6} \cdot m_0\right) \circ M_0^R && \text{(Lemma 6.4)}\\
&\preceq n^{-1/6} m_0 \circ M_0^R
\end{aligned}$$

Similarly, $\sum_{c=1}^{|R|} m_c \circ M_c^R \succeq -n^{-1/6} m_0 \circ M_c^R$. So

$$M^R \succeq (1 - n^{-1/6}) m_0 \circ M_0^R \succeq n^{-d-1}\mathrm{diag}(\widetilde{\mathrm{Cl}})_{P^R \times P^R} \quad \text{(Lem. 6.4 and 6.9(2)).}$$

Apply this to (6.15),

$$M = M^{\emptyset} + \sum_{\emptyset \ne R \in \binom{[n]}{\le d/2}} M^R \succeq M^{\emptyset} \succeq n^{-d-1}\mathrm{diag}(\widetilde{\mathrm{Cl}})_{\binom{[n]}{d/2}\times\binom{[n]}{d/2}}. \tag{6.19}$$

◀

The rest of Section 6 is devoted to proving the Main Lemma 6.9, completed in Subsection 6.7. The key ingredient is Lemma 6.21, stated in Section 6.4. The statement requires the recursive factorization of each M_c^R, which we show as Lemma 6.19 in the upcoming Subsections 6.2 and 6.3.

6.2 The first-approximate factorization of M_c^R

In this subsection and the next, we factorize each matrix M_c^R in (6.15) by the recursive approximate factorization.

Terminology established in Section 5.3 will be used. We start by defining the first-approximate factorization (cf. Definition 5.17).

▶ **Definition 6.11.** *Fix $R \in \binom{[n]}{\le \frac{d}{2}}$. For every $i = 0, 1, ..., \tau$ define the left-i-factor $L^{R,i}$ to be the matrix of dimension $\binom{[n]}{\frac{d}{2}} \times \binom{[n]}{\le \frac{d}{2}}$,*

$$L^{R,i}(I, A) = \begin{cases} 0 & , \textit{ if } R \not\subseteq I \cap A; \\ \sum\limits_{\substack{T:\ |V(T) \cup I \cup A| \le \tau \\ A = S_l(I,A;T) \\ T \cap E(A) = \emptyset \\ (I,A;T) \text{ left-generated} \\ e_{I,A}(T) = i}} (\frac{\omega}{n})^i \chi_T & , \textit{ o.w.} \end{cases} \tag{6.20}$$

*$(L^{R,j})^\top$ is called the right-j-factor. Call $\widetilde{L^R} = (L^{R,0}, ..., L^{R,\tau})$ the **left factor**, $(\widetilde{L^R})^\top$ the **right factor**. Note these matrices do not depend on "c".*

▶ **Definition 6.12.** *Let D^τ denote the constant diagonal matrix*

$$\operatorname{diag}\left((\frac{\omega}{n})^{\frac{|A|}{2}} \right)_{A: |A| \le d/2} \otimes \mathrm{Id}_{\{0,...,\tau\} \times \{0,...,\tau\}}$$

of dimension $\left(\binom{[n]}{\le d/2} \times (\tau + 1) \right) \times \left(\binom{[n]}{\le d/2} \times (\tau + 1) \right)$.

▶ **Definition 6.13** (Goal factorization of M_c^R)**.** *Our goal is to find a middle matrix Q_c^R of dimension*

$$\left(\binom{[n]}{\le \frac{d}{2}} \times (\tau + 1) \right) \times \left(\binom{[n]}{\le \frac{d}{2}} \times (\tau + 1) \right)$$

s.t. the following factorization approximately holds:

$$M_c^R \approx \underbrace{(L^{R,0}, ..., L^{R,\tau})}_{\widetilde{L^R}} \cdot \left(D^\tau \cdot Q_c^R \cdot D^\tau \right) \cdot \underbrace{(L^{R,0}, ..., L^{R,\tau})^\top}_{\left(\widetilde{L^R}\right)^\top} \tag{6.21}$$

▶ Remark 6.14. Unlike in the non-exact case (section 5.3), here we factorize M_c^R by further distinguishing a parameter pair in $\{0, ..., \tau\} \times \{0, ..., \tau\}$. The reason is that in (6.13), or more broadly in any exact pseudo-expectation generated by the method in Section 3.2, the parameter

$$a = |V(T) \cup I \cup J|$$

appears nestedly in an essential way.

Fixing $(I, J; T)$, previously the coefficient (3.12) is intended as

$$(\frac{\omega}{n})^a = (\frac{\omega}{n})^{e(\mathcal{R}_l) + |V(\mathcal{R}_m)| + e(\mathcal{R}_r)}$$

as in Remark 5.12, which naturally factors into the left, middle, right terms. Here, however, there are terms like $\binom{a+l-d}{c} \cdot \binom{n-a}{l-c}$ that are not log-additive in a. Also, the reason we chose the d-generating function as in Def. 3.11 is exactly to prove the positiveness of $\mathbb{E}[Q_{0,0}^R]$ in this harder situation. This is eventually made clear by Prop. 6.28 and Cor. 6.30.

To approach the goal decomposition (6.21), in the coefficients in M_c^R (6.16) we separate the main factor

$$(\frac{\omega}{n})^a = (\frac{\omega}{n})^{e(\mathcal{R}_l)} \cdot (\frac{\omega}{n})^{|V(\mathcal{R}_m)|} \cdot (\frac{\omega}{n})^{e(\mathcal{R}_r)}$$

into left, right, and middle factors as before, while leave the factor $Y_c(r, a)$ for the middle matrix $Q_c^R\Big((\cdot, e_l), (\cdot, e_r)\Big)$ to bear , where the index (e_l, e_r) has the natural intended meaning.

▶ **Definition 6.15** (First-approximate factorization by $Q^R_{c,0}$)**.** *Define $Q^R_{c,0}$ to be the $\{0,...,\tau\}\times\{0,...,\tau\}$-block matrix, each block of dimension $\binom{[n]}{\le d/2}\times\binom{[n]}{\le d/2}$, that is 0 outside of the principal minor*

$$S^R \times S^R, \quad S^R = \{(A,i)\in \binom{[n]}{\le d/2}\times\{0,...,\tau\} \mid A\supseteq R, |A|+i\ge \frac{d}{2}\}, \tag{6.22}$$

and in this principal minor, $Q^R_{c,0}\Big((A,i),(B,j)\Big) =$

$$\sum_{\substack{T_m:|V(T_m)\cup A\cup B|\le\tau\\ A,B\in \mathrm{mSep}_{A,B}(T_m)}} (\frac{\omega}{n})^{|V(T_m)\cup A\cup B|-\frac{|A|+|B|}{2}} \cdot \underbrace{Y_c\Big(|R|,\ |V(T_m)\cup A\cup B|+(i+j)\Big)}_{\text{defined by (6.13)}} \cdot \chi_{T_m} \tag{6.23}$$

Correspondingly, define

$$\widetilde{L^R}\cdot\left(D^\tau\cdot Q^R_{c,0}\cdot D^\tau\right)\cdot\left(\widetilde{L^R}\right)^\top$$

to be the ***first approximate factorization*** *of M^R_c.*

Some remarks on the definition of $Q^R_{c,0}$ follow.

▶ Remark 6.16 (Intended meaning of parameters in $Q^R_{c,0}$).

(1) The set S^R (6.22) is defined independently of c, where the condition $|A|+i\ge d/2$ is natural because of the intended meaning of i: it is intended as $|V(T')\backslash A|\ge|I|-|A|$ for some ribbon $(I,A;T')$ in $\widetilde{L^R}$. If $|A|+i<d/2$ the corresponding column in $\widetilde{L^R}$ is always 0. Similarly for j.

(2) By definition, $Q^R_{c,0}$ is supported only on those $((A,i),(B,j))\in S^R\times S^R$ with $|A|=|B|$.

(3) Regarding (6.23), as before by Remark 5.12, in "canonical" situations i.e. for outer-canonical products in $\widetilde{L^R}\cdot\left(D^\tau\cdot Q^R_{c,0}\cdot D^\tau\right)\cdot\left(\widetilde{L^R}\right)^\top$,

$$|V(T_m)\cup A\cup B|+(i+j)=|V(T)\cup I\cup J|$$

for ribbons $(I,J;T)$ that take $(A,B;T_m)$ as the middle part of its canonical decomposition and for which $e(\mathcal{R}_l)=i$, $e(\mathcal{R}_r)=j$.

Recall the terminology on the $XYX^\top$-type matrix product, Def 5.15.

▶ **Lemma 6.17** ($Q^R_{c,0}$ indeed gives the first-approximation)**.** *Fix R, $c\le|R|$. For every $(I,J;T)$ s.t. $|V(T)\cup I\cup J|\le\tau$ and $R\subseteq I\cap J$, there is exactly one outer-canonical product in the $XYX^\top$-type matrix product*

$$\widetilde{L^R}\cdot\underbrace{\left(D^\tau\cdot Q^R_{c,0}\cdot D^\tau\right)}_{Y}\cdot\left(\widetilde{L^R}\right)^\top \tag{6.24}$$

which corresponds to the canonical decomposition of $(I,J;T)$, and which gives term

$$M^R_c(I,J;T)\chi_T.$$

Proof. Suppose $R\subseteq I\cap J$. First, note every triple in (6.24) is inner-canonical by definition of $\widetilde{L^R}, Q^R_{c,0}$, so all outer-canonical triples there 1-1 correspond to their triple-product $(I,J;T)$ via the canonical decomposition.

Fix an $(I, J; T)$ and its canonical decomposition, where $|V(T) \cup I \cup J| \le \tau$. $(I, A; T')$ appears exactly once in $\widetilde{L^R}(I, A)$ in block L^{R,e_l}, where $e_l = e_{I,A}(T')$; similarly for $(J, B; T'')$ and $e_r = e_{J,B}(T'')$. And further there is exactly one outer-canonical product in (6.24) corresponding to this triple, with coefficient

$$L^{R,e_l}(I, A; T') \cdot (\frac{\omega}{n})^{\frac{|A|}{2}} \cdot Q^R_{c,0}(A, B; T_m) \cdot (\frac{\omega}{n})^{\frac{|B|}{2}} \cdot L^{R,e_r}(J, B; T''). \tag{6.25}$$

By definition (6.20), (6.23), if $a := |V(T)| \cup I \cup J \le \tau$ then the above coefficient is

$$(\frac{\omega}{n})^a \cdot Y_c(|R|, a) = M^R_c(I, J; T),$$

by comparing (6.13) and (6.16), noticing that

$$a = |V(T) \cup I \cup J| \overset{(\#)}{=} e_l + |V(T_m) \cup A \cup B| + e_r,$$

where (#) is by canonicality. This proves the lemma. ◀

▶ **Definition 6.18** (First error-matrices). *Let $\mathcal{E}_{c,1;\text{negl}}$ be the matrix of the sum of all outer-canonical products in* (6.24) *that exceeds degree, i.e. the resulting*

$$|V(T) \cup I \cup J| > \tau.$$

Let $[\widetilde{L^R} \cdot (D^\tau Q^R_{c,0} D^\tau) \cdot (\widetilde{L^R})^\top]_{\text{non-can}}$ be the matrix of the sum of all products that is non-outer-canonical.

Lemma 6.17 can be restated in the terminology of *approximate form* (Def. 5.15): $\forall R \in \binom{[n]}{\le d/2}$ and $0 \le c \le |R|$,

$$M^R_c = [\widetilde{L^R} \cdot \left(D^\tau Q^R_{c,0} D^\tau\right) \cdot \left(\widetilde{L^R}\right)^\top]_{\text{can}}$$

Equivalently,

$$M^R_c = \widetilde{L^R} \cdot \left(D^\tau Q^R_{c,0} D^\tau\right) \cdot \left(\widetilde{L^R}\right)^\top - [\widetilde{L^R} \cdot \left(D^\tau Q^R_{c,0} D^\tau\right) \cdot \left(\widetilde{L^R}\right)^\top]_{\text{non-can}} - \mathcal{E}^R_{c,1;\text{deg}}. \tag{6.26}$$

As we will see, the crucial fact is that the error matrix $\mathcal{E}_{c,1;\text{main}}$ factorizes through $\widetilde{L^R}, (\widetilde{L^R})^\top$ approximately too, as in the non-exact case. In the next subsection, we show how the recursive factorization method works here in an extended form.

6.3 Recursive factorization: exact case

The main result of this subsection is the following lemma.

▶ **Lemma 6.19** (Recursive approximate factorization; exact case). *For any fixed $R \in \binom{[n]}{\le d/2}$ and $0 \le c \le |R|$, we have the following decomposition.*

$$M^R_c = \widetilde{L^R} \cdot \left[D^\tau \left(Q^R_{c,0} - Q^R_{c,1} + ... \pm Q^R_{c,d}\right) D^\tau\right] \cdot \left(\widetilde{L^R}\right)^\top + \mathcal{E}^R_c, \tag{6.27}$$

where:

(1) *All $Q^R_{c,k}$'s are supported on the principal minor $S^R \times S^R$, where recall*

$$S^R = \{(A, i) \in \binom{[n]}{\le d/2} \times \{0, ..., \tau\} \mid A \supseteq R,\ |A| + i \ge d/2\}.$$

(2) *$Q^R_{c,0}$ is by Definition 6.15;*

(3) *$\forall 1 < k \le d/2$, $Q^R_{c,k}$ is a $(\tau+1)\times(\tau+1)$-block-matrix with the (i,j)-block*

$$Q^R_{c,k}\Big((A,i),(B,j)\Big) = \sum_{T_m:|V(T_m)\cup A\cup B|\le\tau} q^R_{c,k}(\mathcal{R}_m,i,j)\cdot\chi_{T_m} \tag{6.28}$$

(within $S^R\times S^R$), where we naturally denote $\mathcal{R}_m=(A,B;T_m)$; these $q^R_{c,k}(\cdot,i,j)$'s are symmetric w.r.t. shapes, and

$$\forall(i,j)\quad |q^R_{c,k}(\mathcal{R}_m,i,j)| \le \tau^{5\tau}\cdot(\frac{\omega}{n^{1-\epsilon}})^{s-p+k/3}, \tag{6.29}$$

where as usual $s=\frac{|A|+|B|}{2}$, p is the max number of vertex-disjoint paths from A to B in $\mathcal{R}_m$.

(4) *For any G, $\mathcal{E}^R_c(G)$ is supported within rows and columns that is clique in G and contains R. Moreover, w.p. $>1-n^{-9\log n}$,*

$$\|\mathcal{E}^R_c\| < n^{-\epsilon\tau/2}. \tag{6.30}$$

Proof of Lemma 6.19 Like before, the key is to look at one round of the factorization. The following lemma is strictly parallel to Lemma 5.25. Again fix $R\subseteq\binom{[n]}{d/2}$, $c\le|R|$; for convenience denote $n_1 := \binom{[n]}{d/2}\times(\tau+1)$ in the following.

▶ **Lemma 6.20** (One round of factorization; exact case)**.** *Let $\widetilde{L^R}$ be from Def. 6.11, Q^R be any $n_1\times n_1$-matrix supported on $S^R\times S^R$ and*

$$Q^R((A,i),(B,j)) = \sum_{T_m:\ |V(T_m)\cup A\cup B|\le\tau} (\frac{\omega}{n})^{|V(\mathcal{R}_m)|} q(\mathcal{R}_m,i,j)\cdot\chi_{T_m} \tag{6.31}$$

where $\mathcal{R}_m$ denotes $(A,B;T_m)$, and $q(\cdot,i,j)$ is symmetric w.r.t. shapes for any fixed (i,j). Now we define matrix $Q',\mathcal{E}'_{\text{negl}}$ so that the following holds:

$$[\widetilde{L^R}\cdot Q\cdot(\widetilde{L^R})^\top]_{\text{non-can}} = [\widetilde{L^R}\cdot Q'\cdot(\widetilde{L^R})^\top]_{\text{can}} + \mathcal{E}'_{\text{negl}}. \tag{6.32}$$

Namely, let Q' be supported on $S^R\times S^R$,

$$Q'((A,i),(B,j)) = \sum_{T_m:\ |V(T_m)\cup A\cup B|\le\tau} (\frac{\omega}{n})^{|V(\mathcal{R}_m)|} q'(\mathcal{R}_m,i,j)\cdot\chi_{T_m} \tag{6.33}$$

where the coefficients $q'(\mathcal{R}_m,i,j)$ are as follows. Fix any $\mathcal{R}_m=(A,B;T_m)$ and (i,j). Let $t=|V(\mathcal{R}_m)|\le\tau$, $s=\frac{|A|+|B|}{2}$. For every generalized ribbon $\mathcal{R}^*_m$ that contains $\mathcal{R}_m$ as its largest ribbon and $|V(\mathcal{R}^*_m)|\le\tau$, fix any a ribbon pair $(\mathcal{R}'_l,\mathcal{R}'_r)$ so that $(\mathcal{R}'_l,\mathcal{R}^*_m,\mathcal{R}'_r)$ is the separating factorization for some ribbon triple, $|V(\mathcal{R}'_l)|,|V(\mathcal{R}'_r)|\le\tau$ ***and***

$$(e(\mathcal{R}'_l),e(\mathcal{R}'_r)) = (i,j). \tag{6.34}$$

If there is no such choice, exclude this $\mathcal{R}^*_m$ in the summation below. Then:

$$\begin{aligned} q'(\mathcal{R}_m,i,j) &= \sum_{\substack{\mathcal{R}^*_m:\text{ gen. ribbon on }(A,B)\\ |V(\mathcal{R}^*_m)|\le\tau\\ \text{largest ribbon is }\mathcal{R}_m}} (\frac{\omega}{n})^{|\mathcal{I}(\mathcal{R}^*_m)|}\cdot q''(\mathcal{R}^*_m,i,j)\quad\textit{where}\\ q''(\mathcal{R}^*_m,i,j) &= \sum_{\substack{(z,i_1,j_1):\\ 1\le z\le d/2}} \sum_{\substack{\mathcal{P}=(\mathcal{R}_l,\mathcal{R},\mathcal{R}_r):\text{ side-inn. can.}\\ \mathcal{P}\to(\mathcal{R}'_l,\mathcal{R}^*_m,\mathcal{R}'_r)\text{ for the fixed }\mathcal{R}'_l,\mathcal{R}'_r\\ z(\mathcal{P})=z,\ e(\mathcal{R}_l)=i_1,e(\mathcal{R}_r)=j_1}} (\frac{\omega}{n})^z\cdot q(\mathcal{R},i_1,j_1). \end{aligned} \tag{6.35}$$

Note $q''(\mathcal{R}_m, i, j)$ doesn't depend on the choice $(\mathcal{R}'_l, \mathcal{R}'_r)$ by ***(the full of)*** *Proposition 5.24. Thus $q'(\cdot, i, j)$ is also symmetric w.r.t. shapes.*

$\mathcal{E}'_{\text{negl}}$ *is defined s.t.* (6.32) *holds. Then the conclusions are:*

(1) *W.p. $> 1 - n^{-9\log n}$ over G,*

$$\left\|\mathcal{E}'_{\text{negl}}\right\| \leq \max\{q(\cdot)\} \cdot n^{-\epsilon\tau};$$

(2) *If there is a number C for which*

$$\forall \mathcal{R}_m, i, j \quad |q(\mathcal{R}_m, i, j)| \leq C \cdot (\frac{\omega}{n^{1-\epsilon}})^{s-p} \tag{6.36}$$

where p denotes the maximum number of vertex-disjoint paths between A, B in $\mathcal{R}_m$, then

$$\forall \mathcal{R}_m, i, j \quad |q'(\mathcal{R}_m)| \leq C \cdot (\frac{\omega}{n^{1-\epsilon}})^{s-p+1/3}.$$

Proof of Lemma 6.20. The proof is almost the same as that of Lemma 5.25; we point out and explain the differences below.

The support condition (i.e. supported on $S^R \times S^R$) doesn't affect anything since $\widetilde{L^R}$ itself is automatically 0 on columns and rows that are not in S^R.

As step (0) like before, we expand $[\widetilde{L^R} \cdot Q' \cdot (\widetilde{L^R})^\top]_{\text{can}}$ to compare with $[\widetilde{L^R} \cdot Q \cdot (\widetilde{L^R})^\top]_{\text{non-can}}$ term-wise, using Prop. 5.24. Here, notice that when (i, j) and $\mathcal{R}^*_m$ are fixed, the size of any choice of $(\mathcal{R}'_l, \mathcal{R}'_r)$ satisfying (6.34) are also fixed, so the proposition is applicable. The comparison for order on $(\frac{\omega}{n})$ between the two is exactly the same as in step (0) of the proof of Lemma 5.25, and the conclusion is that the matrix $\mathcal{E}'_{\text{negl}}$ collects all terms in $[\widetilde{L^R} \cdot Q \cdot (\widetilde{L^R})^\top]_{\text{non-can}}$ whose $\mathcal{R}^*_m$ in the separating factorization exceeds size τ, i.e. $\mathcal{E}'_{\text{negl}}(I, J) =$

$$\sum_{i,j} \sum_{\substack{(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r)\text{: side. inn. can.} \\ \text{non-outer-can.} \\ \text{all three has size } \leq \tau \\ |V(\mathcal{R}^*_m)| > \tau,\ (e(\mathcal{R}_l), e(\mathcal{R}_r)) = (i,j)}} (\frac{\omega}{n})^{|V(\mathcal{R}_l)| + |V(\mathcal{R}_m)| + |V(\mathcal{R}_r)| - |A| - |B|} q(\mathcal{R}_m, i, j) \chi_T \tag{6.37}$$

where $T = T_l \oplus T_m \oplus T_r$, and we omit writing the default requirement that $\mathcal{R}_l$ ($\mathcal{R}_r$) has the left (right) distinguished vertex set I (J).

The numerical conclusions (1), (2) follow from the same estimates as in Lemma 5.25 (after (5.44) there). We only point out that, for (1), the estimate there is actually loose enough s.t. with even an extra $(1+\tau)^2$-factor (from union bound on blocks) it is still smaller than $n^{-\epsilon\tau}$. ◀

Now we can prove Lemma 6.19.

Proof for Lemma 6.19. Apply the one-round factorization Lemma 6.20 to

$$[\widetilde{L^R} \cdot \left(D^\tau Q^R_{c,i} D^\tau\right) \cdot \left(\widetilde{L^R}\right)^\top]_{\text{non-can}}$$

for $i = 0$, we get $Q^R_{c,1}$, $\mathcal{E}'_{1;\text{negl}}$ (for ease of notation, we hide the index R, c for this negligible matrix). Then repeat this for $i = 1$ we get $\mathcal{E}_{c,1;\text{deg}}$, $Q^R_{c,2}$, and $\mathcal{E}'_{2,\text{negl}}$. Continuing this, as the result we get the recursive factorization

$$\begin{aligned} M^R_c = &\widetilde{L^R} \cdot \left[D^\tau \left(Q^R_{c,0} - Q^R_{c,1} + ... \pm Q^R_{c,d}\right) D^\tau\right] \cdot \left(\widetilde{L^R}\right)^\top - \\ &\left(\mathcal{E}^R_{c,1;\text{deg}} - \mathcal{E}^R_{c,2;\text{deg}} + ... \pm \mathcal{E}^R_{c,d;\text{deg}}\right) + \left(\mathcal{E}'_{1;\text{negl}} + ... + \mathcal{E}'_{d;\text{negl}}\right). \end{aligned} \tag{6.38}$$

Again, here it uses that $Q^R_{c,d+1} = 0$, by the same proposition 5.26.

(1) All $Q^R_{c,k}$ is supported within $S^R \times S^R$ by definition of each round (Lemma 6.20);

(2) By definition.

(3) The coefficients of each $Q^R_{c,k}$ $(k = 0, 1, ..., d)$, $\{q^R_{c,k}(\cdot, i, j)\}$ is always symmetric w.r.t. shapes from Lemma 6.20. Moreover, from definition (6.23),

$$\forall \mathcal{R}_m, i, j \quad |q^R_{c,0}(\mathcal{R}_m)| = |Y_c(|R|, |\mathcal{R}_m|)| \le \tau^{5\tau}$$

where the last one is by Lemma 6.8(4). Since $Q^R_{c,0}$ is special in that for all $\mathcal{R}_m = (A, B; T_m)$ appearing in it, there are $|A| = |B|$ many vertex-disjoint paths between A, B in $\mathcal{R}_m$, i.e. $s = p$, where as usual when $\mathcal{R}_m$ is fixed we use $s = \frac{|A|+|B|}{2}$ and p denotes the max number of vertex-disjoint paths between A, B. So the above can be equivalently written as

$$\forall \mathcal{R}_m, i, j \quad |q^R_{c,0}(\mathcal{R}_m)| \le (\frac{\omega}{n^{1-\epsilon}})^{s-p} \tau^{5\tau}. \tag{6.39}$$

Now use Lemma 6.20(2), where notice the "$q(\cdot)$" in there corresponds to $q^R_{c,k}$ here, since the "Q" matrix is $D^\tau Q^R_{c,k} D$ so the "$(\frac{\omega}{n})^{|V(\mathcal{R}_m)|} q(\cdot)$" is $(\frac{\omega}{n})^{|V(\mathcal{R}_m)|-s} \cdot (\frac{\omega}{n})^s \cdot q^R_{c,k}$. As the result, we get the recursive bound

$$\forall \mathcal{R}_m, i, j \quad |q^R_{c,k}(\mathcal{R}_m, i, j)| \le \tau^{5\tau} \cdot (\frac{\omega}{n^{1-\epsilon}})^{s-p+k/3}.$$

(4) First, when plugged in any G, both

$$M^R_c \quad \text{and} \quad \widetilde{L^R} \cdot \left[D^\tau \left(Q^R_{c,0} - Q^R_{c,1} + ... \pm Q^R_{c,d} \right) D^\tau \right] \cdot \left(\widetilde{L^R} \right)^\top$$

are supported within clique rows and columns that contain R by their definition. So it must be the case for their difference, $\mathcal{E}^R_c$, too. Next we only need to give the norm bound. By (6.38), the final error matrix is

$$\mathcal{E}^R_c = -\left(\mathcal{E}^R_{c,1;\text{deg}} - \mathcal{E}^R_{c,2;\text{deg}} + ... \pm \mathcal{E}^R_{c,d;\text{deg}}\right) + \left(\mathcal{E}'_{1;\text{negl}} + ... + \mathcal{E}'_{d;\text{negl}}\right).$$

Note by Lemma 6.20(2), by induction all $|q^R_{c,k}| < \tau^{5\tau}$. For each $\mathcal{E}'_{k;\text{negl}}$, by Lemma 6.20(1) w.p. $> 1 - n^{-9 \log n}$, $\left\| \mathcal{E}'_{k;\text{negl}} \right\| < \tau^{5\tau} n^{-\epsilon\tau} < n^{-0.9\epsilon\tau}$.

As for $\mathcal{E}^R_{c,k;\text{deg}}$, recall by definition 5.15 on (I, J) it is the sum of outer-canonical products in $\widetilde{L^R} \cdot (D^\tau Q^R_{c,i-1} D^\tau) \cdot \left(\widetilde{L^R}\right)^\top (I, J)$ s.t. $|V(T) \cup I \cup J| > \tau$. So

$$\mathcal{E}^R_{c,k;\text{deg}}(I, J) = \sum_{\substack{(\mathcal{R}_l, \mathcal{R}_m, \mathcal{R}_r): \\ \text{semi-inn.can.} \\ \text{outer.can.} \\ |V(T) \cup I \cup J| > \tau}} (\frac{\omega}{n})^{|V(T) \cup I \cup J|} \cdot q^R_{c,k-1}(\mathcal{R}_m, e(\mathcal{R}_l), e(\mathcal{R}_r)) \chi_T$$

where as usual $s = s(\mathcal{R}_m)$ is the average of its two side vertex-sets, $T = T_l \oplus T_m \oplus T_r$, and in the summation R_l (R_r) should have I (J) as the left (right) set. Note the above uses $|V(T) \cup I \cup J| = e_l + e_r + |V(\mathcal{R}_m)|$ from the outer- and semi-inner- canonicality. Moreover, any fixed $(I, J; T)$ can come from at most $3^{3\tau}$ triples as their vertex set union is $|V(T) \cup I \cup J|$ by canonicality. Since $3\tau \ge |V(T) \cup I \cup J| > \tau$ and w.h.p. $|q^R_{c,k-1}(\cdot)| < \tau^{5\tau}$, use Lemma 4.2 and we get that w.p. $> 1 - n^{-10 \log n}$,

$$\left| \mathcal{E}^R_{c,k;\text{deg}}(I, J) \right| < \tau^{6\tau} \cdot \sum_{c=0}^{3\tau} (\frac{\omega}{n})^{\max\{\tau, c\}} \cdot (n^{c/2} 2^{c^2} n^{4 \log \log n}) < n^{-2\epsilon\tau}.$$

So by union bound over (I, J), $\left\| \mathcal{E}^R_{c,k;\text{deg}} \right\| < n^{-d/4} n^{-2\epsilon\tau} < n^{-\epsilon\tau}$ w.p. $> 1 - n^{-9.5 \log n}$.

Together, sum the two and by union bound over k, we get that w.p. $> 1 - n^{-9 \log n}$, $\left\| \mathcal{E}^R_c \right\| < n^{-\epsilon\tau/2}$. ◀

6.4 Positiveness of the middle matrices: proof overview

Now we use the approximate decomposition of M^R_c's to prove the Main Lemma 6.9. Recall for each R, $c \le |R|$, by Lemma 6.19

$$M^R_c = \widetilde{L^R} \cdot \left[D^\tau \underbrace{\left(Q^R_{c,0} - Q^R_{c,1} + ... \pm Q^R_{c,d}\right)}_{:=Q^R_c} D^\tau \right] \cdot \left(\widetilde{L^R}\right)^\top + \mathcal{E}^R_c.$$

The key is the following lemma. Recall $S^R = \{(A,i) \in \binom{[n]}{\le d/2} \times \{0,...,\tau\} \mid A \supseteq R, |A| + i \ge \frac{d}{2}\}$.

▶ **Lemma 6.21.** *W.p.* $> 1 - n^{-8\log n}$ *over* G*, the following holds.*

(1) $\forall R \in \binom{[n]}{\le d/2}$,

$$Q^R_{0,0} - Q^R_{0,1} + ... \pm Q^R_{0,\frac{d}{2}} \;\succeq\; \tau^{-7\tau} \cdot \operatorname{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R \times S^R},$$

where recall $S^R = \{(A,i) \in \binom{[n]}{\le d/2} \times \{0,...,\tau\} \mid A \supseteq R, |A| + i \ge \frac{d}{2}\}$.

(2) $\forall R$, $0 < c \le |R|$

$$\pm\omega^{-c}\left(Q^R_{c,0} - Q^R_{c,1} + ... \pm Q^R_{c,\frac{d}{2}}\right) \;\preceq\; n^{-c/4} \cdot \operatorname{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R \times S^R}.$$

The proof of Lemma will span the upcoming three subsections, completed at the end of Section 6.6. The Main Lemma 6.9 then follows by standard steps (Section 6.7).

Proof plan for Lemma 6.21. Fix an $R \in \binom{[n]}{\le d/2}$. We will prove the lemma by three ingredients: Corollary 6.36, Lemma 6.37, Lemma 6.38.

Corollary 6.36 (in Section 6.5, 6.6): Positiveness of $Q^R_{0,0}$. This is the last real technical challenge. We use a natural "*structural part + pseudo-random part*" decomposition of $Q^R_{0,0}$ (Def. 6.23), aiming to show that on their common support, the structural part is positive enough and the pseudo-random part is small enough in norm. The main difficulty here is in analyzing $\mathbb{E}[Q^R_{0,0}]$ which, ultimately, is about the choice of generating function F in Definition 3.11.

Lemma 6.37, 6.38 (Section 6.6): Other $Q^R_{c,k}$'s ($k > 0$ or $c > 0$), when timed with ω^{-c}, are small and appropriately supported. These two lemmas are proved by standard means.

We will follow this plan in the next two subsections. Here we end this subsection with two definitions for preparation.

▶ **Definition 6.22.** *Let the* ***root diagonal-clique matrix*** *be*

$$D_{\mathrm{Cl}}(A,B) = \begin{cases} 0 & \text{, } \textit{if } A \ne B; \\ 2^{-\binom{|A|}{2}/2} \cdot \widetilde{\mathrm{Cl}}_A = 2^{-\binom{|A|}{2}/2} \sum_{T \subseteq E[A]} \chi_T & \text{, } \textit{o.w.} \end{cases} \tag{6.40}$$

of dimension $\binom{[n]}{\le d/2} \times \binom{[n]}{\le d/2}$*, so that* $D^2_{\mathrm{Cl}}(A,A) = \widetilde{\mathrm{Cl}}(A)$ *for all* $A \in \binom{[n]}{d/2}$*. Define*

$$D^\tau_{\mathrm{Cl}} := D_{\mathrm{Cl}} \otimes \mathrm{Id}_{\{0,...,\tau\} \times \{0,...,\tau\}}. \tag{6.41}$$

which is also diagonal.

▶ **Definition 6.23.** *The* ***structural-pseudorandom decomposition*** *of* $Q^R_{0,0}$ *is*

$$Q^R_{0,0} = D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{Cl} + \left(Q^R_{0,0} - D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{Cl}\right), \tag{6.42}$$

where the summand $D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{Cl}$ *is called the* ***structural part****, and the summand* $\left(Q^R_{0,0} - D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{Cl}\right)$ *the* ***pseudo-random part****.*

6.5 Positiveness of $\mathbb{E}[Q^R_{0,0}]$

▶ **Proposition 6.24** (Expression of $\mathbb{E}[Q^R_{c,0}]$). *Fix $R \in \binom{[n]}{\le d/2}$ and $0 \le c \le |R|$. let $r = |R|$. Recall S^R is defined by (6.22).*

(1) *$\mathbb{E}[Q^R_{c,0}]$ is supported on the blockwise partial-diagonals*

$$\{\Big((A,i),(A,j)\Big) \in S^R \times S^R\}.$$

(i.e. requires $R \subseteq A$ and $|A| + \min\{i,j\} \ge d/2$)

(2) *For all $\Big((A,i),(A,j)\Big) \in S^R \times S^R$, $\mathbb{E}[Q^R_{c,0}]\Big((A,i),(A,j)\Big) =$*

$$\sum_{l=c}^{r}(-1)^{r-l}\frac{\binom{r}{l}}{(l-c)!}\binom{|A|+i+j+l-d}{c}\frac{\Big(|A|+8\tau^2+(l-c)+(i+j)\Big)!}{(8\tau^2)!} + O\left(\frac{\tau^{1.5\tau}}{n}\right). \tag{6.43}$$

In particular, for $c = 0$,

$$\mathbb{E}[Q^R_{0,0}]\Big((A,i),(A,j)\Big) = \sum_{l=0}^{r}(-1)^{r-l}\frac{\binom{r}{l}}{l!}\cdot\frac{\Big(|A|+8\tau^2+l+(i+j)\Big)!}{(8\tau^2)!} + O\left(\frac{\tau^{1.5\tau}}{n}\right). \tag{6.44}$$

(3) *For every $A \in \binom{[n]}{\le d/2}$ let $1_{A,A}$ be the $\binom{[n]}{\le d/2} \times \binom{[n]}{\le d/2}$-matrix with a single 1 on position (A,A). Then*

$$\mathbb{E}[Q^R_{0,0}] = \sum_{\substack{A\subseteq\binom{[n]}{\le d/2}\\ A\supseteq R}} 1_{A,A} \otimes \left[\left(\sum_{l=0}^{r}(-1)^{r-l}\frac{\binom{r}{l}}{l!}\cdot P_{|A|+l}\right) + E^R_A\right] \tag{6.45}$$

where, for every fixed A, $P_{|A|+l} and E^R_A$ are $(\tau+1)\times(\tau+1)$-matrices both supported on the principal minor $\{i \mid d/2 - |A| \le i \le \tau\} \times \{i \mid d/2 - |A| \le i \le \tau\}$ with the following property:

$$\|E^R_A\| < \frac{\tau^{2\tau}}{n}, \tag{6.46}$$

and

$$P_{|A|+l}(i,j) = \frac{\Big(|A|+l+8\tau^2+(i+j)\Big)!}{(8\tau^2)!}, \quad d/2 - |A| \le i,j \le \tau. \tag{6.47}$$

Proof. For (1), the constant terms in (6.23) correspond to $T_m = \emptyset$, which is nonzero only when $A = B$ for A, B in S^R.

For (2), from definition (6.23) we notice again $T_m = \emptyset$ and $A = B$. $\mathbb{E}[Q^R_{c,0}((A,i),(A,j))] = Y_c(\underbrace{|R|}_{:=r}, \underbrace{|A|+i+j}_{:=a})$, which expands to:

$$\sum_{l=c}^{r}(-1)^{r-l}\binom{r}{l}\underbrace{\binom{a+l-d}{c}}_{\text{Def. 6.5}}\binom{n-a}{l-c}n^{-(l-c)}\frac{(a+l-c+8\tau^2)!}{(8\tau^2)!}. \tag{6.48}$$

Now use

$$\binom{n-a}{l-c}n^{-(l-c)} = \frac{1}{(l-c)!}\frac{(n-a)...(n-a-(l-c)+1)}{n^{l-c}} = \frac{1}{(l-c)!}(1-O(d^2/n))$$

and

$$\left|\binom{r}{l}\binom{a+l-d}{c}\binom{n-a}{l-c}n^{-(l-c)}\frac{(a+l-c+8\tau^2)!}{(8\tau^2)!}\right| < (4d)^d \cdot (9\tau^2)^d < \tau^\tau$$

to (6.48), we get (6.43). Further, in (6.48) when $c = 0$ we have $\binom{a+l-d}{0} = 0$ regardless of $a + l - d$ (any value of it, positive, negative or 0). And the same analysis gives (6.44).

For (3), each E^R_A has dimension $(\tau+1)\times(\tau+1)$ and each entry is absolutely $< \tau^{1.5\tau}/n$ from part (2). ◀

▶ Remark 6.25 (Specialty of $c = 0$). Comparing $\mathbb{E}[Q^R_{0,0}]$ and $\mathbb{E}[Q^R_{c,0}]$ (6.43), (6.44), the specialty of the case $c = 0$ is that the factor $\binom{|A|+l-d}{0}$ is **always** 1, which is important for $\mathbb{E}[Q^R_{0,0}]$ to be positive. In cases $c > 0$, $\binom{|A|+l-d}{c}$ might be 0 or negative depending on the order between $0, c, |A| + l - d$, making $\mathbb{E}[Q^R_{c,0}]$ possibly not PSD.

▶ **Definition 6.26.** *For every $m, t \in \mathbb{N}$, define the **factorial Hankel matrix** to be*

$$H_{m,t}(i,j) = (i+j+t)! \quad \forall 0 \le i,j \le m. \tag{6.49}$$

The following is our key observation on the structure of these matrices.

▶ **Proposition 6.27** (Almost common decomposition of $\{H_{m,t}\}$).

(1) *The matrix family $\{H_{m,t}\}$ have decomposition*

$$H_{m,t} = L_m \cdot \left(N_{m,t}\cdot D_{m,t}\cdot (N_{m,t})^\top\right)\cdot (L_m^\top)$$

where $L_m, D_{m,t}$ are diagonal, $N_{m,t}$ is lower-triangular

$$L_m(i,i) = i! \qquad D_{m,t}(i,i) = \prod_{t'=1}^{t}(i+t') \qquad N_{m,t}(i,j) = \binom{i+t}{i-j}$$

In particular, L_m is independent of t, and $H_{m,t}$ is positive.

(2) *Let J_m denote the $(1+m)\times(1+m)$ lower-triangular Jordan block*

$$J_m(i,j) = \begin{cases} 1 & \text{, if } i = j \text{ or } i = j+1; \\ 0 & \text{, o.w.} \end{cases}$$

Then the "left factors" $N_{m,t}$ satisfy the recursive relation

$$N_{m,t+1} = N_{m,t}\cdot J_m. \tag{6.50}$$

Proof. This follow from a direct inspection. ◀

▶ **Proposition 6.28.** *If parameters m, t, r satisfy*

$$t + 1 > 8 \cdot \max\{r^2, m\} \tag{6.51}$$

then

$$H_{m,t+1} \succeq 2r^2 H_{m,t}.$$

Proof. By Proposition 6.27 it suffices to show that under (6.51),

$$J_m \cdot D_{m,t+1} \cdot J_m^\top \succeq 2r^2 D_{m,t}.$$

Equivalently, we need to compare the quadratic forms for fixed m:

$$q_{t+1}(x) := (x^\top J_m) D_{m,t+1} (J_m^\top x) \quad \text{v.s.} \quad q_t(x) := 2r^2 \cdot x^\top D_{m,t} x \tag{6.52}$$

where $x^\top = (x_0, ..., x_m)$ is the formal variable row-vector. Define polynomials

$$\alpha(y) = 2r^2 \prod_{t'=1}^{t} (y + t'), \quad \beta(y) = \prod_{t'=1}^{t+1} (y + t').$$

By definition of $D_{m,t}$, J_m,

$$q_{t+1}(x) = \sum_{i=0}^{m} \beta(i)(x_i + x_{i+1})^2, \quad x_{m+1} := 0;$$
$$q_t(x) = \sum_{i=0}^{m} \alpha(i) x_i^2.$$

To compare the two, note

$$q_{t+1}(x) = \sum_{i=0}^{m} \beta(i) \cdot (x_i + x_{i+1})^2 =$$
$$\sum_{i=0}^{m} \left[\alpha(i) x_i^2 + \Big(\beta(i) - \alpha(i)\Big) \cdot (x_i + \frac{\beta(i)}{\beta(i) - \alpha(i)} x_{i+1})^2 \; - \; \frac{\beta(i)^2}{\beta(i) - \alpha(i)} x_{i+1}^2 \right]$$

So if for $0 \le i \le m$ let

$$b_i = 1 - \frac{\alpha(i)}{\beta(i)} - \frac{\beta(i-1)}{\beta(i)} \frac{1}{b_{i-1}}, \quad b_0 = 1 - \frac{\alpha(0)}{\beta(0)}, \tag{6.53}$$

then

$$q_{t+1}(x) = \underbrace{\sum_{i=0}^{m} \alpha(i) x_i^2}_{q_t(x)} \; + \; \sum_{i=0}^{m} \beta(i) b_i \cdot (x_i + \frac{1}{b_i} x_{i+1})^2. \tag{6.54}$$

▷ Claim 6.29. In (6.53), for all $i \le m$ we have $b_i > 1/2$.

Proof. By definition, $b_0 = 1 - \frac{2r^2}{(t+1)}$ and

$$b_i = 1 - \frac{2r^2}{(t + 1 + i)} - \frac{i}{(t + 1 + i)} \cdot \frac{1}{b_{i-1}}, \quad i \ge 1. \tag{6.55}$$

Use induction for the claim: $b_0 = 1 - \frac{2r^2}{t+1} > 1/2$ by (6.51). For $1 \le i \le m$,

$$\begin{aligned} b_i &= 1 - \frac{2r^2}{t+1+i} - \frac{i}{t+1+i} \cdot \frac{1}{b_{i-1}} \\ &\ge 1 - \frac{2r^2}{t+1} - \frac{m}{t+1} \cdot 2 > 1/2 \quad \text{by (6.51) and the inductive hypothesis.} \end{aligned}$$

◁

By (6.54) and positiveness of each b_i (Claim 6.29), $q_{t+1}(x) \ge q_t(x)$. The lemma is proved. ◀

Now we apply Proposition 6.28 to matrices $P_{|A|+l}$ (6.47). Note

$$P_{|A|+l} = \frac{1}{(8\tau^2)!} H_{\tau-(d/2-|A|),\ d-|A|+8\tau^2+l}$$

where A is fixed, l varies; below, we regard $P_{|A|+l}$ as a matrix on its support.

▶ **Corollary 6.30** (Positiveness of $\mathbb{E}[Q^R_{0,0}]$). *In the decomposition* (6.45) *of* $\mathbb{E}[Q^R_{0,0}]$,

$$\left(\sum_{l=0}^{r} (-1)^{r-l} \frac{\binom{r}{l}}{l!} \cdot P_{|A|+l}\right) + E^R_A \ \succ\ \operatorname{diag}\left(\tau^{-6\tau}\right)_{0\le i\le\tau-(d/2-|A|)} \tag{6.56}$$

where we naturally regarded matrices as on their support

$$\{i \mid d/2 - |A| \le i \le \tau)\}^2 \cong \{0, ..., \tau - (d/2 - |A|)\}^2.$$

In particular, by (6.45)

$$\mathbb{E}[Q^R_{0,0}] \ \succ \sum_{\substack{A\subseteq\binom{[n]}{\le d/2} \\ A\supseteq R}} 1_{A,A} \otimes \operatorname{diag}\left(\tau^{-6\tau}\right)_{d/2-|A|\le i\le\tau} = \operatorname{diag}\left(\tau^{-6\tau}\right)_{S^R\times S^R} \tag{6.57}$$

where recall $S^R = \{(A, i) \mid R \subseteq A, |A| + i \ge d/2\}$.

Proof. The "in particular" part is straightforward from (6.56) by checking the support, and that tensoring with a nonzero PSD matrix preserves the relation $\succ$. In below we prove for (6.56).

Fix A, let

$$\tau_0 = \tau - (d/2 - |A|), \quad t_0 = d - |A| + 8\tau^2. \tag{6.58}$$

Then

$$\sum_{l=0}^{r} (-1)^{r-l} \frac{\binom{r}{l}}{l!} \cdot P_{|A|+l} = \frac{1}{(8\tau^2)!} \cdot (X_r + X_{r-2} + ...) \tag{6.59}$$

where, $\forall 0 \le v \le \lfloor r/2 \rfloor$,

$$X_{r-2v} = \frac{\binom{r}{r-2v}}{(r-2v)!} \cdot \left(H_{\tau_0,t_0+r-2v} - \underbrace{\frac{(r-2v)^2}{(2v+1)}}_{\le r^2} H_{\tau_0,t_0+r-2v-1}\right), \quad H_{\tau_0,-1} := 0.$$

Since $t_0 > 8\max\{r^2, \tau_0\}$, by Proposition 6.28

$$X_{r-2v} \succeq \frac{\binom{r}{r-2v}}{(r-2v)!} \cdot \max\{\frac{1}{2} H_{\tau_0,t_0+r-2v},\ r^2 H_{\tau_0,t_0+r-2v-1}\} \quad \forall 0 \le v \le r/2.$$

So in (6.59), in particular,

$$\sum_{l=0}^{r}(-1)^{r-l}\frac{\binom{r}{l}}{l!}\cdot P_{|A|+l} \succeq \frac{1}{(8\tau^2)!}\cdot H_{\tau_0,t_0} \overset{\text{Prop. 6.27}}{=} L\left(N_{t_0}\cdot\frac{D_{t_0}}{(8\tau^2)!}\cdot(N_{t_0})^\top\right)L \tag{6.60}$$

where we temporarily abuse the notation by omitting the index τ_0 in the RHS.

Using the following claim, we can finish the proof of (6.56):

$$\begin{aligned}\text{RHS of (6.60)} &\succ L\cdot\operatorname{diag}\left(\tau^{-5\tau}\right)_{0\le i\le\tau_0}\cdot L \qquad \text{(by Claim 6.31)}\\ &\succeq \operatorname{diag}\left(\tau^{-5\tau}\right)_{0\le i\le\tau_0},\end{aligned}$$

while by Proposition 6.24 (3),

$$\left\|E_A^R\right\| < \frac{\tau^{2\tau}}{n} < \tau^{-6\tau} \qquad \text{(parameter regime)}.$$

So LHS of (6.56) $\succeq$ $\operatorname{diag}\left(\tau^{-5\tau}-\tau^{-6\tau}\right)_{0\le i\le\tau_0}$ $\succeq$ RHS of (6.56). ◀

▷ Claim 6.31. In notation of Corollary 6.30,

$$N_{t_0}^{-1}(i,j) = (-1)^{i-j}\binom{i+t_0}{i-j} \quad 0\le i,j\le\tau_0 \tag{6.61}$$

and

$$N_{t_0}\cdot\frac{D_{t_0}}{(8\tau^2)!}\cdot(N_{t_0})^\top \succ \operatorname{diag}\left(\tau^{-5\tau}\right)_{0\le i\le\tau_0}. \tag{6.62}$$

Proof. For (6.61), multiply this matrix with N_{t_0} then the (i,j)th entry is

$$\sum_{j\le k\le i}(-1)^{i-k}\binom{i+t_0}{i-k}\binom{k+t_0}{k-j} = \sum_{k'=0}^{i'}(-1)^{i'-k'}\binom{i'+j+t_0}{i'-k'}\binom{k'+j+t_0}{k'}$$

where $i'=i-j$, $k'=k-j$. To see this is identity matrix, use generating functions: let $D_m[(1+x)^a]$ denote the coefficient of x^m in $(1+x)^a$, $m\ge0, a\in\mathbb{Z}$, the above RHS is

$$\begin{aligned}&(-1)^{i'}\sum_{k'=0}^{i'}D_{i'-k'}[(1+x)^{i'+j+t_0}]\cdot D_{k'}[(1+x)^{-(t_0+j+1)}]\\ =&(-1)^{i'}D_{i'}[(1+x)^{i'+j+t_0-(t_0+j+1)}] = (-1)^{i'}D_{i'}[(1+x)^{i'-1}] = 1_{i'=0}.\end{aligned}$$

For (6.62), it is equivalent to

$$\frac{D_{t_0}}{(8\tau^2)!} \succ N_{t_0}^{-1}\cdot\tau^{-5\tau}\cdot(N_{t_0}^{-1})^\top. \tag{6.63}$$

To upper bound the RHS, let $a_0=\tau^{-5\tau}$, consider the quadratic form

$$x^\top N_{t_0}^{-1}\cdot a_0\cdot(N_{t_0}^{-1})^\top x = a_0\sum_{j=0}^{\tau_0}y_j^2, \tag{6.64}$$

where by (6.61),

$$y_j = \left(x^\top N_{t_0}^{-1}\right)_j = \sum_{i=j}^{\tau_0}(-1)^{i-j}\binom{i+t_0}{i-j}x_i.$$

By Cauchy-Schwartz, $y_j^2 \le \tau_0 \cdot \sum_{i=j}^{\tau_0} \binom{i+t_0}{i-j}^2 x_i^2$, so

$$\begin{aligned}\text{RHS of (6.64)} = a_0 \sum_{j=0}^{\tau_0} y_j^2 \;&\le\; a_0 \sum_{i=0}^{\tau_0} x_i^2 \cdot \left(\tau_0 \sum_{j=0}^{i} \binom{i+t_0}{i-j}^2\right) \\ &< \sum_{i=0}^{\tau_0} \left(\tau^{-5\tau} \cdot (9\tau^2)^{2i+2}\right) x_i^2 .\end{aligned}$$

Now (6.63) follows since for each i, in the LHS of (6.63)

$$\begin{aligned}\frac{D_{t_0}(i,i)}{(8\tau^2)!} &\ge (8\tau^2)^{-(d/2-|A|)} \qquad \text{(by definition)} \\ &> \tau^{-2d} > \tau^{-5\tau} \cdot (9\tau^2)^{2i+2}\end{aligned}$$

using $i \le \tau_0 < \tau$, $d \ll \tau$. So (6.63) holds. ◁

We get the main conclusion of this subsection:

▶ **Corollary 6.32** (Positiveness of the structural part of $Q^R_{0,0}$ (Def. 6.23)).

$$\underbrace{D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{\mathrm{Cl}}}_{\text{stractural part of } Q^R_{0,0}} \;\succeq\; \tau^{-6\tau} \cdot \operatorname{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R \times S^R} .$$

Proof. This follows from Cor. 6.30 and that $D^2_{\mathrm{Cl}}(A,A) = \widetilde{\mathrm{Cl}}(A)$ for all A in Def. 6.22. ◀

6.6 Rest bounds: $Q^R_{c,k}$s

In this subsection, we bound the rest matrices:

$$\underbrace{Q^R_{0,0} - D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{\mathrm{Cl}}}_{\text{pseudo}-\text{random part of } \mathrm{Q}^{\mathrm{R}}_{0,0} \text{ (Def. 6.23)}} \quad , \quad Q^R_{0,k}\ (k>0), \qquad \omega^{-c} \cdot Q^R_{c,k}\ (c>0, k\ge 0)$$

by three Lemmas 6.34, 6.37, 6.38, respectively, which would prove Lemma 6.21.

The arguments are quite standard but somewhat lengthy, as one needs to be careful on the block structure and the support of the matrices.

▶ **Definition 6.33** (0-1 diagonal-clique matrix). *Recall the matrix D^τ_{Cl} from Def. 6.22. Denote by D' its 0-1 valued version, i.e. D' is also diagonal and has entries*

$$D'((A,i),(A,i)) = \mathrm{Cl}_A, \quad \forall A \in \binom{[n]}{\le d/2}\ \forall 0 \le i \le \tau.$$

▶ **Lemma 6.34** (Bound on pseudo-random part of $Q^R_{0,0}$). *W.p. $> 1 - n^{-9\log n}$ the following holds: $\forall R \in \binom{[n]}{\le d/2}$,*

$$\pm\underbrace{(Q^R_{0,0} - D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{\mathrm{Cl}})}_{\text{pseudo}-\text{random part of } Q^R_{0,0}}(G) \;\preceq\; n^{-\epsilon} \cdot \operatorname{diag}\left(\widetilde{\mathrm{Cl}}(G)\right)_{S^R \times S^R} \tag{6.65}$$

Proof. Fix R. For simplicity, in this proof abbreviate:

$$Q_{\mathrm{ps}} := Q^R_{0,0} - D^\tau_{\mathrm{Cl}} \cdot \mathbb{E}[Q^R_{0,0}] \cdot D^\tau_{\mathrm{Cl}} = \left(Q_{\mathrm{ps},(i,j)}\right)_{0 \le i,j \le \tau}$$

("ps" for pseudo-random), which is a $(\tau+1) \times (\tau+1)$-block matrix.

In block (i,j), by Def. 6.15 and Prop. 6.24, $Q_{\mathrm{ps},(i,j)}$ is supported within

$$S_{i,j} \times S_{i,j} \quad \text{where} \quad S_{i,j} := \{A \mid |A| + \min\{i,j\} \geq d/2\}.$$

And for each $A \neq B$,

$$Q_{\mathrm{ps},(i,j)}(A,B) = \; Q^R_{0,0}((A,i),(B,j)) = \sum_{\substack{T_m\colon |V(T_m)\cup A\cup B|\leq\tau \\ A,B\in \mathrm{mSep}_{A,B}(T_m)}} (\frac{\omega}{n})^{|V(T_m)\cup A\cup B| - \frac{|A|+|B|}{2}} \cdot q(A,B;T_m)\cdot \chi_{T_m}; \tag{6.66}$$

and

$$Q_{\mathrm{ps},(i,j)}(A,A) = \sum_{T_m\colon 1\leq |V(T_m)\setminus A|\leq \tau-|A|} (\frac{\omega}{n})^{|V(T_m)\cup A|-|A|} \cdot q(A,A;T_m)\cdot\chi_{T_m}. \tag{6.67}$$

Here we have abbreviated $q(A,B;T_m) := Y_0\Big(|R|, |V(T_m)\cup A\cup B| + (i+j)\Big)$ ((6.23)) and have omitted the indices $|R|, i+j$ when they are fixed. Two properties we need:

$$q(A,B;T_m) \text{ depends only on } |V(T_m)\cup A\cup B| \text{ when fixing } (A,B); \tag{6.68}$$

$$\Big|q(A,B;T_m)\Big| < \tau^{5\tau} \qquad \text{(by Lemma 6.8 (4)).} \tag{6.69}$$

By (6.68), $Q_{\mathrm{ps},(i,j)}(A,B)$ always factors through $\mathrm{Cl}_{A\cup B}$ and so also through $\mathrm{Cl}_A\mathrm{Cl}_B$. In particular,

$$Q_{\mathrm{ps}} = D' \cdot Q_{\mathrm{ps}} \cdot D' \tag{6.70}$$

where D' is the 0-1 diagonal-clique matrix (Definition 6.33).

▷ Claim 6.35. W.p. $> 1 - n^{-9.5\log n}$ the following holds:

$$\forall (i,j) \qquad \pm Q_{\mathrm{ps},(i,j)} \;\prec\; n^{-1.1\epsilon}\cdot \mathrm{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R_{\min\{i,j\}}\times S^R_{\min\{i,j\}}}$$

where $S^R_a := \{A \in \binom{[n]}{\leq d/2} \mid |A| + a \geq d/2\}$.

The lemma follows from this claim and (6.70). Namely, consider a different decomposition of Q_{ps} as follows. For every $b \in [0, \frac{d}{2}]$, let

$$I_b := \{i \mid d/2 - b \leq i \leq \tau\}$$

and $Q_{\mathrm{ps};b}$ be the principal minor $W_b := \big(P^R_b \times I_b\big) \times \big(P^R_b \times I_b\big)$ of Q_{ps} (0 elsewhere), where $P^R_b = \{A \subseteq [n] \mid R \subseteq A, |A| = b\}$. Then we have

$$\{((A,i),(B,j)) \in S^R \times S^R \mid 0 \leq |A| = |B| \leq d/2\} = \bigsqcup_{b=0}^{d/2} W_b.$$

Since $Q^R_{c,0}$ is supported only on those $((A,i),(B,j)) \in S^R \times S^R$ with $|A| = |B|$ (Remark 6.16(2)), in particular for $c = 0$ we have

$$Q_{ps} = \sum_{b=0}^{d/2} Q_{\mathrm{ps};b}. \tag{6.71}$$

Each $Q_{\text{ps};b}$ is block-wise in blocks $I_b \times I_b$, each block a principal minor of $Q_{\text{ps},(i,j)}$. So by Claim 6.35 w.p. $> 1 - n^{-9.5\log n}$ any $(\pm)$ such a block $\prec n^{-1.5\epsilon} \cdot \operatorname{diag}\left(2^{\binom{b}{2}}\right)_{\binom{[n]}{b}\times\binom{[n]}{b}}$, so $\pm Q_{\text{ps};b} \prec \tau^2 \cdot n^{-1.5\epsilon}\operatorname{diag}\left(2^{\binom{b}{2}}\right)_{W_b} \prec n^{-\epsilon}\operatorname{diag}\left(2^{\binom{b}{2}}\right)_{W_b}$. Hence by (6.71) and the union bound over b, $\pm Q_{\text{ps}} \prec n^{-\epsilon}\operatorname{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R\times S^R}$ w.p. $1 - n^{-9\log n}$. Finally, insert this to the middle of (6.70), where notice $\widetilde{\text{Cl}}_A = 2^{\binom{|A|}{2}} \cdot \text{Cl}_A$, $\text{Cl}_A = \text{Cl}_A^2$, we get (6.65). ◀

Proof of Claim 6.35. We use the norm bounds from Section 4. Fix (i,j), consider consider

$$Q^{\text{diag}}_{\text{ps},(i,j)} \quad \text{and} \quad Q^{\text{off}}_{\text{ps},(i,j)} = Q_{\text{ps},(i,j)} - Q^{\text{diag}}_{\text{ps},(i,j)}.$$

Diagonal part. For $Q^{\text{diag}}_{\text{ps},(i,j)}$, by (6.67) for any (A,A) in the support (i.e. $|A| + i \geq d/2$, $|A| + j \geq d/2$),

$$Q^{\text{diag}}_{\text{ps},(i,j)}(A,A) = \widetilde{\text{Cl}}_A \cdot \underbrace{\left(\sum_{\substack{T_m:\ 1\leq|V(T_m)\backslash A|\leq\tau-|A| \\ T_m\cap E[A]=\emptyset}} (\frac{\omega}{n})^{|V(T_m)\backslash A|} q(A,A;T_m)\cdot\chi_{T_m}\right)}_{:=g(A)}.$$

For every fixed A in support, this $g(A)$ can be bounded by norms of diagonal graphical matrices, as follows. First, $q(A,A;T_m)$ depends only on $|V(T_m)\backslash A|$ (we have fixed R,i,j,A), so temporarily denote it as $q(|V(T_m)\backslash A|)$. For every $1 \leq v \leq \tau - |A|$, let $\mathcal{U}^v_1, ..., \mathcal{U}^v_{h(v)}$ be all different shapes $(A,A;T)$ (Def. 4.7) s.t. $T \cap E[A] = \emptyset$ and $|V(T)\backslash A| = v$. Clearly,

$$h(v) \leq 2^{|A|v+v^2} \quad \text{since we required } T\cap E[A]=\emptyset. \tag{6.72}$$

So w.p. $> 1 - n^{-9.6\log n}$ the following holds:

$$\begin{aligned}
|g(A)| &= \left|\sum_{v=1}^{\tau-|A|}(\frac{\omega}{n})^v q(v)\cdot\left(\sum_{x=1}^{h(v)}\underbrace{\sum_{\substack{T_m:(A,A;T_m)\text{ has} \\ \text{shape }\mathcal{U}^v_x}}\chi_{T_m}}_{=M_{\mathcal{U}^v_x}(A,A)\text{ by Def. 4.7}}\right)\right| \\
&\leq \sum_{v=1}^{\tau-|A|}(\frac{\omega}{n})^v q(v)\cdot\sum_{x=1}^{h(v)}\left\|M_{\mathcal{U}^v_x}\right\| \qquad \text{(each } M_{\mathcal{U}^v_x} \text{ is diag.)} \\
&\leq \sum_{v=1}^{\tau-|A|}(\frac{\omega}{n})^v \tau^{5\tau}\sum_{x=1}^{h(v)}\left\|M_{\mathcal{U}^v_x}\right\| \qquad \text{(by (6.69))} \\
&< \sum_{v=1}^{\tau}(\frac{\omega}{n})^v\tau^{5\tau}\cdot 2^{|A|v+v^2}\cdot n^{\frac{v}{2}}2^{O(|A|+v)} \text{ (by (6.72) and Thm. 4.8)} \\
&< \sum_{v=1}^{\tau} n^{-3\epsilon v}\cdot n^{\epsilon v} < n^{-1.2\epsilon} \qquad \text{(by the parameter regime)}
\end{aligned}$$

Off-diagonal part. Similarly, by symmetry of the coefficients (6.68), $Q^{\text{off}}_{\text{ps},(i,j)}$ is a sum of graphical matrices. I.e. let $\mathcal{U}^{s,t}_1, ..., \mathcal{U}^{s,t}_{h(s,t)}$ be the collection of distinct shapes $(A,B;T)$ s.t. $|A| = |B| = s$, $A \neq B$, $A, B \in \text{mSep}_{A,B}(T)$ and $|V(T) \cup A \cup B| = t$, then by (6.66), $Q^{\text{off}}_{\text{ps},(i,j)}$ is a block-diagonal matrix for blocks $s = d/2 - i, ..., d/2$ according to $s = |A| = |B|$, the sth block being

$$Q^{\text{off}}_{\text{ps},(i,j)}(s) = \sum_{t:\ s<t\leq\tau} (\frac{\omega}{n})^{t-s} \sum_{x=1}^{h(s,t)} q(\mathcal{U}^{s,t}_x) M_{\mathcal{U}^{s,t}_x}$$

where naturally we denote $q(A, B; T_m) = q(\mathcal{U}^{s,t}_x)$ if $(A, B; T_m)$ has shape $\mathcal{U}^{s,t}_x$. By Theorem 4.8, w.p. $> 1 - n^{-9.8\log n}$,

$$\left\|Q^{\text{off}}_{\text{ps},(i,j)}(s)\right\| \leq \sum_{s<t\leq\tau} (\frac{\omega}{n})^{t-s} \cdot h(t,s) \cdot n^{\frac{t-s}{2}} 2^{O(t)} (\log n)^{O(t-s)} \tag{6.73}$$

Also, clearly $h(t,s) \leq 2^{\binom{t}{2}+O(t)}$. Therefore, with the same high probability

$$\begin{aligned}\text{RHS of (6.73)} \leq & \sum_{\substack{d/2-\max\{i,j\}\leq s\leq d/2\\ s<t\leq\tau}} (\frac{\omega}{n})^{t-s} 2^{\binom{t}{2}+O(t)} n^{\frac{t-s}{2}} (\log n)^{O(t-s)} \\ < & \sum_{\substack{d/2-\max\{i,j\}\leq s\leq d/2\\ s<t\leq\tau}} n^{-2\epsilon(t-s)} 2^{O(t)} 2^{\binom{s}{2}} (2^{t+s}\log n)^{O(t-s)} \\ < & 2^{\binom{s}{2}} \cdot n^{-1.9\epsilon}. \qquad \text{(in our parameter regime)}\end{aligned}$$

Adding these diagonal blocks, we get that $\pm Q^{\text{off}}_{\text{ps},(i,j)} \prec n^{-1.9\epsilon} \cdot \text{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R_{\min\{i,j\}} \times S^R_{\min\{i,j\}}}$.

Finally, by the union bound we get that w.p. $> 1 - n^{-9.5\log n}$,

$$\pm Q_{\text{ps},(i,j)} = \pm(Q^{\text{diag}}_{\text{ps},(i,j)} + Q^{\text{off}}_{\text{ps},(i,j)}) \prec n^{-1.5\epsilon} \cdot \text{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R_{\min\{i,j\}} \times S^R_{\min\{i,j\}}},$$

completing the proof. ◁

▶ **Corollary 6.36** (Positiveness of $Q^R_{0,0}$). *For every $R \in \binom{[n]}{\leq d/2}$, w.p. $> 1 - n^{-8\log n}$ over G*

$$Q^R_{0,0}(G) \succeq \tau^{-6.1\tau} \cdot \text{diag}\left(\widetilde{\text{Cl}}(G)\right)_{S^R \times S^R}.$$

Proof. By Lemma 6.34 and Corollary 6.32, where $\tau^{-6.1\tau} \gg n^{-\epsilon/10}$ in our parameter regime. ◀

▶ **Lemma 6.37** (Bounds on $Q^R_{0,k}$). *W.p. $> 1-n^{-9\log n}$ the following holds. For all $R \in \binom{[n]}{\leq d/2}$ and all $1 \leq k \leq d/2$,*

$$\pm Q^R_{0,k}(G) \preceq n^{-k/10} \cdot \text{diag}\left(\widetilde{\text{Cl}}(G)\right)_{S^R \times S^R}.$$

Proof. We will use union bound over (R, k) so fix them first. **For the fixed R, $k(> 0)$, in this proof we abbreviate:**

$$Q^R_{0,k} \leftrightarrow Q.$$

Recall the definition of $Q^R_{0,k}$ (Lemma 6.19 (3)): Q is supported within $S^R \times S^R$,

$$Q\Big((A,i),(B,j)\Big) = \sum_{T_m:|V(T_m)\cup A\cup B|\leq\tau} (\frac{\omega}{n})^{t-s} q^R_{0,k}(\mathcal{R}_m, i, j) \cdot \chi_{T_m}. \tag{6.74}$$

where $t = |A \cup B|$, $s = \frac{|A|+|B|}{2}$. Abbreviate $q^R_{0,k}$ as q_k. By Lemma 6.19(3),

$$q_k(\cdot, i, j) \text{ is symmetric w.r.t. shapes for all fixed } (i, j); \tag{6.75}$$

$$|q_k(\mathcal{R}_m, i, j)| \le \tau^{5\tau} \cdot (\frac{\omega}{n^{1-\epsilon}})^{s-p+k/3} \tag{6.76}$$

where $t = |A \cup B|$, $s = \frac{|A|+|B|}{2}$, p is the maximum number of vertex-disjoint paths from A to B in $(A, B; T_m)$.

By symmetry of q_k's, $Q((A, i), (B, j))$ factors through $\mathrm{Cl}(A)\mathrm{Cl}(B)$, so

$$Q = D' \cdot Q \cdot D'. \tag{6.77}$$

where D' is by Definition 6.33. It suffices to show:

$$\text{w.p.} > 1 - n^{-9.5 \log n} \qquad \pm Q \prec n^{-k/10} \cdot \mathrm{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R \times S^R}. \tag{6.78}$$

This is because, like in the proof of Lemma 6.34, we can insert (6.78) to the middle of (6.77) which proves the lemma for the fixed R, k.

In below we prove (6.78). First, express each block of Q as a sum of graphical matrices. As a block-matrix, $Q = (Q_{(i,j)})_{0 \le i,j \le \tau}$ where $Q_{(i,j)}$ is supported on those A's s.t. $|A| + i \ge d/2$. **For any fixed (i, j)** any $(s_1, s_2) \in \{0, ..., d/2\}^2$ s.t. $s_1 + i \ge d/2$, $s_2 + j \ge d/2$, and any $t \ge \max\{s_1, s_2\}$, let $\mathcal{U}^{t;s_1,s_2}_1, ..., \mathcal{U}^{t;s_1,s_2}_{h(t;s_1,s_2)}$ be all different shapes $(A, B; T)$ where $|A| = s_1$, $|B| = s_2$, $|V(T) \cup A \cup B| = t$. Then by (6.74) and symmetry,

$$Q_{(i,j)} = \sum_{\substack{(t;s_1,s_2) \\ s_1+i, s_2+j \ge d/2 \\ \tau \ge t \ge s_1,s_2}} \sum_{x=1}^{h(t;s_1,s_2)} q_k(\mathcal{U}^{(t;s_1,s_2)}_x, i, j) \cdot M_{\mathcal{U}^{(t;s_1,s_2)}_x}.$$

This equation can be naturally viewed block-wise w.r.t. (s_1, s_2), i.e.

$$Q_{(i,j)} = \sum_{\substack{s_1,s_2 \\ s_1+i, s_2+j \ge d/2}} Q_{(s_1,i),(s_2,j)} \tag{6.79}$$

where

$$Q_{(s_1,i),(s_2,j)} := \sum_{\substack{t: \\ s_1,s_2 \le t \le \tau}} \sum_{x=1}^{h(t;s_1,s_2)} q_k(\mathcal{U}^{(t;s_1,s_2)}_x, i, j) \cdot M_{\mathcal{U}^{(t;s_1,s_2)}_x}. \tag{6.80}$$

Note that $Q_{(s_1,i),(s_2,j)}$ is a $\binom{[n]}{s_1} \times \binom{[n]}{s_2}$-matrix on the (i, j)th block of Q.

By Theorem 4.8 and (6.76), w.p. $> 1 - n^{-10 \log n}$

$$\left\|Q_{(s_1,i),(s_2,j)}\right\| \le \sum_{\substack{t:\ t \le \tau \\ t \ge s_1,s_2}} h(t; s_1, s_2) \cdot (\frac{\omega}{n})^{t-s} (\frac{\omega}{n^{1-\epsilon}})^{s-p+k/3} \cdot n^{\frac{t-p}{2}} 2^{O(t)} (\log n)^{O(t-s)} \tag{6.81}$$

where, as usual, $s = \frac{s_1+s_2}{2}$ and p is the maximum number of vertex-disjoint paths between the two distinguished subsets in the shape. Since

$$h(t; s_1, s_2) \le 2^{\binom{t}{2}+O(t)} = 2^{\binom{s}{2}+O(t)+(t+s)\cdot(t-s)},$$

we can bound the RHS of (6.81) (note $k > 0$, $2^{O(t)} < n^{\epsilon/10}$, $\tau^{5\tau} < n^{1/30}$) by

$$< 2^{\binom{s}{2}} \cdot \tau^{5\tau} n^{-k/6} n^{-\epsilon(t-s)} < 2^{\binom{s}{2}} n^{-k/8}. \tag{6.82}$$

Finally, sum over all double-blocks and use Cauchy-Schwartz. Namely, regard each $Q_{(s_1,i),(s_2,j)}$ now as on $S^R \times S^R$ (extended by 0's), then

$$Q = \sum_{\substack{(s_1,i),(s_2,j)\\ s_1+i,s_2+j\geq d/2}} Q_{(s_1,i),(s_2,j)} \tag{6.83}$$

and for each $(s_1,i),(s_2,j)$ in the summand,

$$\pm Q_{(s_1,i),(s_2,j)} \;\prec\; n^{-k/8}\cdot\left(2^{\binom{s_1}{2}}\mathrm{Id}_{(s_1,i),(s_1,i)} \;+\; 2^{\binom{s_2}{2}}\mathrm{Id}_{(s_2,j),(s_2,j)}\right)/2$$

by (6.82) and Cauchy-Schwartz. So by (6.83), w.p. $> 1-n^{-9.5\log n}$,

$$\pm Q \;\prec\; \tau^2 n^{-k/8}\mathrm{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R\times S^R} \;\prec\; n^{-k/10}\mathrm{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R\times S^R}.$$

(6.78) is proved. ◀

▶ **Lemma 6.38** (Bounds on $Q^R_{c,k}$, $c>0$). *W.p. $> 1-n^{-9\log n}$ the following holds: $\forall(R,c,k)$ where $R\in\binom{[n]}{\leq d/2}$, $0<c\leq |R|$ and $0\leq k\leq d/2$,*

$$\pm\omega^{-c}\cdot Q^R_{c,k} \;\preceq\; n^{-c/3}\cdot\mathrm{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R\times S^R}. \tag{6.84}$$

Proof. The proof is almost the same as the previous one (Lemma 6.37). First, by a union bound over all such (R,c,k), it suffices to show that w.p. $> 1-n^{-9.5\log n}$ the inequality holds for a fixed (R,c,k); we do it below.

Fix (R,c,k) as in the condition. If $k>0$ then the proof is identical to that of Lemma 6.37 ($c=0$), since the same coefficient-size condition and symmetry condition (6.75), (6.76) hold here by Lemma 6.19, and moreover, the matrix $Q^R_{c,k}$ is supported within $S^R\times S^R$ too.

So we only need to deal with the case $c>0$, $k=0$, i.e. $Q^R_{c,0}$. By Definition 6.15, the matrix is supported on $S^R\times S^R$ with expression $Q^R_{c,0}\Big((A,i),(B,j)\Big) =$

$$\sum_{\substack{T_m:|V(T_m)\cup A\cup B|\leq\tau\\ A,B\in\mathrm{mSep}_{A,B}(T_m)}} (\frac{\omega}{n})^{|V(T_m)\cup A\cup B|-\frac{|A|+|B|}{2}}\cdot Y_c\big(|R|,\ |V(T_m)\cup A\cup B|+(i+j)\big)\cdot\chi_{T_m} \tag{6.85}$$

where $\left|Y_c\big(|R|,\ |V(T_m)\cup A\cup B|+(i+j)\big)\right| < \tau^{5\tau}$ by Lemma 6.8 (4). If for every fixed $(A,B;T_m)$ denote $t=|V(T_m)\cup A\cup B|$, $s=\frac{|A|+|B|}{2}(=|A|=|B|$ in this case), then the coefficient in (6.85) is bounded by $(\frac{\omega}{n})^{t-s}\cdot\tau^{5\tau}$. Therefore, we have the support condition, the symmetry, and the size condition on the coefficients as in Lemma 6.37, so we can proceed exactly the same as there till equation (6.81), where a single term in its RHS now becomes

$$h(t;s_1,s_2)\cdot(\frac{\omega}{n})^{t-s}\tau^{5\tau}\cdot n^{\frac{t-p}{2}}2^{O(t)}(\log n)^{O(t-s)}.$$

Note in (6.85) any appearing ribbon $\mathcal{R}_m=(A,B;T_m)$ satisfies $A,B\in\mathrm{mSep}_{A,B}(T_m)$ so $p=s$ (the specialty of the case $k=0$). So we can replace the bound on the RHS of (6.82) by $\tau^3 2^{\binom{s}{2}}\cdot n^{-3\epsilon(t-s)}\tau^{5\tau}2^{O(t)} < 2^{\binom{s}{2}}\tau^{6\tau}$, and then proceed to the last line of the proof there, with the bound now being

$$\pm Q^R_{c,0} \;\prec\; \tau^{7\tau}\cdot\mathrm{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R\times S^R}.$$

In particular, since $c \geq 1$, $\omega = n^{\frac{1}{2}-4\epsilon}$ (assuming $\epsilon < 1/40$) and $\tau^{7\tau} < n^{1/15}$, we get $\pm\omega^{-c} \cdot Q^R_{c,0} \prec n^{-c/3} \cdot \text{diag}\left(2^{\binom{|A|}{2}}\right)_{S^R \times S^R}$ by our parameters. Once again like before, using $Q^R_{c,0} = D' \cdot Q^R_{c,0} \cdot D'$ we get that $\pm\omega^{-c} \cdot Q^R_{c,0} \preceq n^{-c/3} \cdot \text{diag}\left(\widetilde{\text{Cl}}\right)_{S^R \times S^R}$. ◀

Lemma 6.21 follows immediately from Corollary 6.36, Lemma 6.37, 6.38.

6.7 Last step

Now we prove the Main Lemma 6.9, hence Theorem 6.1. For any fixed R, recall the notation $P^R = \{I \in \binom{[n]}{d/2} \mid R \subseteq I\}$.

Lemma 6.9 recast. W.p. $1 - n^{-5\log n}$ it holds that for all $R \subseteq \binom{[n]}{d/2}$:

$$M^R_0 \succeq n^{-d} \cdot \text{diag}(\widetilde{\text{Cl}})_{P^R \times P^R}; \tag{6.86}$$

$$\pm\,\omega^{-c} M^R_c \preceq n^{-c/6} \cdot M^R_0, \quad \forall 0 < c \leq |R|. \tag{6.87}$$

Further recall that $D^\tau = \text{diag}\left((\frac{\omega}{n})^{\frac{|A|}{2}}\right)_{A:|A|\leq\frac{d}{2}} \otimes \text{Id}_{\{0,...,\tau\}\times\{0,...,\tau\}}$ (Def. 6.12), and that $S^R = \{(A,i) \in \binom{[n]}{\leq d/2} \times \{0,...,\tau\} \mid A \supseteq R, |A| + i \geq \frac{d}{2}\}$. The following lemma will be handy.

▶ **Lemma 6.39.** $\forall R \in \binom{[n]}{\leq d/2}$,

$$\widetilde{L^R} D^\tau \cdot \text{diag}\left(\widetilde{\text{Cl}}\right)_{S^R \times S^R} \cdot D^\tau (\widetilde{L^R})^\top \succeq (\frac{\omega}{n})^{d/2} \text{diag}\left(\widetilde{\text{Cl}}\right)_{P^R \times P^R}$$

when evaluated on any G.

Proof. Fix any $R \in \binom{[n]}{\leq d/2}$. Without confusion, we omit subscript $S^R \times S^R$ by regarding the supports as the vertex-set $[n'] = [n] - R$ and regarding the corresponding matrix indices as $\binom{[n']}{d'/2}$ or $\binom{[n']}{\leq d'/2}$, where $d'/2 = d/2 - |R|$. τ is unchanged. We will still use $\widetilde{\text{Cl}}(X)$ to mean $\widetilde{\text{Cl}}(X \sqcup R)$ for $X \subseteq [n']$.

Since $D^\tau \text{diag}(\widetilde{\text{Cl}}) D^\tau$ is nonnegative and diagonal for any G, we have

$$\widetilde{L^R}\left(D^\tau \cdot \text{diag}\left(\widetilde{\text{Cl}}\right) \cdot D^\tau\right)(\widetilde{L^R})^\top \succeq L^{R,0}\left(D^\tau \cdot \text{diag}\left(\widetilde{\text{Cl}}\right) \cdot D^\tau\right)(L^{R,0})^\top, \tag{6.88}$$

where recall $\widetilde{L^R} = (L^{R,0}, ..., L^{R,\tau})$. Further, $L^{R,0} = (L^{R,0}_0, ..., L^{R,0}_{d'/2})$, where $L^{R,0}_t$ is the matrix on column set $\binom{n'}{t}$. In particular,

$$L^{R,0}_{d/2-|R|} = \left(0, ..., 0, \text{diag}\left(\widetilde{\text{Cl}}\right)_{\binom{[n']}{d'/2} \times \binom{[n']}{d'/2}}\right)$$

since in the definition of $L^{R,0}$ (Def. 6.11) only ribbons $\mathcal{R} = (I, A; T')$ with 0-reduced size can occur, and with the other conditions on it this simply means that $A = I$ and $T' \subseteq E(I)$. This implies

$$\text{RHS of (6.88)} \succ (\frac{\omega}{n})^{d/2} \cdot \text{diag}\left(\widetilde{\text{Cl}}\right)_{\binom{[n']}{d'/2} \times \binom{[n']}{d'/2}}.$$

Translated back to $[n]$ and $d/2$, this is exactly the bound in the lemma. ◀

Proof for Lemma 6.9. Fix $R \in \binom{[n]}{\leq d/2}$. By Lemma 6.19, for all $c \leq |R|$

$$M_c^R = \widetilde{L^R} \cdot \left[D^\tau \left(Q_{c,0}^R - Q_{c,1}^R + ... \pm Q_{c,d}^R\right) D^\tau\right] \cdot \left(\widetilde{L^R}\right)^\top + \mathcal{E}_c^R. \tag{6.89}$$

The following bounds all hold w.p. $> 1 - n^{-8\log n}$ from the corresponding lemmas, and we take union bound so the overall probability is $> 1 - n^{-5\log n}$.

For (6.86). Fix R, we have:

$$\begin{aligned}
M_0^R &= \widetilde{L^R} \cdot \left[D^\tau \left(Q_{0,0}^R - Q_{0,1}^R + ... \pm Q_{0,d}^R\right) D^\tau\right] \cdot \left(\widetilde{L^R}\right)^\top + \mathcal{E}_0^R \\
&\succeq \tau^{-7\tau} \left[\widetilde{L^R} \cdot D^\tau \operatorname{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R \times S^R} D^\tau \cdot \left(\widetilde{L^R}\right)^\top\right] + \mathcal{E}_0^R \quad \text{(Lem. 6.21(1))} \\
&\succeq \tau^{-7\tau} (\frac{\omega}{n})^{d/2} \cdot \operatorname{diag}\left(\widetilde{\mathrm{Cl}}\right)_{P^R \times P^R} + \mathcal{E}_0^R \quad \text{(Lemma 6.39)} \\
&\succeq (\tau^{-7\tau} (\frac{\omega}{n})^{d/2} - n^{-\epsilon\tau/2}) \cdot \operatorname{diag}\left(\widetilde{\mathrm{Cl}}\right)_{P^R \times P^R} \quad \text{(Lemma 6.19(4))} \\
&\succeq n^{-d} \cdot \operatorname{diag}(\widetilde{\mathrm{Cl}})_{P^R \times P^R} \quad \text{(parameter regime)}
\end{aligned}$$

For (6.87). Fix R, $1 \leq c \leq |R|$, we have:

$$\begin{aligned}
M_c^R &= \widetilde{L^R} \cdot \left[D^\tau \left(Q_{c,0}^R - Q_{c,1}^R + ... \pm Q_{c,d}^R\right) D^\tau\right] \cdot \left(\widetilde{L^R}\right)^\top + \mathcal{E}_c^R \\
&\preceq \omega^c n^{-c/4} \left[\widetilde{L^R} D^\tau \cdot \operatorname{diag}\left(\widetilde{\mathrm{Cl}}\right)_{S^R \times S^R} \cdot D^\tau \left(\widetilde{L^R}\right)^\top\right] + \mathcal{E}_c^R \quad \text{(Lem. 6.21(2))} \\
&\preceq \omega^c n^{-c/4} \left[\tau^{7\tau} (M_0^R - \mathcal{E}_0^R)\right] + \mathcal{E}_c^R \quad \text{(Lem. 6.21(1) and (6.89))} \\
&\preceq \omega^c n^{-c/5} M_0^R + \left(\omega^c n^{-c/5} + 1\right) n^{-\epsilon\tau/2} \operatorname{diag}(\mathrm{Cl})_{P^R \times P^R} \quad \text{(Lem. 6.19(4))}
\end{aligned}$$

So

$$\begin{aligned}
\omega^{-c} M_c^R &\preceq n^{-c/5} M_0^R + 2n^{-\epsilon\tau/2} \cdot \operatorname{diag}(\mathrm{Cl})_{P^R \times P^R} \\
&\preceq (n^{-c/5} + 2n^d n^{-\epsilon\tau/2}) M_0^R \quad \text{((6.86) and } \widetilde{\mathrm{Cl}} \geq \mathrm{Cl}) \\
&\preceq n^{-c/6} \cdot M_0^R \quad (c \leq |R| \leq d/2 \text{ and parameter regime})
\end{aligned}$$

The same analysis holds for $-\omega^{-c} M_c^R$. ◀

7 Concluding remarks

We established the average $\Omega(\epsilon^2 \log n / \log\log n)$ SOS degree lower bound for Exact Clique with clique-size $\omega = n^{1/2-\epsilon}$, which is nearly optimal in both parameters ω, d. We also refreshed the techniques for the Non-Exact Clique problem in hope to make them simpler and generalizable. Some open problems follow.

(1) Can we remove the $\log\log n$ factor in d? Perhaps it helps to first find a conceptual explanation of Definition 3.11.

(2) How about the same problem on $G(n, p)$, $p \neq \frac{1}{2}$ and for suitable ω? For Non-Exact Clique, we can define the pseudo-expectation similarly as in Section 3.1.2. Also, using the Fourier orthonormal basis

$$\chi_T = \prod_{e \in T} \frac{x_e - (2p-1)}{2\sqrt{p(1-p)}} \qquad \forall T \subseteq E[n], \tag{7.1}$$

where $x_e(G)$ is the ± 1-indicator of edge e, we have the corresponding version of norm bounds in Section 4 since the trace-power method works the same. The questions is, what is the best meaningful degree lower bound for varying p (especially small p)? How about the exact case?

(3) What can be said when G is drawn from other random models, or is pseudo-random?

References

1 Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures & Algorithms*, 13(3-4):457–466, 1998.

2 Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180, 2010.

3 Sanjeev Arora, Boaz Barak, Markus Brunnermeier, and Rong Ge. Computational complexity and information asymmetry in financial products. *Communications of the ACM*, 54(5):101–107, 2011.

4 Boaz Barak, Fernando GSL Brandao, Aram W Harrow, Jonathan Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 307–326, 2012.

5 Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing*, 48(2):687–735, 2019.

6 Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. In *Proceedings of International Congress of Mathematicians (ICM)*, 2014.

7 Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on hilbert's nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 3(1):1–26, 1996.

8 Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *Conference on Learning Theory*, pages 1046–1066. PMLR, 2013.

9 P Delsarte. An algebraic approach to association schemes of coding theory, phillips j, 1973.

10 Yash Deshpande and Andrea Montanari. Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems. In *Conference on Learning Theory*, pages 523–562. PMLR, 2015.

11 Fernando Escalante. Schnittverbände in graphen. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 38, pages 199–220. Springer, 1972.

12 Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures & Algorithms*, 16(2):195–208, 2000.

13 Uriel Feige and Robert Krauthgamer. The probable value of the lovász–schrijver relaxations for maximum independent set. *SIAM Journal on Computing*, 32(2):345–370, 2003.

14 Dima Grigoriev and Nicolai Vorobjov. Complexity of null-and positivstellensatz proofs. *Annals of Pure and Applied Logic*, 113(1-3):153–160, 2001.

15 Samuel B Hopkins, Pravesh Kothari, Aaron Henry Potechin, Prasad Raghavendra, and Tselil Schramm. On the integrality gap of degree-4 sum of squares for planted clique. *ACM Transactions on Algorithms (TALG)*, 14(3):1–31, 2018.

16 Samuel B Hopkins, Pravesh K Kothari, and Aaron Potechin. Sos and planted clique: Tight analysis of mpw moments at all degrees and an optimal lower bound at degree four. *arXiv preprint*, 2015. `arXiv:1507.05230`.

17 Samuel B Hopkins, Pravesh K Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The power of sum-of-squares for detecting hidden structures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 720–731. IEEE, 2017.

18 Mark Jerrum. Large cliques elude the metropolis process. *Random Structures & Algorithms*, 3(4):347–359, 1992.

19 R Karp. Probabilistic analysis of some combinatorial search problems. traub, jf (ed.): Algorithms and complexity: New directions and recent results, 1976.

20 Pravesh Kothari, Ryan O'Donnell, and Tselil Schramm. Sos lower bounds with hard constraints: think global, act local. *arXiv preprint*, 2018. `arXiv:1809.01207`.

21 Pravesh K Kothari and Ruta Mehta. Sum-of-squares meets nash: Optimal lower bounds for finding any equilibrium. *arXiv preprint*, 2018. `arXiv:1806.09426`.

22 Luděk Kučera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212, 1995.

23 Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

24 Dhruv Medarametla and Aaron Potechin. Bounds on the norms of uniform low degree graph matrices. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

25 Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 87–96, 2015.

26 Ryan O'Donnell. Sos is not obviously automatizable, even approximately. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

27 Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

28 Pavel A Pevzner, Sing-Hoi Sze, et al. Combinatorial approaches to finding subtle signals in dna sequences. In *ISMB*, volume 8, pages 269–278, 2000.

29 Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. *arXiv preprint*, 2017. `arXiv:1702.05139`.

30 Naum Z Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

31 Evgenij E Tyrtyshnikov. How bad are hankel matrices? *Numerische Mathematik*, 67(2):261–269, 1994.

A Deductions in mod-order analysis (Section 5.2)

A.1 Set-up recap

Ring $\mathbb{A}$ is got by adding fresh variables α and χ_T's to $\mathbb{R}$, where T ranges over edge sets on $[n]$, and they only satisfy the relations $\{\chi_{T'} \cdot \chi_{T''} = \chi_T$ whenever $T' \oplus T'' = T\}$. The **mod-order equation** is

$$L_\alpha \cdot \operatorname{diag}\left(\alpha^{|A|}\right) \cdot (L_\alpha)^\top = M_\alpha \qquad \bmod (*) \tag{A.1}$$

on the $\binom{[n]}{d/2} \times \binom{[n]}{\leq d/2}$-matrix variable L_α in ring $\mathbb{A}$, where

$$M_\alpha(I,J) = \sum_{T:|V(T)\cup I\cup J|\leq\tau} \alpha^{|V(T)\cup I\cup J|}\chi_T \quad \forall I,J : |I| = |J| = d/2,$$

and mod $(*)$ means to mod the ideal $(\{\alpha^{|V(T)\cup I\cup J|+1}\chi_T\}, \{\chi_T : |V(T)\cup I\cup J| > \tau\})$ position-wise on each (I,J). We call $(*)$ the **modularity**. Moreover, if denote

$$L_1'(I,A) = \sum_{T'} \beta_{I,A}(T')\chi_{T'}, \quad \beta_{I,A}(T') \in \mathbb{R}[\alpha]$$

then we require

$$\alpha^{e_{I,A}(T')} \mid \beta_{I,A}(T') \quad \forall I, A, T' \tag{A.2}$$

where $e_{I,A}(T')$ is the reduced size $|V(T') \cup I \cup A| - s_{I,A}(T')$ (Def. 4.11).

Expressed in terms, equations (A.1), (A.2) become the following.

$$\sum_{A \in \binom{[n]}{\leq d/2}} \sum_{\substack{T', T'': \\ T' \oplus T'' = T}} \alpha^{|A|} \cdot \beta_{I,A}(T') \cdot \beta_{J,A}(T'') = \alpha^{|V(T) \cup I \cup J|} \mod \alpha^{|V(T) \cup I \cup J|+1} \tag{A.3}$$

for every $(I, J; T)$ with $|V(T) \cup I \cup J| \leq \tau$, and

$$\alpha^{e_{I,A}(T')} \mid \beta_{I,A}(T') \tag{A.4}$$

for every $(I, A; T')$.

The main observation (Lemma 5.6) is the following.

▶ **Lemma A.1** (Order match). *In the LHS of equation* (A.3)*, only products* $\alpha^{|A|} \cdot \beta_{I,A}(T') \cdot \beta_{J,A}(T'')$ *that satisfies the following are non-zero modulo* $(*)$*.*

$$A \text{ is a min-separator for both } (I, A; T'),\ (J, A; T''); \tag{A.5}$$

$$(V(T') \cup I \cup A) \ \cap \ (V(T'') \cup J \cup A) = A. \tag{A.6}$$

Moreover, (A.5)*,* (A.6) *imply that*

$$A \text{ is a min-separator of } (I, J; T) \text{ (where } T = T' \oplus T''\text{)}; \tag{A.7}$$

$$|V(T') \cup I \cup A|,\ |V(T'') \cup J \cup A| \leq \tau. \tag{A.8}$$

Proof. Pick a term $\alpha^{|A|} \cdot \beta_{I,A}(T') \cdot \beta_{J,A}(T'')$ form the LHS of (A.3). By (A.4),

$$\text{its order in } \alpha \geq |A| + |V(T') \cup I \cup A| - s_{I,A}(T') + |V(T'') \cup A \cup J| - s_{J,A}(T'').$$

By modularity on the RHS of (A.3), the term is non-zero only if

$$\text{its order in } \alpha \leq |V(T) \cup I \cup J| \quad \text{and} \quad |V(T) \cup I \cup J| \leq \tau$$

where $T = T' \oplus T''$. This implies

$$|V(T') \cup I \cup A| + |V(T'') \cup J \cup A| \leq \underbrace{|V(T) \cup I \cup J|}_{\textcircled{1}} + \underbrace{(s_{I,A}(T') + s_{J,A}(T'') - |A|)}_{\textcircled{2}} \tag{A.9}$$

Note $\textcircled{2} \leq |A|$ and "=" holds iff $s_{I,A}(T') = s_{J,A}(T'') = |A|$. While the LHS above

$$= \underbrace{|(V(T') \cup I \cup A) \cup (V(T'') \cup J \cup A)|}_{\geq |V(T) \cup I \cup J| = \textcircled{1}} + \underbrace{|(V(T') \cup I \cup A) \cap (V(T'') \cup J \cup A)|}_{\geq |A| \geq \textcircled{2}}.$$

Therefore, (A.9) could hold only when all "="'s hold, which means: (1). A is a min-separator of $(I, A; T')$, $(J, A; T'')$; (2). $(V(T') \cup I \cup A) \cup (V(T'') \cup J \cup A) = V(T) \cup I \cup J$; (3). $(V(T') \cup I \cup A) \cap (V(T'') \cup J \cup A) = A$.

Next, we show (1),(3) imply $A \in \mathrm{mSep}_{I,J}(T)$ (and also (2), actually). By (3), T', T'' could overlap only in $E(A)$. Now $T = T' \oplus T''$, so

$$T = T' \sqcup T'' \quad \text{modulo } E(A) \tag{A.10}$$

(also $\Rightarrow$ $V(T') \cup V(T'') \subseteq V(T) \cup A$). By (1) there are $|A|$ many vertex-disjoint paths $p_1, , , .p_{|A|}$ from I to A in T', and similarly $q_1, ..., q_{|A|}$ from J to A in T''. These paths are also present in T by (A.10) – where it naturally assumes every path touches A only once at its endpoint. By (3) again, any p_i, q_j do not intersect beside endpoint in A so they are paired to $|A|$ many vertex-disjoint paths from I to J in T, all passing A (this also implies $A \subseteq V(T) \cup I \cup J$). On the other hand, if p is a path in T from I not passing A, then it is a path on $I \cup V(T')$ by induction using (3). Now by (3) again we have $(V(T') \cup I) \cap J \subseteq A$, so p can't reach J. So $A \in \mathrm{mSep}_{I,J}(T)$.

Finally, under the above implications, $V(T') \cup I \cup A \subseteq V(T) \cup I \cup J$ and similarly for $V(T'') \cup J \cup A$, so both have size $\leq \tau$. ◀

By this lemma, we can assume that in an imagined solution, $\beta_{I,A}(T') \neq 0$ only when it satisfies the conditions (A.5), (A.8) on its part. If assume further that the solution is *symmetric* (which looks plausible), i.e. $\beta_{I,A}(T') = \beta_{J,B}(T'')$ whenever $(I, A; T')$, $(J, B; T'')$ are of the same shape, then this lemma is particularly informative about some special $(I, J; T)$'s.

▶ **Corollary A.2.** *If $(I, J; T)$ has a **unique** min-separator A, then*

$$\sum_{\substack{T', T'' :\ T' \oplus T'' = T \\ \text{(A.5), (A.6) hold}}} \beta_{I,A}(T') \cdot \beta_{J,A}(T'') = \alpha^{e_{I,J}(T)} \tag{A.11}$$

where $e_{I,J}(T) = |V(T) \cup I \cup J| - s_{I,J}(T)$. In particular, in symmetric solution,

$$\sum_{T_1 \subseteq E(A)} \beta_{I,A}(T_1 \oplus T')^2 = \alpha^{2 \cdot e_{I,A}(T')} \tag{A.12}$$

for all $(I, A; T')$ such that

$$A \text{ is the unique min-separator of } (I, A; T'). \tag{A.13}$$

Proof. The first part is directly from Lemma 5.6. For the "in particular" part, let $(I, A; T')$ satisfy (A.13). By mirroring $(I, A; T')$ through A, we get a $(J, A; T'')$ that satisfies the same condition and they together satisfy (A.5), (A.6). There are always enough vertices in $[n]$ to carry out this mirroring operation. By the symmetry assumption, $\beta_{I,A}(T') = \beta_{J,A}(T'')$. From mirroring it is not hard to see that A is the unique min-separator of $(I, J; T = T' \oplus T'')$, so for this triple $(I, J; T)$ equation (A.11) holds, giving that $\sum_{T_1 \subseteq E(A)} \beta_{I,A}(T' \oplus T_1)^2 = \alpha^{|V(T) \cup I \cup J| - |A|} = \alpha^{2(|V(T') \cup I \cup A| - |A|)}$. ◀

We can summarize what we got as follows. If let all $\beta_{I,A}(T' \oplus T_1)$'s in equation (A.12) be equal (which is a plausible assumption), then $\beta_{I,A}(T') = 2^{-\binom{|A|}{2}/2} \cdot \alpha^{e_{I,A}(T')}$ (take all + signs). Collecting these terms, we get the following matrix

$$L_1' : \quad L_1'(I, A) = \sum_{\substack{T' :\ |V(T') \cup I \cup A| \leq \tau \\ \text{(A.13) holds} \\ T' \cap E(A) = \emptyset}} 2^{-\binom{|A|}{2}/2} \cdot \alpha^{|V(T') \cup I \cup A| - |A|} \chi_{T'} \cdot \widetilde{\mathrm{Cl}}_A$$

where $\widetilde{\mathrm{Cl}}_A = \sum_{T \subseteq E(A)} \chi_T$. To see how far this is from a solution, notice $\widetilde{\mathrm{Cl}}_A^2 = 2^{\binom{|A|}{2}} \widetilde{\mathrm{Cl}}_A$ and consider

$$L_1' \cdot \mathrm{diag}\left(\alpha^{|A|}\right) \cdot (L_1')^\top = L_1 \cdot \mathrm{diag}\left(\alpha^{|A|} \cdot \widetilde{\mathrm{Cl}}_A\right) \cdot L_1^\top \tag{A.14}$$

where L_1 is the matrix in $\mathbb{A}$ as below (which is cleaner than L_1' to use).

▶ **Definition A.3.** $\forall I \in \binom{[n]}{d/2}$, $A \in \binom{[n]}{\le d/2}$,

$$L_1(I,A) := \sum_{\substack{T':\ |V(T')\cup I\cup A|\le\tau \\ \text{(A.13) holds} \\ T'\cap E(A)=\emptyset}} \alpha^{|V(T')\cup I\cup A|-|A|}\chi_{T'}. \tag{A.15}$$

Surely L_1' is not a solution to the mod-order equation, since (A.14) equals (mod (*)) only the part of M_α consisting of the special $(I,J;T)$'s from Corollary A.2. For a general $(I,J;T)$, Lemma A.1 only says:

$$\sum_{\substack{A,T',T'':\ T'\oplus T''=T \\ A\in \mathrm{mSep}_{I,J}(T) \\ \text{(A.5),(A.6) hold}}} \beta_{I,A}(T')\beta_{J,A}(T'') = \alpha^{e_{I,J}(T)} \mod \alpha^{e_{I,J}(T)+1}. \tag{A.16}$$

To see how to proceed further, we inspect a further weakening: polarization.

A.2 Polarized solution

Roughly speaking, polarization weakens linear equations about "x_i^2's" by replacing these terms with multi-linear "x_iy_i's", where $\vec{y}$ are fresh variables. Then we can plug in any "tentative" solution $\vec{x_0}$ to solve for $\vec{y}$ more easily (as the equations are linear in $\vec{y}$), and see how to modify $\vec{x_0}$ further.

▶ **Definition A.4.** *The* ***polarized*** *mod-order equation w.r.t.* L_1 *is:*

$$L_1\cdot \mathrm{diag}\left(\alpha^{|A|}\cdot \widetilde{\mathrm{Cl}}_A\right)\cdot L_2^\top = M_\alpha \qquad \mod (*) \tag{A.17}$$

where $(*)$ *is the modularity in* (A.1)*,* L_1 *is by* (A.15)*,* L_2 *is the variable matrix*

$$L_2(I,A) = \sum_{T':\ |V(T')\cup I\cup A|\le\tau} \beta^{(2)}_{I,A}(T')\chi_{T'} \tag{A.18}$$

satisfying $\alpha^{e_{I,A}(T')} \mid \beta^{(2)}_{I,A}(T')$ *for all* (I,A,T')*.*

In this polarized form, the essential condition (A.16) becomes

$$\sum_{\substack{A,T',T'':\ T'\oplus T''=T \\ (I,A;T')\text{ appears in } L_1 \\ \text{(A.5),(A.6) hold}}} \alpha^{e_{I,A}(T')}\cdot \beta^{(2)}_{J,A}(T'') = \alpha^{e_{I,J}(T)} \mod \alpha^{e_{I,J}(T)+1}. \tag{A.19}$$

By (A.19), existence of a solution L_2 at least requires the following condition: for general $(I,J;T)$, there always exist "$(I,A;T')$ appearing in L_1" and T'' which satisfy the condition in the LHS of (A.19). By a direct (but careful) check, this condition is actually **equivalent to** an essential part of the following graph-theoretic fact due to Escalante (its "In particular" part).

▶ **Fact A.1** ([11]; also Appendix A.3 of [5]). *For any ribbon* $(I,J;T)$*, the set of all min-separators,* $\mathrm{mSep}_{I,J}(T)$*, has a natural poset structure: min-separators* $A_1 \le A_2$ *iff* A_1 *separates* $(I,A_2;T)$*, or equivalently as can be checked, iff* A_2 *separates* $(J,A_1;T)$*. The set is further a* ***lattice*** *under this partial-ordering:* $\forall A_1,A_2\in \mathrm{mSep}_{I,J}(T)$ *their join and meet exist. In particular, there exist a unique* ***minimum*** *and* ***maximum****.*

Denote the minimum by $S_l(I,J;T)$ *and the maximum by* $S_r(I,J;T)$*, which is the "leftmost" and "rightmost" min-separator, respectively.*

By this fact, some $(I, A; T')$ indeed appears in (A.19) with $A = S_l(I, J; T)$. Moreover, (A.19) is naturally satisfied if take

$$L_2(J, A) = \sum_{\substack{T'':\ |V(T'')\cup J\cup A|\le\tau \\ A\in \mathrm{mSep}_{J,A}(T'') \\ T''\cap E(A)=\emptyset \\ (J,A;T'')\text{ left-generated}}} \alpha^{e_{J,A}(T'')}\chi_{T''}. \tag{A.20}$$

Here, recall being left-generated means every vertex is either in A or can be connected from J without touching A. Also, with this L_2 only one product in the LHS of (A.19) contributes to the right modulo $\alpha^{e_{I,J}(T)+1}$. We get:

▶ **Proposition A.5.** *The pair* (L_1, L_2) *is a solution to the polarized mod-order equation* (A.17)*,* (A.18)*.*

Remove the polarization. One more use of fact A.1 actually shows that, if move the "left-generated" condition from L_2 to L_1, then L_2 itself *effectively* factors through L_1, i.e. we can replace $\mathrm{diag}(\widetilde{\mathrm{Cl}}) \cdot L_2^\top$ by some $X \cdot L_1^\top$ in (A.17). This is the idea behind the following proposition (Prop. 5.8 recast).

▶ **Proposition A.6** (Mod-order diagonalization)**.** *Let*

$$L_\alpha(I, A) := \sum_{\substack{T':\ |V(T')\cup I\cup A|\le\tau \\ A=S_l(I,A;T') \\ T'\cap E(A)=\emptyset \\ (I,A;T')\text{ left-generated}}} \alpha^{e_{I,A}(T')}\chi_{T'},$$

$$Q_{0,\alpha}(A, B) := \sum_{\substack{T_m:\ |T\cup A\cup B|\le\tau \\ A,B\in \mathrm{mSep}_{A,B}(T_m)}} \alpha^{e_{A,B}(T_m)}\chi_{T_m}$$

(where T_m *indicates "middle"). Then*

$$L_\alpha \cdot [\mathrm{diag}\left(\alpha^{\frac{|A|}{2}}\right) \cdot Q_{0,\alpha} \cdot \mathrm{diag}\left(\alpha^{\frac{|A|}{2}}\right)] \cdot L_\alpha^\top = M_\alpha \qquad \mathrm{mod}\ (*) \tag{A.21}$$

where $(*)$ *is the modularity in* (A.1)*.*

Proof. Given Fact A.1, we immediately have the *canonical decomposition* of graphs as in Definition 5.11 and Remark 5.12. This implies that in the LHS of (A.21) only the products from canonical triples are non-zero modulo $(*)$, and they give M_α. ◀

Thus we get a "$L_1(-)L_1^\top$"-shape decomposition, meaning that we do not lose much from the polarization step if recall the goal is only about PSDness.

A Direct Product Theorem for One-Way Quantum Communication

Rahul Jain
Centre for Quantum Technologies & Department of Computer Science,
National University of Singapore, Singapore
Majulab, UMI 3654, Singapore

Srijita Kundu
Centre for Quantum Technologies, National University of Singapore, Singapore

Abstract

We prove a direct product theorem for the one-way entanglement-assisted quantum communication complexity of a general relation $f \subseteq \mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$. For any $0<\varepsilon<\delta<\frac{1}{2}$ and any $k\geq 1$, we show that

$$\mathrm{Q}^1_{1-(1-\varepsilon)^{\Omega(k/\log|\mathcal{Z}|)}}(f^k) = \Omega\left(k\cdot \mathrm{Q}^1_\delta(f)\right),$$

where $\mathrm{Q}^1_\varepsilon(f)$ represents the one-way entanglement-assisted quantum communication complexity of f with worst-case error ε and f^k denotes k parallel instances of f.

As far as we are aware, this is the first direct product theorem for the quantum communication complexity of a general relation – direct sum theorems were previously known for one-way quantum protocols for general relations, while direct product theorems were only known for special cases. Our techniques are inspired by the parallel repetition theorems for the entangled value of two-player non-local games, under product distributions due to Jain, Pereszlényi and Yao [24], and under anchored distributions due to Bavarian, Vidick and Yuen [4], as well as message compression for quantum protocols due to Jain, Radhakrishnan and Sen [29]. In particular, we show that a direct product theorem holds for the distributional one-way quantum communication complexity of f under any distribution q on $\mathcal{X}\times\mathcal{Y}$ that is anchored on one side, i.e., there exists a y^* such that $q(y^*)$ is constant and $q(x|y^*)=q(x)$ for all x. This allows us to show a direct product theorem for general distributions, since for any relation f and any distribution p on its inputs, we can define a modified relation $\tilde{f}$ which has an anchored distribution q close to p, such that a protocol that fails with probability at most ε for $\tilde{f}$ under q can be used to give a protocol that fails with probability at most $\varepsilon+\zeta$ for f under p.

Our techniques also work for entangled non-local games which have input distributions anchored on any one side, i.e., either there exists a y^* as previously specified, or there exists an x^* such that $q(x^*)$ is constant and $q(y|x^*)=q(y)$ for all y. In particular, we show that for any game $G=(q,\mathcal{X}\times\mathcal{Y},\mathcal{A}\times\mathcal{B},\mathsf{V})$ where q is a distribution on $\mathcal{X}\times\mathcal{Y}$ anchored on any one side with constant anchoring probability, then

$$\omega^*(G^k) = \left(1-(1-\omega^*(G))^5\right)^{\Omega\left(\frac{k}{\log(|\mathcal{A}|\cdot|\mathcal{B}|)}\right)}$$

where $\omega^*(G)$ represents the entangled value of the game G. This is a generalization of the result of [4], who proved a parallel repetition theorem for games anchored on both sides, i.e., where both a special x^* and a special y^* exist, and potentially a simplification of their proof.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Quantum complexity theory

Keywords and phrases Direct product theorem, parallel repetition theorem, quantum communication, one-way protocols, communication complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.27

Related Version *Full Version*: `https://arxiv.org/abs/2008.08963`

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 27; pp. 27:1–27:28
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding This work is supported by the National Research Foundation, including under NRF RF Award No. NRF-NRFF2013-13, the Prime Minister's Office, Singapore and the Ministry of Education, Singapore, under the Research Centres of Excellence program and by Grant No. MOE2012-T3-1-009 and in part by the NRF2017-NRF-ANR004 *VanQuTe* Grant.

1 Introduction

A fundamental question in complexity theory is: given k independent instances of a function or relation, does computing them require k times the amount of resources required to compute a single instance of the function or relation? Suppose solving one instance of some problem with success probability at least $1-\varepsilon$ requires c units of some resource. A natural way to solve k independent instances of this problem would be to solve them independently, which requires ck units of the resource. A *direct sum theorem* for this problem would state that any algorithm for solving k instances which uses $o(ck)$ units of resource has success probability at most $1-\varepsilon$. A *direct product theorem* for the problem would state that any algorithm for solving k instances that uses $o(ck)$ units of resource has success probability at most $(1-\varepsilon)^{\Omega(k)}$. Hence a direct product theorem is the stronger result of the two.

In this paper, we deal with direct product theorems in the model of communication complexity. In this model, there are two parties Alice and Bob, who receive inputs x and y respectively, and wish to jointly compute a relation f. They can use local computation, public coins, and communicate with each other using classical messages, in the classical model; use local unitaries, shared entanglement, and communicate with each other using quantum messages, in the quantum model. The resource of interest is the number of bits/qubits communicated; so the parties are allowed to share an arbitrary amount of randomness or entanglement, and perform local operations of arbitrary complexity.

Direct product theorems in communication are related to *parallel repetition theorems* for *non-local games*. In a non-local game, two parties Alice and Bob are given inputs x and y respectively from some specified distribution, and without communicating with each other, they are required to give answers a and b respectively to a referee. They are considered to win the game if $\mathsf{V}(a,b,x,y)$ holds for a specified predicate V. In the classical model, the players are allowed to share randomness, and in the quantum model they are allowed to share entanglement. A parallel repetition theorem shows that the maximum probability of winning k independent instances of a non-local game is $p^{\Omega(k)}$, if the maximum probability of winning a single instance of it is p, regardless of the amount of shared randomness or entanglement used. Direct product theorems in communication are often proved by combining techniques used to prove direct sum theorems in communication, which require message compression, and parallel repetition theorems for games.

In classical communication complexity, there is a long line of works on direct sum and direct-product theorems including [40, 14, 1, 41, 27, 28, 30, 5, 38, 44, 22, 21, 18, 35, 32, 2, 12, 11, 10, 7, 13, 20, 25, 37, 9, 43]. A parallel repetition theorem for the classical value of general two-player non-local games was first shown by Raz [39], and the proof was subsequently simplified by Holenstein [19].

In quantum communication complexity, a direct sum theorem is known for the entanglement-assisted one-way [30], *simultaneous-message-passing* (SMP), entanglement-assisted [30] and unassisted models [21]. A strong parallel repetition theorem for the quantum value of a general two-player non-local game is not known. Parallel repetition theorems were shown for special classes of games such as XOR games [15], unique games [34] and projection games [17]. When the type of game is not restricted but the input distribution is, parallel repetition theorems have been shown under product distributions [24] and *anchored* distributions [4, 3]. For general games under general distributions, the best current result is due to Yuen [46], which shows that the quantum value of k parallel instances of a general game goes down polynomially in k, if the quantum value of the original game is strictly less than 1. No direct product theorems for quantum communication for a general function had previously been known. However, a direct product theorem has been shown for the generalized discrepancy method [42], which is a lower bound technique that often characterizes (multi-round) quantum communication complexity. [5] showed a direct product theorem for functions whose one-way quantum communication is characterized by VC dimension, and [36] showed a direct product theorem for symmetric functions.

Combining ideas from Jain, Pereszlényi and Yao [24] and the message compression scheme from Jain, Radhakrishnan and Sen [30], it is possible to show a strong direct product theorem for one-way quantum communication under product distributions. To deal with non-product distributions, we borrow the idea of anchored distributions due to Bavarian, Vidick and Yuen [4, 3], which allows us to prove a direct product theorem for the worst case one-way quantum communication complexity of a general function. We make some crucial changes in the definition of correlation-breaking random variable as used by [4] which help us use one-sided anchored distribution and simplify their proof. This simplification is in fact crucial for us to combine the anchored distribution technique with the message compression argument of [30] in the communication complexity setting. We elaborate further on our proof techniques in Section 1.2.

Parallel repetition and direct product theorems have a number of applications. For example, Raz's parallel repetition theorem [39] can be used to prove the PCP theorem [16]; the [4] parallel repetition theorem was used to prove the recent $\mathrm{MIP}^* = \mathrm{RE}$ result [33]. Sherstov's direct product theorem for generalized discrepancy was used in [8] to prove a near-optimal lower bound on the bounded-round quantum communication complexity of set disjointness. [36] used their direct product theorem to prove time-space tradeoffs for solving certain problems. We expect our result to have similar applications.

1.1 Our results

Let $\mathrm{Q}^1_\varepsilon(f)$ denote that the one-way entanglement-assisted quantum communication complexity of a relation f, with worst-case error ε. Let f^k denote k parallel instances of f. Our strong direct product theorem is as follows.

▶ **Theorem 1.** *For any relation $f \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, and any $0 < \varepsilon, \zeta < \frac{1}{2}$,*

$$\mathrm{Q}^1_{1-(1-\varepsilon)^{\Omega(\zeta^6 k/\log|\mathcal{Z}|)}}(f^k) = \Omega\left(k\left(\zeta^5 \cdot \mathrm{Q}^1_{\varepsilon+\zeta}(f) - \log\log(1/\zeta)\right)\right).$$

Let $\omega^*(G)$ represent the entangled value of a two-player non-local game G, and let G^k denote k parallel instances of G. We call a distribution q on $\mathcal{X} \times \mathcal{Y}$ *anchored on one side* with *anchoring probability* ζ if one of the following conditions holds:

(i) There exists an $x^* \in \mathcal{X}$ such that $q(x^*) = \zeta$ and $q(y|x^*) = q(y)$ for all $y \in \mathcal{Y}$,

(ii) There exists a $y^* \in \mathcal{Y}$ such that $q(y^*) = \zeta$ and $q(x|y^*) = q(x)$ for all $x \in \mathcal{X}$.

The game will be called *anchored on both sides* with anchoring probability ζ if both conditions hold simultaneously.

Then our parallel repetition theorem is stated as follows.

▶ **Theorem 2.** *For a two-player non-local game $G = (q, \mathcal{X} \times \mathcal{Y}, \mathcal{A} \times \mathcal{B}, \mathsf{V})$ such that q is a distribution anchored on one side with anchoring probability ζ,*

$$\omega^*(G^k) = \left(1 - (1 - \omega^*(G))^5\right)^{\Omega\left(\frac{\zeta^2 k}{\log(|\mathcal{A}|\cdot|\mathcal{B}|)}\right)}.$$

One can get a game anchored on one side (say the $\mathcal{Y}$ side) from a general game in the following way: in the anchored game, the referee chooses (x, y) from the original probability distribution, and with probability ζ replaces y with a new input y^*. If Bob's input is y^*, then the referee accepts any answer from the players. In a game anchored on both sides, the referee must instead replace x with x^* and y with y^* independently with probability ζ, and accept if either Alice's input is x^* or Bob's input is y^*. It is clear that anchoring makes the game easier. In this light, a parallel repetition theorem for anchoring games can be thought of as follows: for a general game G, there exists a simple transformation taking it to another game $\tilde{G}$ such that

1. If $\omega^*(G) = 1$, then $\omega^*(\tilde{G}^k) = 1$.
2. If $\omega^*(G) < 1$, then $\omega^*(\tilde{G}^k) = \exp(-\Omega(k))$.

The merit of our result here is that the transformation involved for anchoring on one side changes the game less than the transformation involved in anchoring it on both sides.

We note that the definition of anchoring used in [4, 3] is more general: instead of single inputs x^*, y^*, they consider anchoring sets $\mathcal{X}^* \subseteq \mathcal{X}$ and $\mathcal{Y}^* \subseteq \mathcal{Y}$, such that $q(\mathcal{X}^*), q(\mathcal{Y}^*) \geq \zeta$, and whenever $x \in \mathcal{X}^*$ or $y \in \mathcal{Y}^*$, $q(x, y) = q(x)q(y)$. However, it appears this generalized definition is not more useful from the perspective of anchoring transformations. While our technique could go through for the one-sided version of this definition of anchoring, we do not state or prove it as such for the sake of simplicity.

Unlike in the case of communication, worst-case success probability is usually not considered for non-local games. But one could define a game $G_{\mathrm{wc}} = (\mathcal{X} \times \mathcal{Y}, \mathcal{A} \times \mathcal{B}, \mathsf{V})$ without an associated distribution, and the worst-case winning probability ω^*_{wc} over all inputs of this can be considered. As long as Alice and Bob are allowed to share randomness (which they are, in the quantum case), Yao's lemma [45] holds just like in the case of communication, relating the worst-case winning probability to distributional winning probability. Hence, by choosing $\zeta = (1 - \omega^*_{\mathrm{wc}}(G_{\mathrm{wc}}))/2$ and using the same arguments as in the case of communication, Theorem 2 leads to the following corollary about the worst-case winning probability of any game.

▶ **Corollary 3.** *For any two-player non-local game $G_{\mathrm{wc}} = (\mathcal{X} \times \mathcal{Y}, \mathcal{A} \times \mathcal{B}, \mathsf{V})$,*

$$\omega^*_{\mathrm{wc}}(G^k_{\mathrm{wc}}) = \left(1 - (1 - \omega^*_{\mathrm{wc}}(G_{\mathrm{wc}}))^7\right)^{\Omega\left(\frac{k}{\log(|\mathcal{A}|\cdot|\mathcal{B}|)}\right)}.$$

This is in fact also implied by the result of [4], although it is not explicitly observed by them.

1.2 Proof overview

We describe how to prove the parallel repetition and direct product theorems in the distributional setting first, and we shall later describe how to go from there to the worst case setting. We use the information theoretic framework for parallel repetition established by [39] and [19].The broad idea is as follows: for a given relation $\tilde{f} \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, let the one-way quantum communication required to compute a single copy with constant success be c. Now consider a one-way quantum protocol $\mathcal{P}$ for $\tilde{f}^k$ which has communication $o(ck)$, in which we can condition on the success of some t coordinates. If the success probability in these

t coordinates is already as small as we want, then we are done. Otherwise, we exhibit a $(t+1)$-th coordinate i, such that conditioned on the success on the t coordinates, the success of i in $\mathcal{P}$ is bounded away from 1. This is done by showing that if the success probability in the t coordinates is not too small, then we can give a protocol $\mathcal{P}'$ for $\tilde{f}$ whose communication is $o(c)$ and whose success probability is constant – a contradiction.

$\mathcal{P}'$ works by embedding its input into the i-th coordinate of a shared quantum state representing the final input, output, message and discarded registers of $\mathcal{P}$, conditioned on the success event in the t coordinates, which we denote by $\mathcal{E}$. Suppose the quantum state conditioned on $\mathcal{E}$, when Alice and Bob's inputs are x_i and y_i respectively at the i-th coordinates, is $|\varphi\rangle_{x_iy_i}$. On input (x_i, y_i) in $\mathcal{P}'$, Alice and Bob will by means of local unitaries and communication try to get the shared state close to $|\varphi\rangle_{x_iy_i}$, on which Bob can perform a measurement to get an outcome z_i. The state $|\varphi\rangle_{x_iy_i}$ is such that the resulting probability distribution $\mathsf{P}_{X_iY_iZ_i}$ is the distribution of $X_iY_iZ_i$ in $\mathcal{P}$ conditioned on success. Hence our proof mainly consists of showing how Alice and Bob can get the shared state close to $|\varphi\rangle_{x_iy_i}$. The proof technique for a parallel repetition theorem is the same, except one cannot, and need not, use communication to get the shared state $|\varphi\rangle_{x_iy_i}$ there.

1.2.1 Product distribution parallel repetition

In [24] the following three states are considered: $|\varphi\rangle_{x_i}$ which is the superposition of $|\varphi\rangle_{x_iy_i}$ over the distribution of Y_i, $|\varphi\rangle_{y_i}$ which is the superposition over the distribution of X_i, and $|\varphi\rangle$ which is the superposition over both. In this setting, $X_1 \ldots X_k$ are initially in product with all of Bob's registers and $Y_1 \ldots Y_k$ are in product with all of Alice's registers. If the probability of $\mathcal{E}$ is large, then conditioning on it, the following can be shown:

1. By chain rule of mutual information, there is an X_i whose mutual information with Bob's registers in $|\varphi\rangle$ is small. Hence by Uhlmann's theorem, there exist unitaries U_{x_i} acting on Alice's registers that take $|\varphi\rangle$ close to $|\varphi\rangle_{x_i}$.
2. Similarly, the mutual information between Y_i and Alice's registers in $|\varphi\rangle$ is small, and hence there exist unitaries U_{y_i} acting on Bob's registers that take $|\varphi\rangle$ close to $|\varphi\rangle_{y_i}$.
3. Since U_{x_i} and U_{y_i} act on disjoint registers, using a commuting argument and the monotonicity of ℓ_1 distance under quantum operations, $U_{x_i} \otimes U_{y_i}$ takes $|\varphi\rangle$ close to $|\varphi\rangle_{x_iy_i}$.

Alice and Bob can thus share $|\varphi\rangle$ as entanglement, and get close to $|\varphi\rangle_{x_iy_i}$ by local operations.

1.2.2 Product distribution direct product

It is possible to combine techniques from the product parallel repetition theorem above and a message compression technique from [30] to give a direct product theorem for one-way quantum communication complexity under product distributions, and we give a proof outline here.

If the communication protocol involves a message from Alice to Bob, we cannot then get the state $|\varphi\rangle_{x_iy_i}$ by applying Uhlmann unitaries on both Alice and Bob's registers: because of Alice's message, the dependence of $|\varphi\rangle_{x_iy_i}$ on x_i can be quite large. Instead, we use the result of [26, 30] to do the transformation from $|\varphi\rangle$ to $|\varphi\rangle_{x_i}$ on Alice's side via a projector instead. By [30], as long as $|\varphi\rangle$ is the superposition of $|\varphi\rangle_{x_i}$ over the X_i distribution, such a projector Π_{x_i} always exists and its success probability depends on the mutual information between X_i and Bob's registers. This success probability is not close to 1, but as long as it is not too small, Alice and Bob can share multiple copies of $|\varphi\rangle$ and Alice can perform the $\{\Pi_{x_i}, \mathbb{1} - \Pi_{x_i}\}$ measurement on all of them. With high probability, she succeeds on at least one copy, and her message to Bob is then just the index of the copy she succeeds on.

Overall, the steps analogous to the parallel repetition proof are as follows:

1. If the message size in $\mathcal{P}$ is $o(ck)$ bits, by the chain rule of mutual information, the information between X_i and Bob's registers is $o(c)$. Hence by [30], there exist projectors Π_{x_i} acting on Alice's registers, which succeed with probability $2^{-o(c)}$ on $|\varphi\rangle$, and on success, take $|\varphi\rangle$ close to $|\varphi\rangle_{x_i}$.
2. Since there is no communication from Bob to Alice, by the same argument as in the case for games, there exist unitaries U_{y_i} acting on Bob's registers, that take $|\varphi\rangle$ close to $|\varphi\rangle_{y_i}$.
3. By the same commuting argument, conditioned on the success of Π_{x_i}, $\Pi_{x_i} \otimes U_{y_i}$ takes $|\varphi\rangle$ close to $|\varphi\rangle_{x_iy_i}$.

Hence there is a communication protocol with prior shared entanglement between Alice and Bob to obtain a state close to $|\varphi\rangle_{x_iy_i}$ on inputs (x_i, y_i): Alice and Bob share $2^{o(c)}$ copies of $|\varphi\rangle_{y^*}$ as entanglement; Alice performs the Π_{x_i} measurement on all these copies, and succeeds on at least one copy with high probability. She sends the index of the copy on which she succeeds to Bob, who performs U_{y_i} on the same copy. This protocol has communication $o(c)$, since that is how many classical bits Alice needs in order to encode the index of the successful copy out of $2^{o(c)}$ copies.

1.2.3 Anchored distribution parallel repetition

[3] in their parallel repetition theorem use anchored distributions, which are non-product distributions that "look like" product distributions. However, since overall $X_1 \dots X_k$ are not initially in product with $Y_1 \dots Y_k$, one needs to use what are known as *correlation-breaking variables*. For each i, correlation-breaking variables D_iG_i are such that conditioned on D_iG_i, X_i and Y_i are independent. In particular, D_i is a uniformly distributed bit, and G_i takes values in either $\mathcal{X}$ or $\mathcal{Y}$ depending on whether D_i is 0 or 1, and is highly correlated with either X_i or Y_i in the respective cases. This means that conditioned on $D_i = 0$, $G_i = x^*$ with probability $\Omega(\zeta)$ and conditioned on $D_i = 1$, $G_i = y^*$ with probability $\Omega(\zeta)$.

1. The mutual information between X_i and Bob's registers in $|\varphi\rangle$ conditioned on $D_i = 1$ and G_i is small. Further conditioning on $G_i = y^*$ (which happens with constant probability), the mutual information between X_i and Bob's registers in $|\varphi\rangle_{y^*}$ is small. Hence by Uhlmann's theorem, there exist unitaries U_{x_i} on Alice's registers, taking $|\varphi\rangle_{x^*y^*}$ close to $|\varphi\rangle_{x_iy^*}$.
2. Similarly, the mutual information between Y_i and Alice's registers in $|\varphi\rangle$ conditioning on $D_i = 0$ and $G_i = x^*$ is small, which means there exist unitaries U_{y_i} on Bob's registers, taking $|\varphi\rangle_{x^*y^*}$ close to $|\varphi\rangle_{x^*y_i}$.
3. Using an involved argument, it is possible to show that $U_{x_i} \otimes U_{y_i}$ takes $|\varphi\rangle_{x^*y^*}$ close to $|\varphi\rangle_{x_iy_i}$.

Alice and Bob can thus share $|\varphi\rangle_{x^*y^*}$ in this case, and get close to $|\varphi\rangle_{x_iy_i}$ by local operations.

1.2.4 Anchored distribution direct product

In our direct product proof, since the distribution is anchored on one side, we use correlation-breaking variables that are identical to those in [3] in the $D_i = 1$ case, but in the $D_i = 0$ we consider a simpler distribution where G_i is perfectly correlated with X_i. Here we also clarify what we mean by G_i and Y_i being highly correlated when $D_i = 1$: if $G_i = y^*$, then Y_i is always y^*; but if $G_i = y_i$ for $y_i \neq y^*$, then Y_i still takes value y^* with probability $\Omega(\zeta)$, and is y_i otherwise. The distribution of X_i conditioned on $G_i = y^*$ is the marginal distribution of X_i, while conditioned on y_i, it is the same as the distribution of X_i conditioned on $Y_i = y_i$ (potentially different from the marginal distribution of X_i). Our use of these correlation-breaking variables is quite different from that in [3], however.

1. If the message size is $o(ck)$, the mutual information between X_i and Bob's registers in $|\varphi\rangle$ is $o(c)$, conditioned on $D_i = 1, G_i = y^*$. Since the distribution is anchored on Bob's side, this means that the mutual information between X_i and Bob's registers in $|\varphi\rangle_{y^*}$ is $o(c)$. By [30], there exist projectors Π_{x_i} acting on Alice's registers, which succeed with probability $2^{-o(c)}$ on $|\varphi\rangle_{y^*}$, and on success take it close to $|\varphi\rangle_{x_i y^*}$.
2. The mutual information between Y_i and Alice's registers conditioned on $D_i = 1, G_i \neq y^*$ is small. For each value of $G_i \neq y^*$, there exist only two possible values of Y_i: y_i and y^*, and hence Alice's registers in $|\varphi\rangle_{y_i}$ and $|\varphi\rangle_{y^*}$ must be close on average. By Uhlmann's theorem, there exist unitaries U_{y_i} acting on Bob's registers, taking $|\varphi\rangle_{y^*}$ close to $|\varphi\rangle_{y_i}$.
3. Since the marginal distribution of X_i conditioned on $G_i = y_i$ is approximately the same as the marginal distribution of X_i conditioned on $Y_i = y_i$, we can show by the same commuting argument that conditioned on success of Π_{x_i}, $\Pi_{x_i} \otimes U_{y_i}$ takes $|\varphi\rangle_{y^*}$ close to $|\varphi\rangle_{x_i y_i}$.

Hence there is a communication protocol with prior shared entanglement which allows Alice and Bob to obtain a state close to $|\varphi\rangle_{x_i y_i}$ as a shared state on input (x_i, y_i): this works just like the communication protocol for the product case, except the initial shared entanglement is $2^{o(c)}$ copies of $|\varphi\rangle_{y^*}$ instead. We note that our step 3 above is the simpler argument used in [24] and the product distribution direct product, instead of the more involved technique from [4].

1.2.5 Simplified anchored distribution parallel repetition

Our anchored distribution parallel repetition proof is the same as the anchored direct product proof, except no communication is necessary, since there was no communication in the original protocol. Instead of a projector on Alice's registers taking $|\varphi\rangle_{y^*}$ close to $|\varphi\rangle_{x_i y^*}$, in this case we will have a unitary U_{x_i} doing it. We can argue identically to the direct product proof that there exist U_{y_i} taking $|\varphi\rangle_{y^*}$ close to $|\varphi\rangle_{y_i}$, and $U_{x_i} \otimes U_{y_i}$ takes $|\varphi\rangle_{y^*}$ close to $|\varphi\rangle_{x_i y_i}$.

Our simplification of the techniques [4] is crucial to our direct product proof: we need to use the commuting argument from [30, 24] in order to make use of the message compression scheme. It is not clear whether the involved argument in [4] for the existence of $U_{x_i} \otimes V_{y_i}$ that takes $|\varphi\rangle_{x^* y^*}$ to $|\varphi\rangle_{x_i y_i}$ can work when there needs to be a projector rather than a unitary on Alice's side.

1.2.6 From anchored distribution to worst case direct product

The above argument proves a direct product theorem for the distributional one-way quantum communication complexity of under anchored distributions. However, what we are actually interested in is a direct product theorem for the worst case one-way quantum communication complexity. To get this for a relation f, we consider the distribution under which the distributional communication complexity is equal to the worst case communication complexity of f – this is guaranteed to exist by Yao's lemma. We do an anchoring transformation on f with this distributon to get $\tilde{f}$ with an anchored distribution. Note that it is fine if we can lower bound the distributional communication complexity of $\tilde{f}^k$ with success probability $(1-\varepsilon)^{\Omega(k)}$ under an anchored distribution by k times the worst case communication complexity of f with success probability δ. This is because f^k is harder than $\tilde{f}^k$, and the worst case communication complexity of $\tilde{f}^k$ is lower bounded by its distributional communication complexity under any distribution. By the argument described above, we can lower bound the distributional communication complexity of $\tilde{f}^k$ under the k-tensored anchored distribution with success probability $(1-\varepsilon)^{\Omega(k)}$ by k times the distributional communication complexity of $\tilde{f}$ under the anchored distribution. Now it is easy to go from a distributional protocol for $\tilde{f}$ under the anchored distribution to a protocol for f under the original hard distribution decreasing the success probability by only $O(\zeta)$, since the anchoring transformation only disturbs the original distribution by this amount.

2 Preliminaries

2.1 Probability theory

We shall denote the probability distribution of a random variable X on some set $\mathcal{X}$ by P_X. For any event $\mathcal{E}$ on $\mathcal{X}$, the distribution of X conditioned on $\mathcal{E}$ will be denoted by $\mathsf{P}_{X|\mathcal{E}}$. For joint random variables XY, $\mathsf{P}_{X|Y=y}(x)$ is the conditional distribution of X given $Y = y$; when it is clear from context which variable's value is being conditioned on, we shall often shorten this to $\mathsf{P}_{X|y}$. We shall use $\mathsf{P}_{XY}\mathsf{P}_{Z|X}$ to refer to the distribution

$$(\mathsf{P}_{XY}\mathsf{P}_{Z|X})(x, y, z) = \mathsf{P}_{XY}(x, y) \cdot \mathsf{P}_{Z|X=x}(z).$$

For two distributions P_X and $\mathsf{P}_{X'}$ on the same set $\mathcal{X}$, the ℓ_1 distance between them is defined as

$$\|\mathsf{P}_X - \mathsf{P}_{X'}\|_1 = \sum_{x\in\mathcal{X}} |\mathsf{P}_X(x) - \mathsf{P}_{X'}(x)|.$$

▶ **Fact 4.** *For joint distributions P_{XY} and $\mathsf{P}_{X'Y'}$ on the same sets,*

$$\|\mathsf{P}_X - \mathsf{P}_{X'}\|_1 \le \|\mathsf{P}_{XY} - \mathsf{P}_{X'Y'}\|_1.$$

▶ **Fact 5.** *For two distributions P_X and $\mathsf{P}_{X'}$ on the same set and an event $\mathcal{E}$ on the set,*

$$|\mathsf{P}_X(\mathcal{E}) - \mathsf{P}_{X'}(\mathcal{E})| \le \frac{1}{2}\|\mathsf{P}_X - \mathsf{P}_{X'}\|_1.$$

▶ **Fact 6.** *For two distributions P_X and $\mathsf{P}_{X'}$ on the same set, and any joint distribution $\mathsf{P}_{XX'}$ whose marginals are P_X and $\mathsf{P}_{X'}$ respectively, we have*

$$\|\mathsf{P}_X - \mathsf{P}_{X'}\|_1 \le 2\mathsf{P}_{XX'}(X \ne X').$$

▶ **Fact 7.** *Suppose probability distributions $\mathsf{P}_X, \mathsf{P}_{X'}$ satisfy $\|\mathsf{P}_X - \mathsf{P}_{X'}\|_1 \le \varepsilon$, and an event $\mathcal{E}$ satisfies $\mathsf{P}_X(\mathcal{E}) \ge \alpha$, where $\alpha > \varepsilon$. Then,*

$$\|\mathsf{P}_{X|\mathcal{E}} - \mathsf{P}_{X'|\mathcal{E}}\|_1 \le \frac{2\varepsilon}{\alpha}.$$

Proof. From Fact 5, $\alpha - \varepsilon/2 \le \mathsf{P}_{X'}(\mathcal{E}) \le \alpha + \varepsilon/2$. By definition, there exists an event $\mathcal{E}'$ such that $2(\mathsf{P}_{X|\mathcal{E}}(\mathcal{E}') - \mathsf{P}_{X'|\mathcal{E}}(\mathcal{E}')) = \|\mathsf{P}_{X|\mathcal{E}} - \mathsf{P}_{X'|\mathcal{E}}\|_1$. Now, $\mathsf{P}_X(\mathcal{E} \wedge \mathcal{E}') = \mathsf{P}_X(\mathcal{E})\mathsf{P}_{X|\mathcal{E}}(\mathcal{E}') \ge \alpha\mathsf{P}_{X|\mathcal{E}}(\mathcal{E}')$. Similarly, $\mathsf{P}_{X'}(\mathcal{E} \wedge \mathcal{E}') \le (\alpha + \varepsilon/2)\mathsf{P}_{X'|\mathcal{E}}(\mathcal{E}') \le \alpha\mathsf{P}_{X'|\mathcal{E}}(\mathcal{E}') + \frac{1}{2}\|\mathsf{P}_X - \mathsf{P}_{X'}\|_1$.

Now,

$$\begin{aligned}\|\mathsf{P}_X - \mathsf{P}_{X'}\|_1 &\ge 2(\mathsf{P}_X(\mathcal{E} \wedge \mathcal{E}') - \mathsf{P}_{X'}(\mathcal{E} \wedge \mathcal{E}')) \\ &\ge 2\alpha(\mathsf{P}_{X|\mathcal{E}}(\mathcal{E}') - \mathsf{P}_{X'|\mathcal{E}}(\mathcal{E}')) - \|\mathsf{P}_X - \mathsf{P}_{X'}\|_1 \\ &\ge \alpha\|\mathsf{P}_{X|\mathcal{E}} - \mathsf{P}_{X'|\mathcal{E}}\|_1 - \|\mathsf{P}_X - \mathsf{P}_{X'}\|_1\end{aligned}$$

which gives the required result. ◀

▶ **Fact 8** ([3], Lemma 16). *Suppose XYZ are random variables satisfying $\mathsf{P}_{XY}(x,y^*) = \alpha \cdot \mathsf{P}_X(x)$ for all x. Then,*

$$\left\|\mathsf{P}_{XYZ} - \mathsf{P}_{XY}\mathsf{P}_{Z|X,y^*}\right\|_1 \le \frac{2}{\alpha}\left\|\mathsf{P}_{XYZ} - \mathsf{P}_{XY}\mathsf{P}_{Z|X}\right\|_1.$$

▶ **Corollary 9.** *Supose P_{XY} and $\mathsf{P}_{X'Y'Z'}$ are distributions such that $\mathsf{P}_X(x,y^*) = \alpha \cdot \mathsf{P}_X(x)$ for all x. Then,*

$$\|\mathsf{P}_{X'Z'|y^*} - \mathsf{P}_{X'Z'}\|_1 \le \frac{11}{\alpha}\|\mathsf{P}_{X'Y'Z'} - \mathsf{P}_{XY}\mathsf{P}_{Z'|X'}\|_1.$$

Proof. Let $\|\mathsf{P}_{X'Y'Z'} - \mathsf{P}_{XY}\mathsf{P}_{Z'|X'}\|_1 = \varepsilon$. Note that

$$\|\mathsf{P}_{X|y^*} - \mathsf{P}_{X'|y^*}\|_1 \le \frac{2\varepsilon}{\alpha}$$

by Fact 7. Let $\mathsf{P}_{XYZ''}$ denote the distribution $\mathsf{P}_{XY}\mathsf{P}_{Z'|X'Y'}$.

$$\begin{aligned}
\|\mathsf{P}_{X'Z'} - \mathsf{P}_{XZ''}\|_1 &= \sum_{x,z}\left|\mathsf{P}_{X'}(x)\sum_y \mathsf{P}_{Y'|x}(y)\mathsf{P}_{Z'|xy}(z) - \mathsf{P}_X(x)\sum_y \mathsf{P}_{Y|x}(y)\mathsf{P}_{Z'|xy}(z)\right| \\
&\le \sum_{x,y,z}\left|\mathsf{P}_{X'}(x)\mathsf{P}_{Y'|x}(y) - \mathsf{P}_X(x)\mathsf{P}_{Y|x}(y)\right|\mathsf{P}_{Z'|xy}(z) \\
&= \|\mathsf{P}_{X'Y'} - \mathsf{P}_{XY}\|_1 \le \varepsilon.
\end{aligned}$$

$$\begin{aligned}
\|\mathsf{P}_{XYZ''} - \mathsf{P}_{XY}\mathsf{P}_{Z''|X}\|_1 &\le \|\mathsf{P}_{XYZ''} - \mathsf{P}_{X'Y'Z'}\|_1 + \|\mathsf{P}_{X'Y'Z'} - \mathsf{P}_{XY}\mathsf{P}_{Z'|X'}\|_1 \\
&\quad + \|\mathsf{P}_{XY}\mathsf{P}_{Z'|X'} - \mathsf{P}_{XY}\mathsf{P}_{Z''|X}\|_1 \\
&= \|\mathsf{P}_{XY} - \mathsf{P}_{X'Y'}\|_1 + \|\mathsf{P}_{X'Y'Z'} - \mathsf{P}_{XY}\mathsf{P}_{Z'|X'}\|_1 \\
&\quad + \sum_{x,y}\mathsf{P}_{XY}(x,y)\|\mathsf{P}_{Z'|x} - \mathsf{P}_{Z''|x}\|_1 \\
&\le 2\varepsilon + \sum_x \mathsf{P}_X(x)\sum_{y,z}|\mathsf{P}_{Y|x}(y) - \mathsf{P}_{Y'|x}(y)|\mathsf{P}_{Z'|xy}(z) \\
&\le 2\varepsilon + \sum_{x,y}|\mathsf{P}_X(x)\mathsf{P}_{Y|x}(y) - \mathsf{P}_{X'}(x)\mathsf{P}_{Y'|x}(y)| \\
&\quad + \sum_{x,y}|\mathsf{P}_{X'}(x) - \mathsf{P}_X(x)|\mathsf{P}_{Y'|x}(y) \\
&\le 2\varepsilon + 2\|\mathsf{P}_{XY} - \mathsf{P}_{X'Y'}\|_1 \le 4\varepsilon.
\end{aligned}$$

Combining all this,

$$\begin{aligned}
\|\mathsf{P}_{X'Z'|y^*} - \mathsf{P}_{X'Z'}\|_1 &\le \|\mathsf{P}_{X'Z'|y^*} - \mathsf{P}_{XZ''|y^*}\|_1 + \|\mathsf{P}_{XZ''|y^*} - \mathsf{P}_{XZ''}\|_1 + \|\mathsf{P}_{XZ''} - \mathsf{P}_{X'Z'}\|_1 \\
&\le \|\mathsf{P}_{X|y^*} - \mathsf{P}_{X'|y^*}\|_1 + \|\mathsf{P}_{XZ''|y^*} - \mathsf{P}_{XZ''}\|_1 + \|\mathsf{P}_{XZ''} - \mathsf{P}_{X'Z'}\|_1 \\
&\le \frac{2\varepsilon}{\alpha} + \frac{2}{\alpha}\|\mathsf{P}_{XYZ''} - \mathsf{P}_{XY}\mathsf{P}_{Z''|X}\|_1 + \varepsilon \\
&\le \frac{2\varepsilon}{\alpha} + \frac{8\varepsilon}{\alpha} + \varepsilon \le \frac{11\varepsilon}{\alpha}.
\end{aligned}$$

where we have used Lemma 8 in the third inequality. ◀

▶ **Fact 10** ([19], Corollary 6). *Let $\mathsf{P}_{TU_1\ldots U_kV} = \mathsf{P}_T\mathsf{P}_{U_1|T}\mathsf{P}_{U_2|T}\ldots\mathsf{P}_{U_k|T}\mathsf{P}_{V|TU_1\ldots U_k}$ be a probability distribution over $\mathcal{T}\times\mathcal{U}^k\times\mathcal{V}$, and let $\mathcal{E}$ be any event. Then,*

$$\sum_{i=1}^{k}\|\mathsf{P}_{TU_iV|\mathcal{E}} - \mathsf{P}_{TV|\mathcal{E}}\mathsf{P}_{U_i|T}\|_1 \le \sqrt{k\left(\log(|\mathcal{V}|) + \log\left(\frac{1}{\Pr[\mathcal{E}]}\right)\right)}.$$

▶ **Definition 11** ([19]). *For two distributions P_{XY} and $\mathsf{P}_{X'Y'ST}$, we say (X,Y) is $(1-\varepsilon)$-embeddable in $(X'S, Y'T)$ if there exists a random variable R on a set $\mathcal{R}$ independent of XY and functions $f_A : \mathcal{X} \times \mathcal{R} \to \mathcal{S}$ and $f_B : \mathcal{Y} \times \mathcal{R} \to \mathcal{T}$, such that*

$$\|\mathsf{P}_{XYf_A(X,R)f_B(X,R)} - \mathsf{P}_{X'Y'ST}\|_1 \leq \varepsilon.$$

▶ **Fact 12** ([19, 25]). *If two distributions P_{XY} and $\mathsf{P}_{X'Y'R'}$ satisfy*

$$\|\mathsf{P}_{X'Y'R'} - \mathsf{P}_{XY}\mathsf{P}_{R'|X'}\|_1 \leq \varepsilon \qquad \|\mathsf{P}_{X'Y'R'} - \mathsf{P}_{XY}\mathsf{P}_{R'|Y'}\|_1 \leq \varepsilon,$$

then (X,Y) is $(1-5\varepsilon)$-embeddable in $(X'R', Y'R')$.[1]

2.2 Quantum information

The ℓ_1 distance between two quantum states ρ and σ is given by

$$\|\rho - \sigma\|_1 = \mathrm{Tr}\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)} = \mathrm{Tr}|\rho - \sigma|.$$

The fidelity between two quantum states is given by

$$\mathsf{F}(\rho,\sigma) = \|\sqrt{\rho}\sqrt{\sigma}\|_1.$$

ℓ_1 distance and fidelity are related in the following way.

▶ **Fact 13** (Fuchs-van de Graaf inequality). *For any pair of quantum states ρ and σ,*

$$2(1-\mathsf{F}(\rho,\sigma)) \leq \|\rho-\sigma\|_1 \leq 2\sqrt{1-\mathsf{F}(\rho,\sigma)^2}.$$

For two pure states $|\psi\rangle$ and $|\phi\rangle$, we have

$$\||\psi\rangle\langle\psi| - |\phi\rangle\langle\phi|\|_1 = \sqrt{1-\mathsf{F}\left(|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|\right)^2} = \sqrt{1-|\langle\psi|\phi\rangle|^2}.$$

▶ **Fact 14** (Uhlmann's theorem). *Suppose ρ and σ are mixed states on register X which are purified to $|\rho\rangle$ and $|\sigma\rangle$ on registers XY, then it holds that*

$$\mathsf{F}(\rho,\sigma) = \max_U |\langle\rho|\mathbb{1}_X \otimes U|\sigma\rangle|$$

where the maximization is over unitaries acting only on register Y.

▶ **Fact 15** (Data-processing inequality). *For a quantum channel $\mathcal{E}$ and states ρ and σ,*

$$\|\mathcal{E}(\rho) - \mathcal{E}(\sigma)\|_1 \leq \|\rho - \sigma\|_1 \qquad \text{and} \qquad \mathsf{F}(\mathcal{E}(\rho), \mathcal{E}(\sigma)) \geq \mathsf{F}(\rho,\sigma).$$

The entropy of a quantum state ρ on a register Z is given by

$$\mathsf{S}(\rho) = -\mathrm{Tr}(\rho\log\rho).$$

The relative entropy between two states ρ and σ of the same dimensions is given by

$$\mathsf{S}(\rho\|\sigma) = \mathrm{Tr}(\rho\log\rho) - \mathrm{Tr}(\rho\log\sigma).$$

[1] This fact is equivalent to Lemma 2.11 in [25], although this lemma is stated in terms of relative entropies instead of trace distances between the various distributions. In the proof of the lemma, the relative entropies are converted to the same trace distances as we consider, using Pinsker's inequality. This justifies our statement of the fact, which is tailored towards our application.

The relative min-entropy between ρ and σ is defined as

$$\mathsf{S}_\infty(\rho\|\sigma) = \min\{\lambda : \rho \le 2^\lambda \sigma\}.$$

It is easy to see that $\mathsf{S}(\rho\|\sigma)$ and $\mathsf{S}_\infty(\rho\|\sigma)$ only take finite values when the support of ρ is contained in the support of σ. Moreover, clearly $0 \le \mathsf{S}(\rho\|\sigma) \le \mathsf{S}_\infty(\rho\|\sigma)$ for all ρ and σ.

The ε-smooth relative min-entropy between ρ and σ is defined as

$$\mathsf{S}^\varepsilon_\infty(\rho\|\sigma) = \inf_{\rho' : \|\rho-\rho'\|_1 \le \varepsilon} \mathsf{S}(\rho'\|\sigma).$$

$\mathsf{S}^\varepsilon_\infty(\rho\|\sigma)$ can take a finite value even if the support of ρ is not contained in the support of σ, for example if ρ is ε-close to a state contained within the support of σ. $\mathsf{S}_\infty(\rho\|\sigma)$ cannot be upper bounded by $\mathsf{S}(\rho\|\sigma)$, but $\mathsf{S}^\varepsilon_\infty(\rho\|\sigma)$ can be, due to the Quantum Substate Theorem.

▶ **Fact 16** (Quantum Substate Theorem, [31, 23])**.** *For any two states ρ and σ such that the support of ρ is contained in the support of σ, and any $\varepsilon > 0$,*

$$\mathsf{S}^\varepsilon_\infty(\rho\|\sigma) \le \frac{4\mathsf{S}(\rho\|\sigma)}{\varepsilon^2} + \log\left(\frac{1}{1-\varepsilon^2/4}\right).$$

▶ **Fact 17** (Pinsker's Inequality)**.** *For any two states ρ and σ, $\|\rho-\sigma\|_1 \le \sqrt{\mathsf{S}(\rho\|\sigma)}$.*

▶ **Fact 18.** *If $\sigma = \varepsilon\rho + (1-\varepsilon)\rho'$, then $\mathsf{S}_\infty(\rho\|\sigma) \le \log(1/\varepsilon)$.*

▶ **Fact 19.** *For any three quantum states ρ, σ, φ such that $\mathrm{supp}(\rho) \subseteq \mathrm{supp}(\varphi) \subseteq \mathrm{supp}(\sigma)$,*

$$\mathsf{S}_\infty(\rho\|\sigma) \le \mathsf{S}_\infty(\rho\|\varphi) + \mathsf{S}_\infty(\varphi\|\sigma).$$

▶ **Fact 20.** *For any unitary U, $\mathsf{S}_\infty(U\rho U^\dagger\|U\sigma U^\dagger) = \mathsf{S}_\infty(\rho\|\sigma)$.*

A state of the form

$$\rho_{XY} = \sum_x \mathsf{P}_X(x)|x\rangle\langle x|_X \otimes \rho_{Y|x}$$

is called a CQ (classical-quantum) state, with X being the classical register and Y being quantum. We shall use X to refer to both the classical register and the classical random variable with the associated distribution. As in the classical case, here we are using $\rho_{Y|x}$ to denote the state of the register Y conditioned on $X = x$, or in other words the state of the register Y when a measurement is done on the X register and the outcome is x. Hence $\rho_{XY|x} = |x\rangle\langle x|_X \otimes \rho_{Y|x}$. When the registers are clear from context we shall often write simply ρ_x.

The mutual information between Y and Z with respect to a state ρ on YZ is defined as

$$\mathsf{I}(Y : Z)_\rho = \mathsf{S}(\rho_{YZ}\|\rho_Y \otimes \rho_Z).$$

The conditional mutual information between Y and Z conditioned on a classical register X, is defined as

$$\mathsf{I}(Y : Z|X) = \mathop{\mathbb{E}}_{\mathsf{P}_X}[\mathsf{I}(Y : Z)_{\rho_x}].$$

Mutual information can be seen to satisfy the chain rule

$$\mathsf{I}(XY : Z)_\rho = \mathsf{I}(X : Z)_\rho + \mathsf{I}(Y : Z|X)_\rho.$$

▶ **Fact 21** ([6], Lemma B.7). *For any quantum state ρ_{YZ},*

$$\inf_{\sigma_Z} \mathsf{S}_\infty(\rho_{YZ} \| \rho_Y \otimes \sigma_Z) \leq 2\min\{\log|\mathcal{Y}|, \log|\mathcal{Z}|\}.$$

▶ **Fact 22.** *For CQ states*

$$\rho_{XY} = \sum_x \mathsf{P}_X(x)|x\rangle\langle x|_X \otimes \rho_{Y|x} \qquad \sigma_{XY} = \sum_x \mathsf{P}_{X'}(x)|x\rangle\langle x|_X \otimes \sigma_{Y|x},$$

their relative entropy is given by

$$\mathsf{S}(\rho_{XY}\|\sigma_{XY}) = \mathsf{S}(\mathsf{P}_X\|\mathsf{P}_{X'}) + \mathop{\mathbb{E}}_{\mathsf{P}_X}[\mathsf{S}(\rho_{Y|x}\|\sigma_{Y|x})].$$

▶ **Fact 23.** *Suppose σ_{XYZ} and ρ_{XYZ} are CQ states defined as follows*

$$\sigma_{XYZ} = \sum_{x,y} \mathsf{P}_{XY}(x,y)|x,y\rangle\langle x,y| \otimes \sigma_{Z|xy} \qquad \rho_{XYZ} = \sum_{x,y} \mathsf{P}_{X'Y'}(x,y)|x,y\rangle\langle x,y| \otimes \sigma_{Z|xy},$$

where $\|\mathsf{P}_{XY} - \mathsf{P}_{X'Y'}\|_1 \leq \delta$. Let $\mathsf{I}(Y:Z|X)_\sigma \leq c$. Then, for any $0 < \varepsilon < \frac{1}{4}$,

$$\mathsf{P}_{X'Y'}\left(\mathsf{S}^\varepsilon_\infty(\sigma_{Z|xy}\|\sigma_{Z|x}) > \frac{4c+1}{\varepsilon^3}\right) \leq \varepsilon + \frac{\delta}{2}.$$

Proof. We have $\mathbb{E}_{\mathsf{P}_{XY}}[\mathsf{S}(\sigma_{Z|xy}\|\sigma_{Z|x})] = \mathsf{I}(Y:Z|X)_\sigma \leq c$. By Markov's inequality, this means that

$$\mathsf{P}_{XY}\left(\mathsf{S}(\sigma_{Z|xy}\|\sigma_{Z|x}) > \frac{c}{\varepsilon}\right) \leq \varepsilon.$$

Using the Quantum Substate Theorem, this implies

$$\mathsf{P}_{XY}\left(\mathsf{S}^\varepsilon_\infty(\sigma_{Z|xy}\|\sigma_{Z|x}) > \frac{4c+1}{\varepsilon^3}\right) \leq \mathsf{P}_{XY}\left(\mathsf{S}^\varepsilon_\infty(\sigma_{Z|xy}\|\sigma_{Z|x}) > \frac{4c}{\varepsilon^3} + \log\left(\frac{1}{1-\varepsilon^2/4}\right)\right) \leq \varepsilon.$$

Since $\|\mathsf{P}_{XY} - \mathsf{P}_{X'Y'}\|_1 \leq \delta$, this gives us the required bound of the probability under $\mathsf{P}_{X'Y'}$. ◀

▶ **Fact 24** (Quantum Raz's Lemma, [3]). *Let ρ_{XY} and σ_{XY} be two CQ states with $X = X_1 \ldots X_k$ being classical, and σ being product across all registers. Then,*

$$\sum_{i=1}^k \mathsf{I}(X_i : Y)_\rho \leq \mathsf{S}(\rho_{XY}\|\sigma_{XY}).$$

▶ **Fact 25** ([29], Lemma 2). *Suppose the state*

$$|\sigma\rangle_{X\tilde{X}AB} = \sum_x \sqrt{\mathsf{P}_X(x)}|xx\rangle_{X\tilde{X}}|\sigma\rangle_{AB|x}$$

satisfies $\mathsf{P}_X(\mathsf{S}^\varepsilon_\infty(\sigma_{B|x}\|\sigma_B) > c) \leq \delta$ for some $\delta > 0$. Then there is a family of measurement operators $\{\Pi_x\}_x$ acting only on $X\tilde{X}A$ such that:

(i) *Each Π_x succeeds with probability $\alpha = 2^{-c/\delta}$ on $|\sigma\rangle_{X\tilde{X}AB}$, i.e., $\|\Pi_x \otimes \mathbb{1}_B|\sigma\rangle\|_2^2 = 2^{-c/\delta}$,*

(ii) *$(\Pi_x \otimes \mathbb{1}_B)|\sigma\rangle\langle\sigma|(\Pi_x \otimes \mathbb{1}_B)$ is of the form $|xx\rangle\langle xx| \otimes \rho_x$, for some state ρ_x on AB, and*

$$\mathop{\mathbb{E}}_{\mathsf{P}_X}\left\|\frac{1}{\alpha}(\Pi_x \otimes \mathbb{1}_B)|\sigma\rangle\langle\sigma|_{X\tilde{X}AB}(\Pi_x \otimes \mathbb{1}_B) - |xx\rangle\langle xx|_{X\tilde{X}} \otimes |\sigma\rangle\langle\sigma|_{AB|x}\right\|_1 \leq \varepsilon + 2\delta.$$

The version of the above fact stated here is slightly different from the original statement in [29], in order to suit our application. In the original statement, $\mathsf{I}(X:B)_\sigma$ is used instead, and the superposition state lacks the $\tilde{X}$ register. However, in the proof of the fact in [29], $\mathsf{I}(X:B)_\sigma$ is converted to $\mathsf{P}_X(\mathsf{S}^\varepsilon_\infty(\sigma_{B|x}\|\sigma_B) > c)$ anyway, so the first change makes no difference. The second change also makes no difference as the same projector that takes the superposition state without the $\tilde{X}$ register to $|x\rangle\langle x| \otimes |\sigma\rangle\langle\sigma|_{AB|x}$ takes the superposition state with the $\tilde{X}$ register to $|xx\rangle\langle xx| \otimes |\sigma\rangle\langle\sigma|_{AB|x}$.

2.3 Quantum communication & entangled games

We briefly describe a quantum communication protocol $\mathcal{P}$ for computing a relation $f \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, between two parties Alice and Bob sharing prior entanglement, with inputs x and y respectively.

In each round, either Alice or Bob will apply a unitary on their classical input register, along with the quantum register they received as a message from the other party in the last round, and memory registers they may have kept from previous rounds; after the unitary they will keep some registers as memory and send the rest to the other party as the message for that round. We can always assume that players make "safe" copies of their inputs using CNOT gates in such protocols, so that the input registers come out as is after each round. We also note that though in general we need not consider shared classical randomness in quantum communication protocols, protocols with shared randomness fall under the shared entanglement framework we have described. This is because shared randomness can be obtained by sharing entanglement and then both parties measuring in the same basis.

In a one-way, i.e., a single round protocol, the memory from previous rounds is replaced by Alice's (who we consider to be sending the single message) part of the shared entangled state, and any register she does not send as a message is simply discarded. After Alice's message, Bob performs a projective measurement on his input register, his part of the shared entanglement, and Alice's message, and gives the outcome of this measurement as the output of the protocol, which we shall denote by $\mathcal{P}(x, y)$. We can of course think of this measurement as Bob performing a unitary on the three registers, and then doing a measurement in the computational basis on some $\log |\mathcal{Z}|$ qubits which are designated for the output.

▶ **Definition 26.** *The one-way entanglement-assisted quantum communication complexity, with error $0 < \varepsilon < 1$, of a relation $f \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, denoted by $\mathrm{Q}^1_\varepsilon(f)$, is the minimum message size, i.e., number of qubits sent, in a one-way entanglement-assisted quantum protocol $\mathcal{P}$ such that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$,*

$$\Pr[\mathcal{P}(x, y) \in f(x, y)] \geq 1 - \varepsilon,$$

where the probability is taken over the inherent randomness in the protocol.

▶ **Definition 27.** *For a probability distribution p on $\mathcal{X} \times \mathcal{Y}$, the distributional one-way entanglement-assisted quantum communication complexity of a relation $f \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, with error $0 < \varepsilon < 1$ with respect to p, is defined as the minimum message size of a one-way entanglement-assisted quantum protocol $\mathcal{P}$ such that*

$$\Pr[\mathcal{P}(x, y) \in f(x, y)] \geq 1 - \varepsilon,$$

where the probability is taken over the distribution p on (x, y) as well as the inherent randomness in the protocol.

▶ **Fact 28** (Yao's lemma, [45]). *For any $0 < \varepsilon < 1$, and any relation f, $\mathrm{Q}^1_\varepsilon(f) = \max_p \mathrm{Q}^1_{p,\varepsilon}(f)$.*

A two-player non-local game G is described as $(q, \mathcal{X} \times \mathcal{Y}, \mathcal{A} \times \mathcal{B}, \mathsf{V})$ where q is a distribution over the input set $\mathcal{X} \times \mathcal{Y}$, $\mathcal{A} \times \mathcal{B}$ is the output set, and $\mathsf{V} : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \times \mathcal{B} \to \{0, 1\}$ is a predicate. It is played as follows: a referee selects inputs (x, y) according to q, sends x to Alice and y to Bob. If Alice and Bob are allowed to share entanglement, they perform measurements on their respective halves of the entangled state along with their respective input registers (which we model as performing unitaries and then measuring in the computational basis on some $\log |\mathcal{A}|$ and $\log |\mathcal{B}|$ qubits designated for outputs respectively), and send their outputs (a, b) back to the referee. The referee accepts and Alice and Bob win the game iff $\mathsf{V}(x, y, a, b) = 1$.

▶ **Definition 29.** *The entangled value of a game $G = (q, \mathcal{X} \times \mathcal{Y}, \mathcal{A} \times \mathcal{B}, \mathsf{V})$, denoted by $\omega^*(G)$, is the maximum winning probability of Alice and Bob, averaged over the distribution q as well as inherent randomness in the strategy, over all shared entanglement strategies for G.*

3 Proof of direct product theorem

In this section, we prove Theorem 1, whose statement we recall below.

▶ **Theorem 1.** *For any relation $f \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, and any $0 < \varepsilon, \zeta < \frac{1}{2}$,*

$$\mathrm{Q}^1_{1-(1-\varepsilon)^{\Omega(\zeta^6 k/\log|\mathcal{Z}|)}}(f^k) = \Omega\left(k\left(\zeta^5 \cdot \mathrm{Q}^1_{\varepsilon+\zeta}(f) - \log\log(1/\zeta)\right)\right).$$

3.1 Setup

Let p be the hard distribution on $\mathcal{X} \times \mathcal{Y}$ for $\mathrm{Q}^1_{\varepsilon+12\zeta}(f)$ from Yao's lemma, i.e., $\mathrm{Q}^1_{\varepsilon+12\zeta}(f) = \mathrm{Q}^1_{p,\varepsilon+12\zeta}(f)$. Consider the relation $\tilde{f} \subseteq \mathcal{X} \times (\mathcal{Y} \cup \{y^*\}) \times \mathcal{Z}$ which is the same as f on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ and additionally,

$$(x, y^*, z) \in \tilde{f} \ \ \forall x \in \mathcal{X}, \forall z \in \mathcal{Z}.$$

We can think of p as a distribution on $\mathcal{X} \times (\mathcal{Y} \cup \{y^*\})$ as well, which has $p(y^*) = 0$. Clearly,

$$\mathrm{Q}^1_{p,\gamma}(\tilde{f}) = \mathrm{Q}^1_{p,\gamma}(f) \tag{1}$$

for any error γ, since p has no support on the extra inputs on which $\tilde{f}$ is defined. We also note that

$$\mathrm{Q}^1_{\gamma}(f^k) \geq \mathrm{Q}^1_{\gamma}(\tilde{f}^k) \tag{2}$$

for any γ. This is because any protocol for f^k is also a protocol for $\tilde{f}^k$: on the indices where Bob's input is y^* instead of an element of $\mathcal{Y}$, he pretends he has gotten an input from $\mathcal{Y}$, runs the protocol with this input and gives the answer accordingly. This gives a correct output if the original protocol gives a correct output, since any output is correct when Bob's input in y^*.

For a distribution q related to p, we shall show that

$$\mathrm{Q}^1_{q^k,1-(1-\varepsilon)^{\Omega(\zeta^6 k/\log|\mathcal{Z}|)}}(\tilde{f}^k) \geq \frac{\zeta^5 k}{60} \cdot \mathrm{Q}^1_{p,\varepsilon+12\zeta}(\tilde{f}) - k\log\log\left(\frac{24}{5\zeta}\right). \tag{3}$$

Since $\mathrm{Q}^1_{\gamma}(\tilde{f}^k) \geq \mathrm{Q}^1_{q^k,\gamma}(\tilde{f}^k)$, (1), (2) and (3) imply the theorem. The distribution q is defined as follows

$$\begin{aligned} q(x,y) &= (1-\zeta)\cdot p(x,y) \ \ \forall x \in \mathcal{X}, y \in \mathcal{Y} \\ q(x,y^*) &= \zeta \cdot p(x) \ \ \forall x \in \mathcal{X}. \end{aligned}$$

Clearly, $q(x,y^*) = q(x)q(y^*)$ for all x, and

$$\|p(x,y) - q(x,y)\|_1 \leq 2\zeta. \tag{4}$$

Following [3], for each $i \in [k]$, we shall define a joint distribution $\mathsf{P}_{X_iY_iD_iG_i}$, where the marginal on X_iY_i is $q(x,y)$, and D_iG_i are correlation-breaking variables such that conditioned on $D_iG_i = d_ig_i$, X_i and Y_i are independent. Each $X_iY_iD_iG_i$ is distributed independently of the rest. Each D_i is distributed uniformly in $\{0,1\}$. Depending on the value of D_i, G_i is distributed in the following way:

$$G_i = \begin{cases} x & \text{w.p. } p(x) & \text{if } D_i = 0 \\ y^* & \text{w.p. } 1-(1-\zeta)^{2/3} & \text{if } D_i = 1 \\ y & \text{w.p. } (1-\zeta)^{2/3}\cdot p(y) & \text{if } D_i = 1 \end{cases}$$

Now depending on the value of D_iG_i, X_iY_i is distributed in the following way:

$$X_iY_i = \begin{cases} (x, y^*) & \text{w.p. } \zeta & \text{if } D_i = 0, G_i = x \\ (x, y) & \text{w.p. } (1-\zeta)\cdot p(y|x) & \text{if } D_i = 0, G_i = x \\ (x, y^*) & \text{w.p. } p(x) & \text{if } D_i = 1, G_i = y^* \\ (x, y^*) & \text{w.p. } \left(1-(1-\zeta)^{1/3}\right)\cdot p(x|y) & \text{if } D_i = 1, G_i = y \\ (x, y) & \text{w.p. } (1-\zeta)^{1/3}\cdot p(x|y) & \text{if } D_i = 1, G_i = y. \end{cases}$$

The following lemma is similar to Claim 18 from [3]; we provide a proof for completeness.

▶ **Lemma 30.** *For all $(x, y) \in \mathcal{X} \times (\mathcal{Y} \cup \{y^*\})$, $\mathsf{P}_{X_iY_i}(x, y) = q(x, y)$.*

Proof. It is trivial to see that $\mathsf{P}_{G_iY_i|D_i=0}(x, y) = \mathsf{P}_{X_iY_i|D_i=0}(x, y) = q(x, y)$, since $G_i = X_i$ conditioned on $D_i = 0$. We now prove the $D_i = 1$ case. First consider a $y \in \mathcal{Y}$. Y_i can only take value y if G_i takes value y. Hence,

$$\begin{aligned} \mathsf{P}_{X_iY_i|D_i=1}(x, y) &= \mathsf{P}_{G_i|D_i=1}(y)\cdot \mathsf{P}_{X_iY_i|D_i=1,G_i=y}(x, y) \\ &= (1-\zeta)^{2/3}p(y)\cdot(1-\zeta)^{1/3}p(x|y) \\ &= (1-\zeta)\cdot p(x, y) = q(x, y). \end{aligned}$$

On the other hand, Y_i can take value y^* when $G_i = y^*$ or when $G_i = y$ for any $y \in \mathcal{Y}$. Hence,

$$\begin{aligned} \mathsf{P}_{X_iY_i|D_i=1}(x, y^*) &= \mathsf{P}_{G_i|D_i=1}(y^*)\cdot \mathsf{P}_{X_iY_i|D_i=1,G_i=y^*}(x, y^*) \\ &\quad + \sum_{y\in\mathcal{Y}} \mathsf{P}_{G_i|D_i=1}(y)\cdot \mathsf{P}_{X_iY_i|D_i=1,G_i=y}(x, y^*) \\ &= \left(1-(1-\zeta)^{2/3}\right)\cdot p(x) + (1-\zeta)^{2/3}\left(1-(1-\zeta)^{1/3}\right)\sum_{y\in\mathcal{Y}} p(y)\cdot p(x|y) \\ &= \left(1-(1-\zeta)^{2/3}\right)\cdot p(x) + \left((1-\zeta)^{2/3}-(1-\zeta)\right)\cdot p(x) \\ &= \zeta\cdot p(x) = q(x, y^*). \end{aligned}$$ ◀

In particular the lemma means $\mathsf{P}_{X_iY_i}(x, y^*) = \mathsf{P}_{X_i}(x)\mathsf{P}_{Y_i}(y^*)$. We also note

$$\mathsf{P}_{Y_iG_i|D_i=1}(Y_i \neq G_i) = (1-\zeta)^{2/3}(1-(1-\zeta)^{1/3}) \leq 1 - 2\zeta/3 - 1 + \zeta = \zeta/3. \tag{5}$$

Let $\mathcal{P}$ be any quantum one-way protocol between Alice and Bob, for $\tilde{f}^k \subseteq \mathcal{X}^k \times (\mathcal{Y} \cup \{y^*\})^k \times \mathcal{Z}^k$, which has communication cost ck. $\mathcal{P}$ is depicted in Figure 1. Alice and Bob's inputs are in registers $X = X_1 \dots X_k$ and $Y = Y_1 \dots Y_k$, and they share an entangled pure state uncorrelated with the inputs on registers $E^{\mathrm{A}}E^{\mathrm{B}}$, with Alice holding E^{A} and Bob holding E^{B}. Alice applies a unitary V^{A} on XE^{A}, to get the message register M, and the register A to be discarded. We shall use $|\theta\rangle_{AME^{\mathrm{B}}|x}$ to refer to the pure state in AME^{B} in the protocol after Alice's unitary, for inputs xy ($|\theta\rangle_x$ only depends on y via x). When Alice and Bob's inputs are distributed according to P_{XY}, the state of the protocol after Alice's message, will be given by the following CQ state:

$$\theta_{XYAME^{\mathrm{B}}} = \sum_{xy} \mathsf{P}_{XY}(xy)|xy\rangle\langle xy|_{XY} \otimes |\theta\rangle\langle\theta|_{AME^{\mathrm{B}}|x}.$$

We shall also consider the following purification of it, with the purifying registers $\tilde{X}$ and $\tilde{Y}$:

$$|\theta\rangle_{X\tilde{X}Y\tilde{Y}AME^{\mathrm{B}}} = \sum_{xy}\sqrt{\mathsf{P}_{XY}(xy)}|xxyy\rangle_{X\tilde{X}Y\tilde{Y}}|\theta\rangle_{AME^{\mathrm{B}}|x}.$$

After receiving Alice's message, Bob applies a unitary V^{B} to YME^{B}, after which ME^{B} gets converted to BZ, where $Z = Z_1 \dots Z_k$ are the answer registers. We shall use $|\rho\rangle_{X\tilde{X}Y\tilde{Y}ABZ}$ to refer to $|\theta\rangle_{X\tilde{X}Y\tilde{Y}AME^{\mathrm{B}}}$ after V^{B}. We shall use P_{XYDGZ} to refer to the joint distribution where $XYDG$ are as previously defined; Z is independent of DG given XY, and the conditional distribution of Z given XY is what is obtained by measuring the Z register in the computational basis in $|\rho\rangle$.

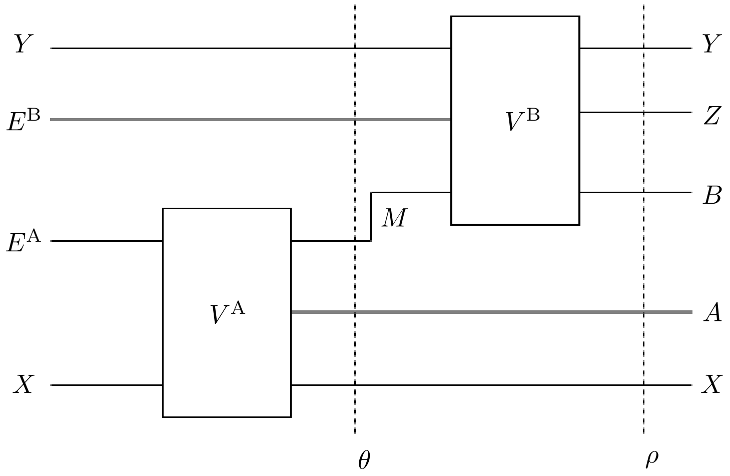


Figure 1 One-way quantum protocol $\mathcal{P}$.

3.2 Proof of Theorem 1

We shall show that if the communication cost ck of $\mathcal{P}$ is $< \frac{\zeta^5 k}{300} \cdot \mathrm{Q}^1_{p,\varepsilon+12\zeta}(\tilde{f}) - k \log\log(24/5\zeta)$, then the success probability of $\mathcal{P}$ is $(1-\varepsilon)^{\Omega(\zeta^6 k/\log|\mathcal{Z}|)}$. This is implied by the following claim, which the rest of the proof will show.

▶ **Lemma 31.** *Let $\delta = \frac{\zeta^6}{1440000}$ and $\delta' = \frac{\zeta^6}{1440000 \log|\mathcal{Z}|}$. For $i \in [k]$, let T_i be the random variable which takes value 1 if $\mathcal{P}$ computes $f(X_i, Y_i)$ correctly, and value 0 otherwise. If the communication cost of $\mathcal{P}$ is $< \frac{\zeta^5 k}{60} \cdot \mathrm{Q}^1_{p,\varepsilon+12\zeta}(\tilde{f}) - k \log\log(24/5\zeta)$, then there exist $\lfloor \delta' k \rfloor$ coordinates $\{i_1, \dots, i_{\lfloor \delta' k \rfloor}\} \subseteq [k]$, such that for all $1 \le r \le \lfloor \delta' k \rfloor - 1$, at least one of the following two conditions holds*

(i) $\Pr\left[\prod_{j=1}^r T_{i_j} = 1\right] \le (1-\varepsilon)^{\delta k}$

(ii) $\Pr\left[T_{i_{r+1}} = 1 \middle| \prod_{j=1}^r T_{i_j} = 1\right] \le 1 - \varepsilon.$

Lemma 31 can be proved inductively. Suppose we have already identified $1 \le t \le \lfloor \delta' k \rfloor$ coordinates in $C = \{i_1, \dots i_t\}$, such that for all $1 \le r \le t-1$, $\Pr\left[T_{i_{r+1}} = 1 | \prod_{j=1}^r T_{i_j} = 1\right] \le 1 - \varepsilon$. Let $\mathcal{E}$ refer to the event $\prod_{i \in C} T_i = 1$. If $\Pr[\mathcal{E}] \le (1-\varepsilon)^{\delta k}$, then we are already done. If not, then we shall show how to identify the $(t+1)$-th coordinate i such that $\Pr[T_i = 1 | \mathcal{E}] \le 1 - \varepsilon$. The process of identifying the first coordinate is also similar, except in that case the conditioning event is empty. Since we only use the lower bound $(1-\varepsilon)^{\delta k}$ on the probability of the conditioning event in our proof, the proof goes through for that case as well.

We shall use the state $|\varphi\rangle$, which is $|\rho\rangle_{X\tilde{X}Y\tilde{Y}ABZ}$ conditioned on $\mathcal{E}$, for the proof of Lemma 31. For any value $DG = dg$, $|\varphi\rangle_{X\tilde{X}Y\tilde{Y}ABZ|dg}$ is defined as:

$$|\varphi\rangle_{X\tilde{X}Y\tilde{Y}ABZ|dg} = \frac{1}{\sqrt{\gamma_{dg}}} \sum_{xy} \sqrt{\mathsf{P}_{XY|dg}(xy)} |xxyy\rangle_{X\tilde{X}Y\tilde{Y}} \otimes \sum_{z_C:(x_C,y_C,z_C)\in \tilde{f}^t} |z_C\rangle_{Z_C} |\tilde{\varphi}\rangle_{ABZ_{\bar{C}}|xyz_C}.$$

Here $|\tilde{\varphi}\rangle_{xyz_C}$ is a subnormalized state with $\| |\tilde{\varphi}\rangle_{ABZ_{\bar{C}}|xyz_C} \|_2^2 = \mathsf{P}_{Z_C|xy}(z_C)$. The overall normalization factor γ_{dg} is the probability of $\mathcal{E}$ conditioned on dg, and satisfies

$$\sum_{dg} \mathsf{P}_{DG}(dg) \cdot \gamma_{dg} = \Pr[\mathcal{E}].$$

It is clear that the distribution of XYZ in $|\varphi\rangle_{X\tilde{X}Y\tilde{Y}ABZ|dg}$ is $\mathsf{P}_{XYZ|\mathcal{E},dg}$. Note that we are using the notation $|\varphi\rangle_{dg}$ without explicitly considering registers DG on which a measurement is done to obtain $|\varphi\rangle_{dg}$. We shall also sometimes use $|\varphi\rangle_{d_{-i}g_{-i}}$ in which the xy distributions are conditioned on $d_{-i}g_{-i}$ instead, which changes the normalization factor to some $\gamma_{d_{-i}g_{-i}}$, everything else remaining the same. $\varphi_{x_iy_id_{-i}g_{-i}}$ refers as usual to the state obtained when a measurement done on the X_iY_i registers (which are actually present in $|\varphi\rangle$) in $|\varphi\rangle_{d_{-i}g_{-i}}$. For $i \notin \bar{C}$, we shall use the states $|\varphi\rangle_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}ABZ_{\bar{C}}|x_iy_ix_Cy_Cz_Cd_{-i}g_{-i}}$ in our proof, which we note are pure states.

Lemma 31 will be proved with the help of the following lemma, whose proof we give later.

▶ **Lemma 32.** *If* $\Pr[\mathcal{E}] \geq (1-\varepsilon)^{\delta k}$*, then there exist a coordinate* $i \in \bar{C}$*, a random variable* $R_i = X_CY_CZ_CD_{-i}G_{-i}$ *and for each* $R_i = r_i$ *a state* $|\varphi'\rangle_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}ABZ_{\bar{C}}|y^*r_i}$ *such that the following conditions hold:*

(i) $\|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},X_i}\|_1 \leq \frac{7\zeta}{120}$

(ii) $\|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},Y_i}\|_1 \leq \frac{7\zeta}{120}$.

(iii) *There exist projectors* $\{\Pi_{x_ir_i}\}_{x_ir_i}$ *acting only on registers* $X_{\bar{C}}\tilde{X}_{\bar{C}}A$ *and unitaries* $\{U_{y_ir_i}\}_{y_ir_i}$ *acting only on* $Y_{\bar{C}}\tilde{Y}_{\bar{C}}BZ_{\bar{C}}$*, such that each* $\Pi_{x_ir_i}$ *succeeds on* $|\varphi'\rangle_{r_i}$ *with probability* $\alpha = 2^{-c'}$ *where* $c' \leq \frac{60c}{\zeta^5}$*, and*

$$\mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \left\| \frac{1}{\alpha}(\Pi_{x_ir_i} \otimes U_{y_ir_i})|\varphi'\rangle\langle\varphi'|_{y^*r_i}(\Pi_{x_ir_i} \otimes U^\dagger_{y_ir_i}) - |\varphi\rangle\langle\varphi|_{x_iy_ir_i} \right\|_1 \leq 21\zeta.$$

Proof of Lemma 31. We give a one-way quantum protocol $\mathcal{P}'$ for $\tilde{f}$, whose inputs are distributed according to $\mathsf{P}_{X_iY_i}$, i.e., q, by embedding Alice and Bob's inputs into the i-th coordinate of $|\varphi\rangle_{x_iy_ir_i}$, as follows:

- Alice and Bob have r according to the distribution required by Fact 12 as shared randomness, and $2^{60c/\zeta^5}\log(24/5\zeta)$ copies of $|\varphi'\rangle_{y^*r_i}$ as shared entanglement, with Alice holding registers $X_{\bar{C}}\tilde{X}_{\bar{C}}A$ and Bob holding registers $Y_{\bar{C}}\tilde{Y}_{\bar{C}}BZ_{\bar{C}}$ of each copy.
- On input (x_i, y_i) from $\mathsf{P}_{X_iY_i}$, using items (i), (ii) of Lemma 32, their shared randomness, and the protocol from Fact 12, Alice and Bob generate random variables $R_i^{\mathrm{A}}R_i^{\mathrm{B}}$ such that

  $$\|\mathsf{P}_{X_iY_iR_i^{\mathrm{A}}R_i^{\mathrm{B}}} - \mathsf{P}_{X_iY_iR_iR_i|\mathcal{E}}\|_1 \leq \frac{7\zeta}{24}.$$

  where R_iR_i denotes two perfectly correlated copies of R_i in $\mathsf{P}_{X_iY_iR_iR_i|\mathcal{E}}$.
- Alice applies the $\{\Pi_{x_ir_i^{\mathrm{A}}}, \mathbb{1} - \Pi_{x_ir_i^{\mathrm{A}}}\}$ measurement according to her input and R_i^{A} on her registers for each copy of the shared entangled state. If the $\Pi_{x_ir_i^{\mathrm{A}}}$ measurement does not succeed on any copy, then she aborts. Otherwise, she sends to Bob a $(\frac{60c}{\zeta^5} + \log\log(24/5\zeta))$-bit message indicating an index where $\Pi_{x_ir_i^{\mathrm{A}}}$ measurement succeeded.

- Bob applies the unitary $U_{y_i r_i^{\mathrm{B}}}$ according to his input and R_i^{B} on the copy of the shared entangled state whose index Alice has sent, and measures the Z_i register of the resulting state to give his output.

To analyze the success of this protocol, first note that

$$\mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \Pr[\text{Result of } Z_i \text{ measurement on } |\varphi\rangle_{x_iy_ir_i} \in \tilde{f}(x_i,y_i)] = \Pr[T_i = 1|\mathcal{E}].$$

Let us first assume Alice and Bob have $(x_i, y_i, r_i^{\mathrm{A}}, r_i^{\mathrm{B}})$ distributed exactly according to $\mathsf{P}_{X_iY_iR_iR_i|\mathcal{E}}$ – we shall denote both r_i^{A} and r_i^{B} by r_i in this case. Alice aborts the protocol if none of her measurements succeed. This happens with probability

$$(1-2^{-c'})^{2^{60c/\zeta^5}\cdot\log(24/5\zeta)} \leq \frac{5\zeta}{24}.$$

If Alice does not abort, then Alice and Bob's state after Bob's unitary is $\frac{1}{\sqrt{\alpha}}\Pi_{x_ir_i} \otimes U_{y_ir_i}|\varphi'\rangle_{y^*r_i}$. From (iii), the expected probability of the Z_i measurement on this state giving an answer $\in \tilde{f}(x_i,y_i)$ is at least $\Pr[T_i = 1|\mathcal{E}] - \frac{21\zeta}{2}$. Hence, if Alice and Bob had $(x_i, y_i, r_i^{\mathrm{A}}, r_i^{\mathrm{B}})$ distributed according to $\mathsf{P}_{X_iY_iR_iR_i|\mathcal{E}}$, then their expected success probability would have been at least $\Pr[T_i = 1|\mathcal{E}] - \frac{21\zeta}{2} - \frac{5\zeta}{24}$. Since Alice and Bob have $(x_i, y_i, r_i^{\mathrm{A}}, r_i^{\mathrm{B}})$ according to $\mathsf{P}_{X_iY_iR_i^{\mathrm{A}}R_i^{\mathrm{B}}}$ instead, their expected success probability is at least

$$\Pr[T_i = 1|\mathcal{E}] - \frac{21\zeta}{2} - \frac{5\zeta}{24} - \frac{7\zeta}{24} \geq \Pr[T_i = 1|\mathcal{E}] - 11\zeta.$$

Since $\|q(x,y) - p(x,y)\|_1 \leq 2\zeta$, when the same protocol is run on X_iY_i distributed according to p instead, it must succeed with probability at least $\Pr[T_i = 1|\mathcal{E}] - 12\zeta$. Since the communication in $\mathcal{P}'$ is at most $(\frac{60c}{\zeta^5} + \log\log(24/5\zeta)) < \mathrm{Q}^1_{p,\varepsilon+12\zeta}(\tilde{f})$, $\Pr[T_i = 1|\mathcal{E}] \geq 1-\varepsilon$ gives the error probability of $\mathcal{P}'$ to be $\leq \varepsilon + 12\zeta$, which is a contradiction. Hence we must have $\Pr[T_i = 1|\mathcal{E}] \leq 1 - \varepsilon$. The desired result thus follows by setting $i_{t+1} = i$. ◀

3.3 Proof of Lemma 32

First we shall show that on expectation over $i \in \bar{C}$, a number of probability distributions conditioned on $\mathcal{E}$ are close to those unconditioned on $\mathcal{E}$. Applying Fact 10 with T and V being trivial and $U_i = X_iY_iD_iG_i$ for $i \in \bar{C}$, we get,

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iD_iG_i|\mathcal{E}} - \mathsf{P}_{X_iY_iD_iG_i}\|_1 \leq \frac{1}{k-t}\sqrt{k\cdot\log((1-\varepsilon)^{-\delta k})} \leq \sqrt{2\delta}. \tag{6}$$

In particular, due to (5), this means

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \mathsf{P}_{Y_iG_i|\mathcal{E},D_i=1}(Y_i = G_i) \geq 1 - \zeta/3 - \sqrt{2\delta}. \tag{7}$$

And since $\mathsf{P}_{G_i|D_i=1}(y^*) = 1-(1-\zeta)^{2/3}$, $\mathsf{P}_{Y_i|D_i=1,G_i=y_i}(y_i) = (1-\zeta)^{1/3}$ for $y_i \in \mathcal{Y}$, we have

$$\zeta+\sqrt{2\delta} \geq 1-(1-\zeta)^{2/3}+\sqrt{2\delta} \geq \mathop{\mathbb{E}}_{i\in\bar{C}} \mathsf{P}_{G_i|\mathcal{E},D_i=1}(y^*) \geq 1-(1-\zeta)^{2/3}-\sqrt{2\delta} \geq 2\zeta/3-\sqrt{2\delta} \tag{8}$$

$$\left(1-\frac{\zeta}{3}+\sqrt{2\delta}\right) \mathop{\mathbb{E}}_{i\in\bar{C}} \mathsf{P}_{G_i|\mathcal{E},D_i=1}(y_i) \geq \mathop{\mathbb{E}}_{i\in\bar{C}} \mathsf{P}_{Y_iG_i|\mathcal{E},D_i=1}(y_i,y_i) \geq (1-\zeta-\sqrt{2\delta}) \mathop{\mathbb{E}}_{i\in\bar{C}} \mathsf{P}_{G_i|\mathcal{E},D_i=1}(y_i). \tag{9}$$

Fact 10 can again be applied with $U_i = X_iY_i$, $T = X_CY_CDG$ and $V = Z_C$. Let $\delta_1 = \delta + \delta'\log|\mathcal{Z}| = \frac{\zeta^6}{720000}$. Then we have,

$$\sqrt{2\delta_1} \geq \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iX_CY_CZ_CDG|\mathcal{E}} - \mathsf{P}_{X_CY_CZ_CDG|\mathcal{E}}\mathsf{P}_{X_iY_i|X_CY_CDG}\|_1$$

$$\begin{aligned} &= \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iX_CY_CZ_CDG|\mathcal{E}} - \mathsf{P}_{X_CY_CZ_CDG|\mathcal{E}}\mathsf{P}_{X_iY_i|D_iG_i}\|_1 \\ &= \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iD_iG_iR_i|\mathcal{E}} - \mathsf{P}_{D_iG_iR_i|\mathcal{E}}\mathsf{P}_{X_iY_i|D_iG_i}\|_1. \qquad (10) \end{aligned}$$

We note that D_i takes value uniformly in $\{0,1\}$ even conditioned on $\mathcal{E}$. Hence from (10),

$$\begin{aligned} \sqrt{2\delta_1} &\geq \frac{1}{2} \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iG_iR_i|\mathcal{E},D_i=0} - \mathsf{P}_{G_iR_i|\mathcal{E},D_i=0}\mathsf{P}_{X_iY_i|G_i,D_i=0}\|_1 \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iR_i|\mathcal{E}}\mathsf{P}_{Y_i|X_i}\|_1 \end{aligned}$$

where we have used the fact that $X_i = G_i$ conditioned on $D_i = 0$. Combining this with the fact that $\mathbb{E}_{i\in\bar{C}} \|\mathsf{P}_{X_i|\mathcal{E}} - \mathsf{P}_{X_i}\|_1 \leq \sqrt{2\delta}$, we have,

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},X_i}\|_1 \leq 3\sqrt{2\delta_1} < \frac{7\zeta^3}{600}. \qquad (11)$$

Due to Corollary 9 we also have from (11),

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iR_i|\mathcal{E},y^*} - \mathsf{P}_{X_iR_i|\mathcal{E}}\|_1 \leq \frac{33\sqrt{2\delta_1}}{\zeta}. \qquad (12)$$

Let $\mathcal{F}_i$ denote the event $Y_i = G_i$. We know $\mathbb{E}_{i\in\bar{C}} \mathsf{P}_{X_iY_iG_i|D_i=1}(\mathcal{F}_i) \geq 1 - \zeta/3 - \sqrt{2\delta}$, from (7). Hence, using Fact 7,

$$\begin{aligned} &\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{Y_iR_i|\mathcal{E}}\mathsf{P}_{X_i|Y_i}\|_1 \\ &= \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iG_iR_i|\mathcal{E},D_i=1,\mathcal{F}_i} - \mathsf{P}_{G_iR_i|\mathcal{E},D_i=1,\mathcal{F}_i}\mathsf{P}_{X_iY_i|G_iD_i=1,\mathcal{F}_i}\|_1 \\ &\leq 6 \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iD_iR_i|\mathcal{E},D_i=1} - \mathsf{P}_{G_iR_i|\mathcal{E},D_i=1}\mathsf{P}_{X_iY_i|G_iD_i=1}\|_1 \leq 6\sqrt{2\delta_1}. \end{aligned}$$

Using $\mathbb{E}_{i\in\bar{C}} \|\mathsf{P}_{Y_i|\mathcal{E}} - \mathsf{P}_{Y_i}\|_1 \leq \sqrt{2\delta}$, we have as before,

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},Y_i}\|_1 \leq 7\sqrt{2\delta_1} = \frac{7\zeta^3}{600}. \qquad (13)$$

Next we shall show the existence of projectors $\Pi_{x_ir_i}$ which take $|\varphi'\rangle_{y^*r_i}$ (which will be defined soon) close to $|\varphi\rangle_{x_iy^*r_i}$. Since M is ck qubits, by Fact 21, for any value $DG = dg$, there exists some state $\sigma_{M|dg}$ such that

$$\mathsf{S}_\infty(\theta_{XY\tilde{Y}E^{\mathrm{B}}M|dg}\|\theta_{XY\tilde{Y}E^{\mathrm{B}}|dg} \otimes \sigma_{M|dg}) \leq 2ck.$$

By Fact 20 we have,

$$\mathsf{S}_\infty\left(\rho_{XY\tilde{Y}BZ|dg}\|V^{\mathrm{B}}(\theta_{XY\tilde{Y}E^{\mathrm{B}}|dg} \otimes \sigma_{M|dg})(V^{\mathrm{B}})^\dagger\right) \leq 2ck.$$

Let $\psi_{X_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}BZ_{\bar{C}}|dg} = \mathrm{Tr}_{Z_C}(V^{\mathrm{B}}(\theta_{XYE^{\mathrm{B}}|dg} \otimes \sigma_{M|x_Cy_Cdg})(V^{\mathrm{B}})^\dagger)$. Note that $\theta_{XY\tilde{Y}E^{\mathrm{B}}|dg} \otimes \sigma_{M|dg}$ is product across X and the other registers, and V^{B} does not act on X. Hence $\psi_{X_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}BZ_{\bar{C}}|dg}$ is also product across X and the other registers, and moreover, all the X_i-s are in product with each other as well. We have,

$$\mathsf{S}_\infty\left(\rho_{XY\tilde{Y}BZ_{\bar{C}}|dg}\|\psi_{XY\tilde{Y}BZ_{\bar{C}}|dg}\right) \leq 2ck.$$

Using Facts 22 and 19, this gives us

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathsf{P}_{X_C Y_C Z_C DG|\mathcal{E}}} \left[\mathsf{S}\left(\varphi_{X_{\bar{C}} Y_{\bar{C}} \tilde{Y}_{\bar{C}} B Z_{\bar{C}} | x_C y_C z_C dg} \| \psi_{X_{\bar{C}} Y_{\bar{C}} \tilde{Y}_{\bar{C}} B Z_{\bar{C}} | x_C y_C dg}\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\mathsf{P}_{Z_C DG|\mathcal{E}}} \left[\mathsf{S}\left(\varphi_{XY\tilde{Y}BZ_{\bar{C}}|z_C dg} \| \psi_{XY\tilde{Y}BZ_{\bar{C}}|dg}\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\mathsf{P}_{Z_C DG|\mathcal{E}}} \left[\mathsf{S}_\infty\left(\varphi_{XY\tilde{Y}BZ_{\bar{C}}|z_C dg} \| \psi_{XY\tilde{Y}BZ_{\bar{C}}|dg}\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\mathsf{P}_{Z_C DG|\mathcal{E}}} \left[\mathsf{S}_\infty\left(\varphi_{XY\tilde{Y}BZ_{\bar{C}}|z_C dg} \| \varphi_{XY\tilde{Y}BZ_{\bar{C}}|dg}\right)\right. \\
&\qquad \left. + \mathsf{S}_\infty\left(\varphi_{XY\tilde{Y}BZ_{\bar{C}}|dg} \| \rho_{XY\tilde{Y}BZ_{\bar{C}}|dg}\right) + \mathsf{S}_\infty\left(\rho_{XY\tilde{Y}BZ_{\bar{C}}|dg} \| \psi_{XY\tilde{Y}BZ_{\bar{C}}|dg}\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\mathsf{P}_{Z_C DG|\mathcal{E}}} \left[\log(1/\mathsf{P}_{Z_C|\mathcal{E}}(z_C)) + \log(1/\Pr[\mathcal{E}]) + 2ck\right] \\
&\leq |C| \log |\mathcal{Z}| + \delta k + 2ck \leq (\delta_1 + 2c)k.
\end{aligned}$$

By Quantum Raz's Lemma,

$$\begin{aligned}
4c + 2\delta_1 &\geq \mathop{\mathbb{E}}_{i \in \bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_C Y_C Z_C DG|\mathcal{E}}} \mathsf{I}(X_i : Y_{\bar{C}} \tilde{Y}_{\bar{C}} B Z_{\bar{C}})_{\varphi_{x_C y_C z_C dg}} \\
&= \mathop{\mathbb{E}}_{i \in \bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{D_i G_i R_i|\mathcal{E}}} \mathsf{I}(X_i : Y_{\bar{C}} \tilde{Y}_{\bar{C}} B Z_{\bar{C}})_{\varphi_{d_i g_i r_i}} \\
&\geq \mathop{\mathbb{E}}_{i \in \bar{C}} \frac{1}{2} \mathsf{P}_{G_i|\mathcal{E}, D_i=1}(y^*) \mathop{\mathbb{E}}_{\mathsf{P}_{R_i|\mathcal{E}, D_i=1, G_i=y^*}} \mathsf{I}(X_i : Y_{\bar{C}} \tilde{Y}_{\bar{C}} B Z_{\bar{C}})_{\varphi_{r_i|D_i=1, G_i=y^*}} \\
&\geq \mathop{\mathbb{E}}_{i \in \bar{C}} \frac{1}{2} (2\zeta/3 - \sqrt{2\delta}) \mathop{\mathbb{E}}_{\mathsf{P}_{R_i|\mathcal{E}, D_i=1, G_i=y^*}} \mathsf{I}(X_i : Y_{\bar{C}} \tilde{Y}_{\bar{C}} B Z_{\bar{C}})_{\varphi_{r_i, D_i=1, G_i=y^*}} \qquad (14)
\end{aligned}$$

where we have used (8) in the last inequality.

Note that $\varphi_{X_{\bar{C}} \tilde{X}_{\bar{C}} Y_{\bar{C}} \tilde{Y}_{\bar{C}} ABZ_{\bar{C}}|x_i r_i, D_i=1, G_i=y^*}$ is the same state as $\varphi_{X_{\bar{C}} \tilde{X}_{\bar{C}} Y_{\bar{C}} \tilde{Y}_{\bar{C}} ABZ_{\bar{C}}|x_i y^* r_i}$, where the value of Y_i is being conditioned on, instead of G_i. $|\varphi\rangle_{r_i, D_i=1, G_i=y^*}$ is the superposition over X_i of $|\varphi\rangle_{x_i r_i, D_i=1, G_i=y^*}$, with the X_i distribution being $\mathsf{P}_{X_i|\mathcal{E}, r_i, D_i=1, G_i=y^*}$. The only difference between $|\varphi\rangle_{y^* r_i}$ and $|\varphi\rangle_{r_i, D_i=1, G_i=y^*}$ is the X_i distribution, which in the former is $\mathsf{P}_{X_i|\mathcal{E}, y^* r_i}$ instead. We shall refer to $|\varphi\rangle_{r_i, D_i=1, G_i=y^*}$ as simply $|\varphi\rangle_{r_i, 1, y^*}$ as now on – note that there is no ambiguity between this and $|\varphi\rangle_{y^* r_i}$. The same goes for the distributions $\mathsf{P}_{X_i R_i|\mathcal{E}, 1, y^*}$ and $\mathsf{P}_{X_i R_i|\mathcal{E}, y^*}$.

$\mathsf{P}_{X_i|1, y^*}$ is the same distribution as $\mathsf{P}_{X_i|y^*}$ and $\mathsf{P}_{R_i|\mathcal{E}, x_i, 1, y^*}$ is the same distribution as $\mathsf{P}_{R_i|\mathcal{E}, x_i y^*}$ for any x_i. Hence,

$$\begin{aligned}
\mathop{\mathbb{E}}_{i \in \bar{C}} \|\mathsf{P}_{X_i R_i|\mathcal{E}, y^*} - \mathsf{P}_{X_i R_i|\mathcal{E}, 1, y^*}\|_1 &\leq \mathop{\mathbb{E}}_{i \in \bar{C}} \left[\|\mathsf{P}_{X_i R_i|\mathcal{E}, y^*} - \mathsf{P}_{X_i|y^*} \mathsf{P}_{R_i|\mathcal{E}, X_i, y^*}\|_1\right. \\
&\qquad \left. + \|(\mathsf{P}_{X_i|1, y^*} - \mathsf{P}_{X_i|\mathcal{E}, 1, y^*}) \mathsf{P}_{R_i|\mathcal{E}, X_i, y^*}\|_1\right] \\
&\leq \mathop{\mathbb{E}}_{i \in \bar{C}} \left[\frac{\|\mathsf{P}_{X_i R_i|\mathcal{E}} - \mathsf{P}_{X_i} \mathsf{P}_{R_i|\mathcal{E}, X_i}\|_1}{2\zeta/3 - \sqrt{2\delta}} + \frac{\|\mathsf{P}_{X_i|\mathcal{E}} - \mathsf{P}_{X_i}\|_1}{2\zeta/3 - \sqrt{2\delta}}\right] \\
&\leq \frac{7\sqrt{2\delta_1}}{\zeta}
\end{aligned}$$

where we have used (8) in the second inequality. Using the above computation and (12), we get,

$$\mathop{\mathbb{E}}_{i \in \bar{C}} \|\mathsf{P}_{X_i R_i|\mathcal{E}} - \mathsf{P}_{X_i R_i|\mathcal{E}, 1, y^*}\|_1 \leq \frac{40\sqrt{2\delta_1}}{\zeta}.$$

Let

$$|\varphi'\rangle_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}ABZ_{\bar{C}}|y^*r_i} = \sum_{x_i}\sqrt{\mathsf{P}_{X_i|\mathcal{E},r_i}}|\varphi\rangle_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}ABZ_{\bar{C}}|x_iy^*r_i},$$

i.e., $|\varphi'\rangle_{y^*r_i}$ is the same state as $|\varphi\rangle_{y^*r_i}$ except that the distribution of distribution of X_i is unconditioned on $Y_i = y^*$. From (14) and Fact 23, we then have that,

$$\mathop{\mathbb{E}}_{i\in\bar{C}}\mathsf{P}_{X_iR_i|\mathcal{E}}\left(\mathsf{S}_\infty^\zeta\left(\varphi'_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}BZ_{\bar{C}}|x_iy^*r_i}\middle\|\varphi'_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}BZ_{\bar{C}}|y^*r_i}\right) > \frac{28(2c+\delta_1)+1}{\zeta^4}\right) \leq \zeta + \frac{20\sqrt{2\delta_1}}{\zeta}.$$

Hence by Fact 25, there exist projectors $\Pi_{x_ir_i}$ acting on registers $X_{\bar{C}}\tilde{X}_{\bar{C}}A$, such that $\Pi_{x_ir_i}$ succeeds with probability $\alpha = 2^{-c'}$ on $|\varphi'\rangle_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}ABZ_{\bar{C}}|y^*r_i}$, where $c' = \frac{60c}{\zeta^5}$, and

$$\begin{aligned}\mathop{\mathbb{E}}_{\in\bar{C}}\mathop{\mathbb{E}}_{\mathsf{P}_{X_iR_i|\mathcal{E}}}\left\|\frac{1}{\alpha}(\Pi_{x_ir_i}\otimes\mathbb{1})|\varphi'\rangle\langle\varphi'|_{y^*r_i}(\Pi_{x_ir_i}\otimes\mathbb{1}) - |\varphi\rangle\langle\varphi|_{x_iy^*r_i}\right\|_1 &\leq 3\zeta + \frac{40\sqrt{2\delta_1}}{\zeta^2}\\ &\leq \frac{7\zeta}{2}.\end{aligned} \tag{15}$$

Next we shall show the existence of unitaries $U_{y_ir_i}$ taking $|\varphi\rangle_{y^*r_i,D_i=1,G_i=y_i}$ close to $|\varphi\rangle_{y_ir_i,D_i=1,G_i=y_i}$. By similar arguments as the ones leading to (14) on Bob's side (except the first step where we consider the information due to Alice's message, which does not apply here), we can alo upper bound $\mathbb{E}_{\mathsf{P}_{X_CY_CZ_CDG|\mathcal{E}}}\left[\mathsf{S}\left(\varphi_{Y_{\bar{C}}X_{\bar{C}}\tilde{X}_{\bar{C}}A|x_Cy_Cz_Cdg}\middle\|\rho_{Y_{\bar{C}}X_{\bar{C}}\tilde{X}_{\bar{C}}A|x_Cy_Cdg}\right)\right]$. Hence by Raz's lemma again,

$$\begin{aligned}2\delta_1 &\geq \mathop{\mathbb{E}}_{i\in\bar{C}}\mathop{\mathbb{E}}_{\mathsf{P}_{D_iG_iR_i|\mathcal{E}}}\mathsf{I}(Y_i : X_{\bar{C}}\tilde{X}_{\bar{C}}A)_{\varphi_{d_ig_ir_i}}\\
&\geq \mathop{\mathbb{E}}_{i\in\bar{C}}\frac{1}{2}(1-\zeta-\sqrt{2\delta})\mathop{\mathbb{E}}_{\mathsf{P}_{R_iG_i|\mathcal{E},D_i=1,G_i\neq y^*}}\mathsf{I}(Y_i : X_{\bar{C}}\tilde{X}_{\bar{C}}A)_{\varphi_{r_i,D_i=1,g_i}}\\
&= \mathop{\mathbb{E}}_{i\in\bar{C}}\frac{1}{2}(1-\zeta-\sqrt{2\delta})\mathop{\mathbb{E}}_{\mathsf{P}_{R_iG_iY_i|\mathcal{E},D_i=1,G_i\neq y^*}}\left[\mathsf{S}\left(\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|y_i,D_i=1,g_i}\middle\|\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|D_i=1,g_i}\right)\right]\\
&\geq \mathop{\mathbb{E}}_{i\in\bar{C}}\frac{1}{2}(1-\zeta-\sqrt{2\delta})\sum_{y_i\in\mathcal{Y}}\mathop{\mathbb{E}}_{\mathsf{P}_{R_i|\mathcal{E},D_i=1,G_i=y_i}}\mathsf{P}_{G_i|\mathcal{E},D_i=1}(y_i)\cdot\\
&\quad\Big[(1-\zeta-\sqrt{2\delta})\|\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|y_i,r_i,D_i=1,G_i=y_i} - \varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|r_i,D_i=1,G_i=y_i}\|_1^2\\
&\quad+(\zeta/3-\sqrt{2\delta})\|\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|y^*,r_i,D_i=1,G_i=y_i} - \varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|r_i,D_i=1,G_i=y_i}\|_1^2\Big].\end{aligned}$$

where we have used (9) and Pinsker's inequality in the last line. Since the ℓ_1 norm obeys triangle inequality, we have,

$$\begin{aligned}&\mathop{\mathbb{E}}_{i\in\bar{C}}\sum_{y_i\in\mathcal{Y}}\mathop{\mathbb{E}}_{\mathsf{P}_{R_i|\mathcal{E},1,y_i}}\mathsf{P}_{G_i|\mathcal{E},1}(y_i)\|\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|y_ir_i,1,y_i} - \varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|y^*r_i,1,y_i}\|_1^2\\
&\leq \mathop{\mathbb{E}}_{i\in\bar{C}}\sum_{y_i\in\mathcal{Y}}\mathop{\mathbb{E}}_{\mathsf{P}_{R_i|\mathcal{E},1,y_i}}\mathsf{P}_{G_i|\mathcal{E},1}(y_i)\cdot 2\Big[\|\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|y_i,r_i,1,y_i} - \varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|r_i,1,y_i}\|_1^2\\
&\quad+\|\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|y^*,r_i,1,y_i} - \varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}A|r_i,1,y_i}\|_1^2\Big]\\
&\leq \frac{4\delta_1}{1-\zeta-\sqrt{2\delta}}\left(\frac{1}{1-\zeta-\sqrt{2\delta}} + \frac{1}{\zeta/3-\sqrt{2\delta}}\right)\\
&\leq \frac{32\delta_1}{\zeta}.\end{aligned}$$

We note that $\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}ABZ_{\bar{C}}|y_ir_i,1,y_i}$ and $\varphi_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}ABZ_{\bar{C}}|y^*r_i,1,y_i}$ are pure states. Hence, using the Fuchs-van de Graaf inequality and Uhlmann's theorem, there exist unitaries $U_{y_ir_i}$ acting only on $Y_{\bar{C}}\tilde{Y}_{\bar{C}}BZ_{\bar{C}}$ such that

$$\begin{aligned}
&= \mathop{\mathbb{E}}_{i\in\bar{C}} \sum_{y_i\in\mathcal{Y}} \mathop{\mathbb{E}}_{\mathsf{P}_{R_i|\mathcal{E},1,y_i}} \mathsf{P}_{G_i|\mathcal{E},1}(y_i) \| |\varphi\rangle\langle\varphi|_{y_ir_i,1,y_i} - (\mathbb{1}\otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{y^*r_i,1,y_i}(\mathbb{1}\otimes U^\dagger_{y_ir_i})\|_1 \\
&\leq \left(\frac{32\delta_1}{\zeta}\right)^{1/4} \qquad (16)
\end{aligned}$$

Finally, we need to show that $\Pi_{x_ir_i}\otimes U_{y_ir_i}$ takes $|\varphi'\rangle_{y^*r_i}$ close to $|\varphi\rangle_{x_iy_ir_i}$. To do this, we shall first show that $U_{y_ir_i}$ in fact takes $|\varphi\rangle_{x_iy^*r_i}$ close to $|\varphi\rangle_{x_iy_ir_i}$. Consider the superoperator $\mathcal{O}_{X_i}$ that measures the register X_i and writes it in a different register.

$$\begin{aligned}
\mathcal{O}_{X_i}(|\varphi\rangle\langle\varphi|_{y_ir_i,1,y_i}) &= \sum_{x_i} \mathsf{P}_{X_i|\mathcal{E},y_ir_i,D_i=1,G_i=y_i}(x_i)|x_i\rangle\langle x_i| \otimes |\varphi\rangle\langle\varphi|_{x_iy_ir_i,1,y_i} \\
&= \sum_{x_i} \mathsf{P}_{X_i|\mathcal{E},y_ir_i,D_i=1,G_i=y_i}(x_i)|x_i\rangle\langle x_i| \otimes |\varphi\rangle\langle\varphi|_{x_iy_ir_i} \\
\mathcal{O}_{X_i}(|\varphi\rangle\langle\varphi|_{y^*r_i,1,y_i}) &= \sum_{x_i} \mathsf{P}_{X_i|\mathcal{E},y^*r_i,D_i=1,G_i=y_i}(x_i)|x_i\rangle\langle x_i| \otimes |\varphi\rangle\langle\varphi|_{x_iy^*r_i}
\end{aligned}$$

where we have made the observation that $|\varphi\rangle\langle\varphi|_{x_iy_ir_i,1,y_i}$ and $|\varphi\rangle\langle\varphi|_{x_iy^*r_i,1,y_i}$ are the same states as $|\varphi\rangle\langle\varphi|_{x_iy_ir_i}$ and $|\varphi\rangle\langle\varphi|_{x_iy^*r_i}$. By Fact 10 we can get,

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iG_iR_i|\mathcal{E},1} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|1,G_i}\|_1 \leq 2\sqrt{2\delta_1}.$$

Hence, for any value $Y_i = y_i$,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iG_iR_i|\mathcal{E},1} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|\mathcal{E},y_i,1,G_iR_i}\|_1 \\
&\leq \mathop{\mathbb{E}}_{i\in\bar{C}} \left[\|\mathsf{P}_{X_iG_iR_i|\mathcal{E},1} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|y_i,1,G_i})\|_1 + \|\mathsf{P}_{G_iR_i|\mathcal{E},1}(\mathsf{P}_{X_i|y_i,1,G_i} - \mathsf{P}_{X_i|\mathcal{E},y_i,1,G_iR_i})\|_1\right] \\
&\leq \mathop{\mathbb{E}}_{i\in\bar{C}} \left[\|\mathsf{P}_{X_iG_iR_i|\mathcal{E},1} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|1,G_i}\|_1 + \frac{2}{\frac{\zeta}{3} - \sqrt{2\delta}}\|\mathsf{P}_{X_iG_iR_i|\mathcal{E},1} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|1,G_i}\|_1\right] \\
&\leq \frac{8\sqrt{2\delta_1}}{\zeta}
\end{aligned}$$

where we have used the fact that for any value $G_i = g_i$, we must have $\mathsf{P}_{Y_i|1,g_i}(y_i) \geq \zeta/3 - \sqrt{2\delta}$. Finally,

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iG_iR_i|\mathcal{E},1} - \mathsf{P}_{X_iY_iR_i|\mathcal{E},1}\|_1 \leq 2\mathsf{P}_{Y_iG_i|\mathcal{E},1}(Y_i \neq G_i) \leq \zeta/3 + \sqrt{2\delta}.$$

Observing that $\mathsf{P}_{X_iY_iR_i|\mathcal{E},1}$ is the same as $\mathsf{P}_{X_iY_iR_i|\mathcal{E}}$ we get,

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|\mathcal{E},y_i,1,G_iR_i}\|_1 \leq \frac{8\sqrt{2\delta_1}}{\zeta} + \frac{\zeta}{3} + \sqrt{2\delta}.$$

Using this and (16) we get,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \| |\varphi\rangle\langle\varphi|_{x_iy_ir_i} - (\mathbb{1}\otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{x_iy^*r_i}(\mathbb{1}\otimes U^\dagger_{y_ir_i})\|_1 \\
&\leq \mathop{\mathbb{E}}_{i\in\bar{C}} \left[\|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|\mathcal{E},y_i,1,G_iR_i}\|_1 + \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{G_iR_i|\mathcal{E},1}\mathsf{P}_{X_i|\mathcal{E},y^*,1,G_iR_i}\|_1\right.
\end{aligned}$$

$$
\begin{aligned}
&+ \mathop{\mathbb{E}}_{\mathsf{P}_{G_iR_i|\mathcal{E},1}} \left\| \mathop{\mathbb{E}}_{\mathsf{P}_{X_i|\mathcal{E},y_ir_i,1,y_i}} |x_i\rangle\langle x_i| \otimes |\varphi\rangle\langle\varphi|_{x_iy_ir_i} - \mathop{\mathbb{E}}_{\mathsf{P}_{X_i|\mathcal{E},y^*r_i,1,y_i}} \mathbb{1} \otimes U_{y_ir_i}|\varphi\rangle\langle\varphi|_{x_iy^*r_i}\mathbb{1}\otimes U^\dagger_{y_ir_i}\right\|_1 \Bigg] \\
&= \frac{16\sqrt{2\delta_1}}{\zeta} + \frac{2\zeta}{3} + 2\sqrt{2\delta} + \left(\frac{32\delta_1}{\zeta}\right)^{1/4} < \frac{7\zeta}{10} \qquad (17)
\end{aligned}
$$

where we have bounded the last term in the first inequality by applying Fact 15 on (16) with $\mathcal{O}_{X_i}$. Notice that we have also removed the conditioning $G_i \neq y^*$, since for $G_i = y^*$, the corresponding states are both $|\varphi\rangle_{x_iy^*r_i}$.

From (15) and (17) we get,

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \left\| \frac{1}{\alpha}(\Pi_{x_ir_i}\otimes U_{y_ir_i})|\varphi'\rangle\langle\varphi'|_{y^*r_i}(\Pi_{x_ir_i}\otimes U^\dagger_{y_ir_i}) - |\varphi\rangle\langle\varphi|_{x_iy_ir_i}\right\|_1 \\
&\leq \mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \Bigg[\bigg\| \frac{1}{\alpha}(\Pi_{x_ir_i}\otimes U_{y_ir_i})|\varphi'\rangle\langle\varphi'|_{y^*r_i}(\Pi_{x_ir_i}\otimes U^\dagger_{y_ir_i}) \\
&\qquad - (\mathbb{1}\otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{x_iy^*r_i}(\mathbb{1}\otimes U^\dagger_{y_ir_i})\bigg\|_1 \\
&\qquad + \left\|(\mathbb{1}\otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{x_iy^*r_i}(\mathbb{1}\otimes U^\dagger_{y_ir_i}) - |\varphi\rangle\langle\varphi|_{x_iy_ir_i}\right\|_1 \Bigg] \\
&= \mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \Bigg[\left\| \frac{1}{\alpha}(\Pi_{x_ir_i}\otimes \mathbb{1})|\varphi'\rangle\langle\varphi'|_{y^*r_i}(\Pi_{x_ir_i}\otimes \mathbb{1}) - |\varphi\rangle\langle\varphi|_{x_iy^*r_i}\right\|_1 \\
&\qquad + \left\|(\mathbb{1}\otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{x_iy^*r_i}(\mathbb{1}\otimes U^\dagger_{y_ir_i}) - |\varphi\rangle\langle\varphi|_{x_iy_ir_i}\right\|_1 \Bigg] \\
&\leq \frac{7\zeta}{2} + \frac{7\zeta}{10} = \frac{21\zeta}{5}. \qquad (18)
\end{aligned}
$$

Using Markov's inequality on (11), (13) and (18), we get an index $i \in \bar{C}$ such that the conditions (i)-(iii) for Lemma 32 hold.

4 Proof of parallel repetition theorem

In this section we prove Theorem 2, whose statement is recalled below.

▶ **Theorem 2.** *For a two-player non-local game $G = (q, \mathcal{X}\times\mathcal{Y}, \mathcal{A}\times\mathcal{B}, \mathsf{V})$ such that q is a distribution anchored on one side with anchoring probability ζ,*

$$
\omega^*(G^k) = \left(1-(1-\omega^*(G))^5\right)^{\Omega\left(\frac{\zeta^2 k}{\log(|\mathcal{A}|\cdot|\mathcal{B}|)}\right)}.
$$

4.1 Setup

The proof of this theorem is very similar to that of the direct product theorem, so we shall only highlight points of difference. Whereas in the communication case, we started with an arbitrary distribution p and defined distribution q anchored on one side close to p, here we start with an already anchored distribution. To preserve similarity with the direct product proof, we shall consider q to be anchored on the $\mathcal{Y}$ side here as well, but the proof goes through analogously for a distribution anchored on the $\mathcal{X}$ side. We define the correlation-breaking variables and the joint distribution P_{XYDG} exactly as before.[2]

[2] The definition of $\mathsf{P}_{X_iY_iD_iG_i}$ in the previous section makes references to $p(x,y)$. Since there is no p in the present case, $p(x,y)$ can simply be replaced by $q(x,y|y\neq y^*)$.

We consider an entangled strategy $\mathcal{S}$ for G^k, where Alice and Bob, with input registers $X = X_1 \dots X_k$ and $Y = Y_1 \dots Y_k$, initially share an entangled state, and perform unitaries V^{A} and V^{A} respectively on their parts of the entangled state and and their input registers. As before, conditioned on any value $DG = dg$, we define the following pure state representing $\mathcal{S}$ after these unitaries:

$$|\theta\rangle_{X\tilde{X}Y\tilde{Y}ABE'^{\mathrm{A}}E'^{\mathrm{B}}|dg} = \sum_{xy} \sqrt{\mathsf{P}_{XY|dg}(xy)}|xxyy\rangle_{X\tilde{X}Y\tilde{Y}} \otimes |\theta\rangle_{ABE^{\mathrm{A}}E^{\mathrm{B}}|xy}$$

where AB are the answer registers which are measured in the computational basis by Alice and Bob to obtain their answers (a, b), and $E'^{\mathrm{A}}E'^{\mathrm{B}}$ are some additional registers which are discarded. We shall use $\mathsf{P}_{XYAB|dg}$ to denote the distribution of $XYAB$ in $|\theta\rangle_{dg}$; P_{XYDGAB} is obtained by averaging over dg.

Let the winning probability of of $\omega^*(G)$ be $1 - 5\varepsilon$ for an appropriate ε. We shall prove the following lemma, which is analogous to the direct product case. It is clear that the lemma implies

$$\omega^*(G^k) \leq (1-\varepsilon)^{\frac{\zeta^2\varepsilon^4 k}{\log(|\mathcal{A}|\cdot|\mathcal{B}|)}} = \left(1 - (1-\omega^*(G))^5\right)^{\Omega\left(\frac{\zeta^2 k}{\log(|\mathcal{A}|\cdot|\mathcal{B}|)}\right)}.$$

▶ **Lemma 33.** *Let $\delta = \frac{\zeta^2\varepsilon^4}{1440000}$ and $\delta' = \frac{\zeta^2\varepsilon^4}{1440000\log(|\mathcal{A}|\cdot|\mathcal{B}|)}$. For $i \in [k]$, let T_i denote the random variable $\mathsf{V}(X_i, Y_i, A_i, B_i)$, where $X_iY_iA_iB_i$ are according to P_{XYAB}. Then there exist $\lfloor\delta' k\rfloor$ coordinates $\{i_1, \dots, i_{\lfloor\delta' k\rfloor}\} \subseteq [k]$, such that for all $1 \leq r \leq \lfloor\delta' k\rfloor - 1$, at least one of the conditions holds*

(i) $\Pr\left[\prod_{j=1}^r T_{i_j} = 1\right] \leq (1-\varepsilon)^{\delta k}$

(ii) $\Pr\left[T_{i_{r+1}} = 1 \middle| \prod_{j=1}^r T_{i_j} = 1\right] \leq 1 - \varepsilon.$

As before, we shall consider that we have identified a set of coordinates $C = \{i_1, \dots, i_t\}$ such that for all $1 \leq r \leq t-1$, $\Pr\left[T_{i_{r+1}} = 1 | \prod_{j=1}^r T_{i_j} = 1\right] \leq 1 - \varepsilon$ and $\Pr[\mathcal{E}] = \Pr\left[\prod_{j=1}^t T_{i_j} = 1\right] \geq (1-\varepsilon)^{\delta k}$, and identify a $(t+1)$-th coordinate i. Let E^{A} and E^{B} to denote $A_{\bar{C}}E'^{\mathrm{A}}$ and $B_{\bar{C}}E'^{\mathrm{B}}$ respectively. We define the following state, which is $|\theta\rangle_{dg}$ conditioned on success in C:

$$\begin{aligned}&|\varphi\rangle_{X\tilde{X}Y\tilde{Y}A_CB_CBE^{\mathrm{A}}E^{\mathrm{B}}|dg} \\ &= \frac{1}{\sqrt{\gamma_{dg}}} \sum_{xy} \sqrt{\mathsf{P}_{XY|dg}(xy)}|xxyy\rangle_{X\tilde{X}Y\tilde{Y}} \otimes \sum_{a_Cb_C:\mathsf{V}^t(x_C,y_C,a_C,b_C)=1} |a_Cb_C\rangle_{A_CB_C}|\tilde{\varphi}\rangle_{E^{\mathrm{A}}E^{\mathrm{B}}|xya_Cb_C}.\end{aligned}$$

Here $|\tilde{\varphi}\rangle_{E^{\mathrm{A}}E^{\mathrm{B}}|xya_Cb_C}$ is a subnormalized state satisfying $\||\tilde{\varphi}\rangle_{E^{\mathrm{A}}E^{\mathrm{B}}|xya_Cb_C}\|_2^2 = \mathsf{P}_{A_CB_C|xy}(a_Cb_C)$.

The following lemma is the analog of Lemma 32, which we shall use to prove Lemma 33.

▶ **Lemma 34.** *If $\Pr[\mathcal{E}] \geq (1-\varepsilon)^{\delta k}$, then there exist a coordinate $i \in \bar{C}$, a random variable $R_i = X_CY_CA_CB_CD_{-i}G_{-i}$, such that the following conditions hold:*

(i) $\|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},X_i}\|_1 \leq \frac{7\varepsilon}{150}$

(ii) $\|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},Y_i}\|_1 \leq \frac{7\varepsilon}{150}$

(iii) *There exist unitaries $\{U_{x_ir_i}\}_{x_ir_i}$ and $\{U_{y_ir_i}\}_{y_ir_i}$ respectively acting only on $X_{\bar{C}}\tilde{X}_{\bar{C}}E^{\mathrm{A}}$ and $Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}$, such that*

$$\mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \left\|(U_{x_ir_i} \otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{y^*r_i}(U^\dagger_{x_ir_i} \otimes U^\dagger_{y_ir_i}) - |\varphi\rangle\langle\varphi|_{x_iy_ir_i}\right\|_1 \leq \frac{36\varepsilon}{5}.$$

It is easy to see how this lemma implies Lemma 33. As in the direct product case, Alice and Bob share $|\varphi\rangle_{y^*r_i}$ as entanglement – though in this case only one copy, as well as classical randomness with which they can produce $R_i^{\mathrm{A}}R_i^{\mathrm{B}}$ satisfying

$$\|\mathsf{P}_{X_iY_iR_i^{\mathrm{A}}R_i^{\mathrm{B}}} - \mathsf{P}_{X_iY_iR_iR_i|\mathcal{E}}\|_1 \leq \frac{7\varepsilon}{30}.$$

Alice and Bob apply $U_{x_ir_i^{\mathrm{A}}}$ and $U_{y_ir_i^{\mathrm{B}}}$ according to their inputs and R_i^{A} and R_i^{B} respectively, on their registers E^{A} and E^{B} of $|\varphi\rangle_{y^*r_i}$. They then measure in the computational basis on the A_iB_i registers of resulting state, to give their outcomes (a_i, b_i). $\Pr[T_i = 1|\mathcal{E}] \geq 1 - \varepsilon$ implies that the resulting strategy for G has success probability $> (1 - 5\varepsilon)$, a contradiction which lets us identify i as the $(t + 1)$-th coordinate.

4.2 Proof of Lemma 34

We can prove

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},X_i}\|_1 \leq \frac{7\varepsilon}{600} \tag{19}$$

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iY_iR_i|\mathcal{E}} - \mathsf{P}_{X_iY_i}\mathsf{P}_{R_i|\mathcal{E},Y_i}\|_1 \leq \frac{7\varepsilon}{600} \tag{20}$$

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \||\varphi\rangle\langle\varphi|_{x_iy_ir_i} - (\mathbb{1}\otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{x_iy^*r_i}(\mathbb{1}\otimes U_{y_ir_i}^\dagger)\|_1 \leq \frac{4\varepsilon}{5} \tag{21}$$

exactly the same way as in the direct product case, except conditioning on z_C is replaced by conditioning on a_Cb_C, which leads to the factor of $\log(|\mathcal{A}|\cdot|\mathcal{B}|)$. The rest of the proof will hence be spent getting Alice's unitaries $U_{x_ir_i}$.

Letting $\delta_1 = \delta + \delta' \log(|\mathcal{A}|\cdot|\mathcal{B}|)$, the following is derived analogously to the direct product case, except for the extra factor in the mutual information bound due to communication:

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{R_i|\mathcal{E},D_i=1,G_i=y^*} \mathsf{I}(X_i : Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}})_{\varphi_{r_i,D_i=1,G_i=y^*}} \leq \frac{10\delta_1}{\zeta} \tag{22}$$

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iR_i|\mathcal{E},y^*} - \mathsf{P}_{X_iR_i|\mathcal{E},1,y^*}\|_1 \leq \frac{7\sqrt{2\delta_1}}{\zeta} \tag{23}$$

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iR_i|\mathcal{E}} - \mathsf{P}_{X_iR_i|\mathcal{E},1,y^*}\|_1 \leq \frac{40\sqrt{2\delta_1}}{\zeta}. \tag{24}$$

From (22), by applying Pinsker's inequality, we get,

$$\mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_iR_i|\mathcal{E},1,y^*}} \|\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|x_ir_i,1,y^*} - \varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|r_i,1,y^*}\|_1 \leq \left(\frac{10\delta_1}{\zeta}\right)^{1/2}$$

Note that $\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|x_ir_i,1,y^*}$ is the same state as $\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|x_iy^*r_i}$. But $\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|r_i,1,y^*}$ is not the same state as $\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|y^*r_i}$, due to the averaging over X_i being done with respect to $\mathsf{P}_{X_i|\mathcal{E},r_i,1,y^*}$ in one, and with respect to $\mathsf{P}_{X_i|\mathcal{E},y^*r_i}$ in the other. However, due to (23) we can say,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{i\in\bar{C}} \mathop{\mathbb{E}}_{\mathsf{P}_{X_iR_i|\mathcal{E},1,y^*}} \|\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|x_iy^*r_i} - \varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|y^*r_i}\|_1 \\
&\leq \left(\frac{10\delta_1}{\zeta}\right)^{1/2} + \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iR_i|\mathcal{E},1,y^*} - \mathsf{P}_{R_i|\mathcal{E},1,y^*}\mathsf{P}_{X_i|\mathcal{E},R_i,y^*}\|_1 \\
&\leq \left(\frac{10\delta_1}{\zeta}\right)^{1/2} + \mathop{\mathbb{E}}_{i\in\bar{C}} \|\mathsf{P}_{X_iR_i|\mathcal{E},y^*} - \mathsf{P}_{X_iR_i|\mathcal{E},1,y^*}\|_1
\end{aligned}$$

$$\leq \frac{2\sqrt{108\delta_1}}{\zeta}.$$

Since $|\varphi\rangle_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{A}}E^{\mathrm{B}}|y^*r_i}$ is a purification of $\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|y^*r_i}$ and $|\varphi\rangle_{X_{\bar{C}}\tilde{X}_{\bar{C}}Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{A}}E^{\mathrm{B}}|x_iy^*r_i}$ is a purification of $\varphi_{Y_{\bar{C}}\tilde{Y}_{\bar{C}}E^{\mathrm{B}}|x_iy^*r_i}$, by the Fuchs-van de Graaf inequality and Uhlmann's theorem we can say that there exist unitaries $U_{x_ir_i}$ on $X_{\bar{C}}\tilde{X}_{\bar{C}}E^{\mathrm{A}}$ such that

$$\mathop{\mathbb{E}}_{i\in\bar{C}}\ \mathop{\mathbb{E}}_{\mathsf{P}_{X_iR_i|\mathcal{E},1,y^*}} \||\varphi\rangle\langle\varphi|_{x_iy^*r_i} - (U_{x_ir_i}\otimes\mathbb{1})|\varphi\rangle\langle\varphi|_{y^*r_i}(U^\dagger_{x_ir_i}\otimes\mathbb{1})\|_1 \leq \left(\frac{2\sqrt{108\delta_1}}{\zeta}\right)^{1/2}$$

and by (24) again,

$$\begin{aligned}\mathop{\mathbb{E}}_{i\in\bar{C}}\ \mathop{\mathbb{E}}_{\mathsf{P}_{X_iR_i|\mathcal{E}}} \||\varphi\rangle\langle\varphi|_{x_iy^*r_i} - (U_{x_ir_i}\otimes\mathbb{1})|\varphi\rangle\langle\varphi|_{y^*r_i}(U^\dagger_{x_ir_i}\otimes\mathbb{1})\|_1 &\leq \left(\frac{2\sqrt{108\delta_1}}{\zeta}\right)^{1/2} + \frac{40\sqrt{2\delta_1}}{\zeta}\\ &\leq 2\left(\frac{10800\delta_1}{\zeta^2}\right)^{1/4}\\ &\leq \varepsilon. \qquad (25)\end{aligned}$$

Combining (25) and (21) we get,

$$\mathop{\mathbb{E}}_{i\in\bar{C}}\ \mathop{\mathbb{E}}_{\mathsf{P}_{X_iY_iR_i|\mathcal{E}}} \left\|(U_{x_ir_i}\otimes U_{y_ir_i})|\varphi\rangle\langle\varphi|_{y^*r_i}(U^\dagger_{x_ir_i}\otimes U^\dagger_{y_ir_i}) - |\varphi\rangle\langle\varphi|_{x_iy_ir_i}\right\|_1 \leq \frac{9\varepsilon}{5}.$$

The result then follows by Markov's inequality.

References

1 Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An Information Statistics Approach to Data Stream and Communication Complexity. In *Proceedings of the 43th Annual IEEE Symposium on Foundations of Computer Science, FOCS '02*, pages 209–218, 2002. doi:10.1109/SFCS.2002.1181944.

2 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to Compress Interactive Communication. *SIAM Journal on Computing*, 42(3):1327–1363, 2013. doi:10.1137/100811969.

3 Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Anchoring Games for Parallel Repetition, 2015. arXiv:1509.07466.

4 Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Hardness Amplification for Entangled Games via Anchoring. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC '17, page 303–316, 2017. doi:10.1145/3055399.3055433.

5 Avraham Ben-Aroya, Oded Regev, and Ronald de Wolf. A Hypercontractive Inequality for Matrix-Valued Functions with Applications to Quantum Computing and LDCs. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS '08*, pages 477–486, 2008. doi:10.1109/FOCS.2008.45.

6 Mario Berta, Matthias Christandl, and Renato Renner. The Quantum Reverse Shannon Theorem Based on One-Shot Information Theory. *Communications in Mathematical Physics*, 306(3):579–615, 2011. doi:10.1007/s00220-011-1309-7.

7 Mark Braverman. Interactive information complexity. *SIAM Journal on Computing*, 44(6):1698–1739, 2015. doi:10.1137/130938517.

8 Mark Braverman, Ankit Garg, Young Kun Ko, Jieming Mao, and Dave Touchette. Near-Optimal Bounds on Bounded-Round Quantum Communication Complexity of Disjointness. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 773–791, 2015. doi:10.1109/FOCS.2015.53.

9 Mark Braverman and Gillat Kol. Interactive Compression to External Information. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC '18, page 964–977, 2018. doi:10.1145/3188745.3188956.

10 Mark Braverman and Anup Rao. Information Equals Amortized Communication. *IEEE Transactions on Information Theory*, 60(10):6058–6069, 2014. doi:10.1109/TIT.2014.2347282.

11 Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct Product via Round-Preserving Compression. In *Automata, Languages, and Programming*, volume 7965 of *Lecture Notes in Computer Science*, pages 232–243. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-39206-1_20.

12 Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct Products in Communication Complexity. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS '13*, pages 746–755, 2013. doi:10.1109/FOCS.2013.85.

13 Mark Braverman and Omri Weinstein. An Interactive Information Odometer and Applications. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 341–350, 2015. doi:10.1145/2746539.2746548.

14 Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Yao. Informational Complexity and the Direct Sum Problem for Simultaneous Message Complexity. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, FOCS '01*, pages 270–278, 2001. doi:10.1109/SFCS.2001.959901.

15 Richard Cleve, William Slofstra, Falk Unger, and Sarvagya Upadhyay. Perfect Parallel Repetition Theorem for Quantum XOR Proof Systems. *Computational Complexity*, 17(2):282–299, 2008. doi:10.1007/s00037-008-0250-4.

16 Irit Dinur. The PCP Theorem by Gap Amplification. *J. ACM*, 54(3):12–es, 2007. doi:10.1145/1236457.1236459.

17 Irit Dinur, David Steurer, and Thomas Vidick. A Parallel Repetition Theorem for Entangled Projection Games. *Computational Complexity*, 24(2):201–254, 2015. doi:10.1007/s00037-015-0098-3.

18 Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The Communication Complexity of Correlation. *IEEE Transactions on Information Theory*, 56(1):438–449, 2010.

19 Thomas Holenstein. Parallel Repetition: Simplifications and the No-Signaling Case. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, page 411–419, 2007. doi:10.1145/1250790.1250852.

20 Rahul Jain. New Strong Direct Product Results in Communication Complexity. *Journal of the ACM*, 62(3), 2015. doi:10.1145/2699432.

21 Rahul Jain and Hartmut Klauck. New Results in the Simultaneous Message Passing Model via Information Theoretic Techniques. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC '09*, pages 369–378, 2009. doi:10.1109/CCC.2009.28.

22 Rahul Jain, Hartmut Klauck, and Ashwin Nayak. Direct Product Theorems for Classical Communication Complexity via Subdistribution Bounds: Extended Abstract. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, pages 599–608, 2008. doi:10.1145/1374376.1374462.

23 Rahul Jain and Ashwin Nayak. Short Proofs of the Quantum Substate Theorem. *IEEE Transactions on Information Theory*, 58(6):3664–3669, 2012.

24 Rahul Jain, Attila Pereszlényi, and Penghui Yao. A Parallel Repetition Theorem for Entangled Two-Player One-Round Games under Product Distributions. In *2014 IEEE 29th Conference on Computational Complexity (CCC '14)*, pages 209–216, 2014.

25 Rahul Jain, Attila Pereszlényi, and Penghui Yao. A Direct Product Theorem for Two-Party Bounded-Round Public-Coin Communication Complexity. *Algorithmica*, 76(3):720–748, 2016. doi:10.1007/s00453-015-0100-0.

26 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. The Quantum Communication Complexity of the Pointer Chasing Problem: The Bit Version. In *FSTTCS 2002: Foundations of Software Technology and Theoretical Computer Science*, volume 2556 of *Lecture Notes in Computer Science*, pages 218–229, 2002. doi:10.1007/3-540-36206-1_20.

27 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A Direct Sum Theorem in Communication Complexity via Message Compression. In *Automata, Languages and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2003. doi:10.1007/3-540-45061-0_26.

28 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A Lower Bound for the Bounded Round Quantum Communication Complexity of Set Disjointness. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS '03*, pages 220–229. IEEE Computer Society, 2003. doi:10.1109/SFCS.2003.1238196.

29 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. Prior Entanglement, Message Compression and Privacy in Quantum Communication. In *20th Annual IEEE Conference on Computational Complexity (CCC '05)*, pages 285–296, 2005.

30 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. Optimal Direct Sum and Privacy Tradeoff Results for Quantum and Classical Communication Complexity, 2008. arXiv:0807.1267.

31 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A Property of Quantum Relative Entropy with an Application to Privacy in Quantum Communication. *Journal of the ACM*, 56(6), 2009. doi:10.1145/1568318.1568323.

32 Rahul Jain and Penghui Yao. A Strong Direct Product Theorem in Terms of the Smooth Rectangle Bound, 2012. arXiv:1209.0263.

33 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*=RE, 2020. arXiv:2001.04383.

34 Julia Kempe, Oded Regev, and Ben Toner. Unique Games with Entangled Provers are Easy. *SIAM Journal on Computing*, 39(7):3207–3229, 2010. doi:10.1137/090772885.

35 Hartmut Klauck. A Strong Direct Product Theorem for Disjointness. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, pages 77–86, 2010. doi:10.1145/1806689.1806702.

36 Hartmut Klauck, Robert Špalek, and Ronald de Wolf. Quantum and Classical Strong Direct Product Theorems and Optimal Time-Space Tradeoffs. *SIAM Journal on Computing*, 36(5):1472–1493, 2007. doi:10.1137/05063235X.

37 Gillat Kol. Interactive Compression for Product Distributions. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 987–998, 2016. doi:10.1145/2897518.2897537.

38 Troy Lee, Adi Shraibman, and Robert Špalek. A Direct Product Theorem for Discrepancy. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC '08*, pages 71–80, 2008. doi:10.1109/CCC.2008.25.

39 Ran Raz. A Parallel Repetition Theorem. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, page 447–456, 1995. doi:10.1145/225058.225181.

40 Alexander A. Razborov. On the Distributional Complexity of Disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.

41 Ronen Shaltiel. Towards Proving Strong direct Product Theorems. *Computational Complexity*, 12(1-2):1–22, 2003. doi:10.1007/s00037-003-0175-x.

42 Alexander A. Sherstov. Strong Direct Product Theorems for Quantum Communication and Query Complexity. *SIAM Journal on Computing*, 41(5):1122–1165, 2012. doi:10.1137/110842661.

43 Alexander A. Sherstov. Compressing Interactive Communication Under Product Distributions. *SIAM Journal on Computing*, 47(2):367–419, 2018. doi:10.1137/16M109380X.

44 Emanuele Viola and Avi Wigderson. Norms, XOR Lemmas, and Lower Bounds for Polynomials and Protocols. *Theory of Computing*, 4(7):137–168, 2008. doi:10.4086/toc.2008.v004a007.

45 Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, pages 222–227, 1977. doi:10.1109/SFCS.1977.24.

46 Henry Yuen. A Parallel Repetition Theorem for All Entangled Games. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP '16)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:13, 2016. doi:10.4230/LIPIcs.ICALP.2016.77.

Quantum Complexity of Minimum Cut

Simon Apers
CWI, Amsterdam, The Netherlands
Université libre de Bruxelles (ULB), Brussels, Belgium

Troy Lee
Centre for Quantum Software and Information, University of Technology Sydney, Australia

Abstract

The minimum cut problem in an undirected and weighted graph G is to find the minimum total weight of a set of edges whose removal disconnects G. We completely characterize the quantum query and time complexity of the minimum cut problem in the adjacency matrix model. If G has n vertices and edge weights at least 1 and at most τ, we give a quantum algorithm to solve the minimum cut problem using $\tilde{O}(n^{3/2}\sqrt{\tau})$ queries and time. Moreover, for every integer $1 \leq \tau \leq n$ we give an example of a graph G with edge weights 1 and τ such that solving the minimum cut problem on G requires $\Omega(n^{3/2}\sqrt{\tau})$ queries to the adjacency matrix of G. These results contrast with the classical randomized case where $\Omega(n^2)$ queries to the adjacency matrix are needed in the worst case even to decide if an unweighted graph is connected or not.

In the adjacency array model, when G has m edges the classical randomized complexity of the minimum cut problem is $\tilde{\Theta}(m)$. We show that the quantum query and time complexity are $\tilde{O}(\sqrt{mn\tau})$ and $\tilde{O}(\sqrt{mn\tau} + n^{3/2})$, respectively, where again the edge weights are between 1 and τ. For dense graphs we give lower bounds on the quantum query complexity of $\Omega(n^{3/2})$ for $\tau > 1$ and $\Omega(\tau n)$ for any $1 \leq \tau \leq n$.

Our query algorithm uses a quantum algorithm for graph sparsification by Apers and de Wolf (FOCS 2020) and results on the structure of near-minimum cuts by Kawarabayashi and Thorup (STOC 2015) and Rubinstein, Schramm and Weinberg (ITCS 2018). Our time efficient implementation builds on Karger's tree packing technique (STOC 1996).

2012 ACM Subject Classification Theory of computation → Quantum query complexity; Mathematics of computing → Graph algorithms; Theory of computation → Quantum complexity theory

Keywords and phrases Quantum algorithms, quantum query complexity, minimum cut

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.28

Funding *Simon Apers*: Supported in part by the Dutch Research Council (NWO) through QuantERA ERA-NET Cofund project QuantAlgo 680-91-034.
Troy Lee: Supported in part by the Australian Research Council Grant No: DP200100950.

Acknowledgements We would like to thank Ronald de Wolf for discussions which started this paper, and in particular a conversation which led to Theorem 35. We also thank Debmalya Panigrahi and Miklos Santha for helpful conversations on this topic.

1 Introduction

Let $G = (V, w)$ be a weighted graph, where $w : \binom{V}{2} \to \mathbb{R}_{\geq 0}$ assigns a non-negative weight to every edge slot. We denote the edges of G, i.e. the edge slots that are given positive weight, by $E(G)$. For a nontrivial set $\emptyset \neq X \subsetneq V$ let $\Delta_G(X)$ be the set of edges of G with exactly one endpoint in X and one endpoint in $\overline{X} = V \setminus X$. A *cut* of G is a set of edges of the form $\Delta_G(X)$ for some nontrivial set $X \subseteq V$. We call X and $\overline{X}$ the *shores* of the cut. The minimum cut problem is to determine the minimum of $\sum_{e\in\Delta_G(X)} w(e)$ over all non trivial subsets X. This is equivalent to the minimum total weight of edges that need to be removed from G in order to disconnect it. We call this minimum value $\lambda(G)$. A set of edges $\Delta_G(X)$ realizing $\lambda(G)$ is called a *minimum cut* of G. If G is unweighted $\lambda(G)$ is known as the *edge connectivity* of G and is the minimum number of edges whose removal disconnects G.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 28; pp. 28:1–28:33
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

Computing the weight of a minimum cut of a graph is a fundamental computational problem that has been extensively studied in theoretical computer science since at least the 1960s [18, 13]. It is also a problem of great practical importance, with applications to clustering algorithms [8] and evaluating network reliability, among others (see [33] for a survey of applications). Classically it is known that edge connectivity can be computed in nearly linear time even by *deterministic* algorithms [26, 21]. For weighted graphs with m edges, the weight of a minimum cut can be determined in nearly linear time[1] $\tilde{O}(m)$ by a randomized algorithm [25, 30, 16] and in almost linear time $O(m^{1+o(1)})$ by a deterministic algorithm [27].

In this work we study quantum algorithms for the minimum cut problem in two standard models for graph problems, the adjacency matrix and the adjacency array models. In the adjacency matrix model a query consists of a pair $\{u, v\}$ of vertices, and the answer is $w(\{u, v\})$. The adjacency array model allows 3 types of queries: one can query the degree of a vertex v, the name of the i^{th} neighbor of v, according to some arbitrary ordering, and the weight of the edge between v and its i^{th} neighbor.

For classical randomized algorithms, in the adjacency matrix model it is known that even deciding if a graph is connected or not requires $\Omega(n^2)$ queries in the worst case [11]. More recently, the randomized query complexity of edge connectivity was studied by Bishnu, Ghosh, Mishra and Paraashar [7] in a common generalization of the adjacency matrix and adjacency array models called the *local query* model. This model allows queries to the degree of a vertex and to the i^{th} neighbor of a vertex v, as in the adjacency array model, and also queries as to whether or not $\{u, v\}$ is an edge, as in the adjacency matrix model. Over simple graphs G with m edges, they show an $\Omega(m)$ lower bound on the number of local queries needed by a randomized algorithm to succeed with probability $2/3$ for both the problems of determining the edge connectivity and outputting a cut realizing the edge connectivity [7, Theorems 2 and 3].

In this work we completely characterize the quantum query and time complexity of the minimum cut problem in the adjacency matrix model. The complexity depends on what we call the *edge-weight ratio*. We say a graph has edge-weight ratio τ if the ratio of the largest weight of the graph to the smallest is at most τ. When the edge-weight ratio of an n-vertex graph is τ, we give a bounded-error quantum algorithm to solve the minimum cut problem using $\tilde{O}(n^{3/2}\sqrt{\tau})$ queries and time in the adjacency matrix model (Theorem 5). For the unweighted case, i.e. the case $\tau = 1$, one can see this bound is tight as Dürr, Heiligman, Høyer, and Mhalla [11] show that even deciding if a graph is connected or not requires $\Omega(n^{3/2})$ quantum queries in the adjacency matrix model. We extend this bound by showing that for any $1 \leq \tau \leq n$ there is a graph family with edge-weight ratio τ for which solving the minimum cut problem requires $\Omega(n^{3/2}\sqrt{\tau})$ quantum queries to the adjacency matrix (Theorem 35). For $\tau \geq n$ one can always use the trivial $O(n^2)$ algorithm, thus our results characterize the quantum query complexity of the minimum cut problem in the adjacency matrix model for any value of τ.

For the adjacency array model, we give a bounded-error quantum algorithm that solves the minimum cut problem in an n vertex, m edge graph with edge-weight ratio τ using $\tilde{O}(\sqrt{mn\tau})$ quantum queries (Theorem 21). The quantum algorithm runs in time $\tilde{O}(\sqrt{mn\tau} + n^{3/2})$ (Theorem 5). In this case we do not know whether the bound is tight in all regimes. For unweighted graphs ($\tau = 1$) the best lower bound we know of is $\Omega(n)$, which again follows from a lower bound for connectivity [11]. For any $\tau > 1$ we show that the minimum cut

[1] The $\tilde{O}(\cdot)$ notation hides polylogarithmic factors in its argument.

problem requires $\Omega(n^{3/2})$ quantum queries to the adjacency array (Theorem 37). Finally, for any $1 \leq \tau \leq 5n/8$ we show a lower bound of $\Omega(\tau n)$ on the number of quantum adjacency array queries for solving the minimum cut problem (Theorem 40).

In addition to computing the weight $\lambda(G)$ of a minimum cut, all of our upper and lower bounds also apply to outputting the edges or shores of a cut realizing $\lambda(G)$.

1.1 Previous work

We are not aware of any previous work on the quantum complexity of *exact* global minimum cut. The closest work to ours in topic is the recent paper of Apers and de Wolf [2], which in particular shows that in a weighted graph a $(1+\varepsilon)$-approximation to the weight of a minimum cut can be found in time $\tilde{O}(n^{3/2}/\epsilon)$ in the adjacency matrix model and time $\tilde{O}(\sqrt{mn}/\epsilon)$ in the adjacency array model. The sparsifier construction of Apers and de Wolf that yields this approximation also plays a key role in our algorithm.

Another key work for us is the seminal paper of Dürr, Heiligman, Høyer and Mhalla [11] which gives tight bounds for the quantum complexity of many graph problems in both the adjacency matrix and adjacency array models. In particular, they show that determining if a graph is connected or not, i.e. determining if the minimum cut value is zero or positive, requires $\Omega(n^{3/2})$ queries in the adjacency matrix model and $\Omega(n)$ queries in the adjacency array model. These are still the best lower bounds we know of for simple graphs[2] even for the more general problem of computing the edge connectivity. Indeed, we show the $\Omega(n^{3/2})$ connectivity lower bound in the adjacency matrix model is a tight lower bound even on the quantum complexity of edge connectivity. In [11] it is also shown that finding a spanning forest in the adjacency matrix model can be done with a quantum algorithm in queries and time $\tilde{O}(n^{3/2})$, which is a result we will make use of in our time efficient algorithm.

Two classical papers which inspired our algorithm are the works of Kawarabayashi and Thorup (KT) [26] and Rubinstein, Schramm, and Weinberg (RSW) [35]. KT give the first near-linear time deterministic algorithm to compute the edge connectivity of a simple graph $G = (V, E)$. A key idea of KT is to look at a *contraction* of the original graph G. Let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be a partition of V. The contraction $G' = \mathrm{Contract}(G, \mathcal{P})$ is a multi-graph whose vertices are labeled by the sets in $\mathcal{P}$ and which has all the edges of G whose endpoints lie in different sets of $\mathcal{P}$. KT first check the cardinality of all *star* cuts of the form $\Delta_G(\{v\})$, which can be done deterministically in linear time. To find the minimum non-star cut, KT show that any simple graph G with minimum degree d has a contraction $G' = \mathrm{Contract}(G, \mathcal{P})$ that preserves all of the near-minimum non-star cuts of G, but which has only $\tilde{O}(n/d)$ vertices and $\tilde{O}(n)$ edges. Moreover, they show how to find such a contraction deterministically in near-linear time. They then use Gabow's $\tilde{O}(\lambda(G)|E(G)|)$ mincut algorithm [15] to find a minimum cut in G'. If G has m edges then $\lambda(G') = \lambda(G) \leq m/n$, and as $|E(G')| \in \tilde{O}(n)$, this gives a time bound that is nearly linear in m.

RSW follow a similar high-level approach to give a classical randomized algorithm that computes the edge connectivity of a simple graph with *cut queries*. In the cut query model, when the input is a graph G, an algorithm can query any nontrivial set X and receive the answer $|\Delta_G(X)|$. RSW show that the edge connectivity of a simple graph can be computed with high probability by a randomized algorithm after $O(n \log(n)^3)$ cut queries. In fact, this algorithm finds *all* minimum cuts of the graph. The RSW algorithm again first evaluates all

[2] We use the term simple graph to mean an undirected, unweighted graph with no self-loops and no multiple edges.

star cuts. They then remove the log factors from the KT result to show there is a partition $\mathcal{P}$ of V such that $G' = \mathrm{Contract}(G, \mathcal{P})$ preserves all near-minimum cuts of G and has only $O(n)$ edges.[3] Moreover, they show how to efficiently learn this contraction with cut queries. The log factors of the original KT proof were also removed via another algorithmic proof by Lo, Schmidt, and Thorup [28].

Our quantum algorithm will follow the approach taken by RSW to learn such a contraction of G, as is detailed in the next section.

1.2 Technical overview

In this overview we focus on the adjacency matrix model. Apart from the lower bound, most ideas carry over in a straightforward way to the adjacency array model. We start off by explaining the lower bound, as this clearly shows the origin of the $n^{3/2}\sqrt{\tau}$ complexity.

Lower bound on the quantum query complexity

For the lower bound we construct a family of graphs on $2n$ vertices with edge weights in $\{1, \tau\}$. Partition the $2n$ vertices into two sets A and B each of size n. Make a complete graph among the vertices in A where every edge has weight τ and do the same to B. This ensures that $w(\Delta_G(X)) \geq \tau(n-1)$ for any $\emptyset \neq X \subset A$, and the same for B. This large value gives us "cover" to hide either $k-1$ or k edges of weight 1 between A and B. If $k < \tau(n-1)$ these edges will constitute the unique minimum cut, and thus an algorithm that outputs the weight of the minimum cut must determine if we hid $k-1$ or k edges. This is equivalent to determining if there are $k-1$ or k marked items in a search space of size n^2, for which a quantum query lower bound of $\Omega(\sqrt{kn^2})$ is known [32]. In our case, with $k = \tau(n-1) - 1$ this gives a bound of $\Omega(n^{3/2}\sqrt{\tau})$. Thus we see that ultimately the lower bound for minimum cut boils down to the difficulty of counting for quantum algorithms. We will see how a similar task arises in the upper bound as well.

Upper bound on the quantum query complexity

We first describe a quantum algorithm for computing the edge connectivity of an unweighted graph. We will follow the outline of the RSW cut query algorithm, which proceeds in the following way. The algorithm first computes the degree of every vertex of G, thereby determining the minimum cardinality of a star cut. The task is then reduced to finding the minimum cardinality of a non-star cut. To do this, the RSW algorithm first produces an *ε-cut sparsifier* of the graph, following an algorithm due to Benczúr and Karger [5]. An ε-cut sparsifier of $G = (V, E)$ is a sparse weighted graph H whose edge set is a subset of E, but where edges are allowed to be weighted. For every nontrivial X the weight of the cut $\Delta_H(X)$ in H is within a factor of $1 \pm \varepsilon$ of $|\Delta_G(X)|$.

For $\epsilon = 1/100$, the algorithm finds an ε-cut sparsifier H of G. The algorithm is able to write H down in memory and then, without further queries, it can compute the weight of a minimum cut in H, say it is $\lambda(H)$, and enumerate all non-star cuts of H whose weight is at most $(1+3\epsilon)\lambda(H)$. With high probability this includes the shores of all non-star minimum cuts of G. Let $\mathcal{T}$ be the set of all shores of these cuts. The algorithm then computes the coarsest partition $\mathcal{P} = \{P_1, \ldots, P_k\}$ of the vertex set with the property that for all $P_j \in \mathcal{P}$

[3] An $O(n)$ bound on the number of edges implies an $O(n/d)$ bound on the number of vertices in a black-box way.

and $u, v \in P_j$ it holds that $u, v \in X$ or $u, v \in \overline{X}$ for all $X \in \mathcal{T}$. We call $\mathcal{P}$ the set of atoms of $\mathcal{T}$, denoted $\mathrm{atoms}(\mathcal{T})$. As $\mathcal{T}$ is the set of shores of all non-star near-minimum cuts, this means that, for every $P_j \in \mathcal{P}$, no non-star near-minimum cut has an edge with both endpoints in P_j; as $\mathcal{P}$ is the coarsest partition with this property, among such partitions it minimizes the number of edges *between* components of the partition. A key fact is that $\mathrm{Contract}(G, \mathcal{P})$ is a sparse graph.

▶ **Lemma 1** ([26, 35, 28]). *Let $G = (V, E)$ be a simple n-vertex graph with minimum degree d. For a nonnegative $\varepsilon < 1$, let $\mathcal{T} = \{X : |X|, |\overline{X}| \geq 2$ and $|\Delta_G(X)| \leq \lambda(G) + \varepsilon d\}$, that is the set of shores of all non-star cuts whose weight is at most $\lambda(G) + \varepsilon d$, and let $G' = \mathrm{Contract}(G, \mathrm{atoms}(\mathcal{T}))$. Then $|E(G')| = O(n)$.*

By the definition of $\mathcal{P}$ in this lemma, one can also see that G' preserves all of the non-star near-minimum cuts of G. As we already know the minimum degree of G, to determine $\lambda(G)$ it suffices to compute the edge connectivity of G'. For a query algorithm, to do this it suffices to learn the $O(n)$ edges of the graph G'; then one can compute the edge connectivity of G' without further queries. The edge connectivity of G is then the minimum of the minimum degree of G and the edge connectivity of G'.

We phrase the RSW algorithm in an abstract way in terms of four computational primitives. We indicate oracle access to G by square brackets and put the parameters explicitly given to the routines in parentheses.

1. $\mathrm{FindMinStar}[G](\delta)$ – a routine that given oracle access to G finds the minimum weight of a star cut of G with error probability at most δ.
2. $\mathrm{Cut\text{-}Sparsifier}[G](\varepsilon, \delta)$ – a routine that given oracle access to G outputs an ε-cut sparsifer of G with error probability at most δ.
3. $\mathrm{LearnCutAtoms}(H, \lambda, \delta)$ – a routine that given an explicit description of a graph H, a cut threshold λ, and an error probability δ, outputs $\mathcal{P}$, the atoms of the shores of all cuts of weight at most λ, with error probability at most δ.
4. $\mathrm{LearnContraction}[G](\mathcal{P}, M, \delta)$ – a routine that given oracle access to G and a partition $\mathcal{P}$ of the vertex set, learns $\mathrm{Contract}(G, \mathcal{P})$ if it has at most M edges and otherwise outputs NULL, again with error probability at most δ.

In Theorem 19, we show a general upper bound on the query complexity of edge connectivity in terms of the sum of the query complexity of the routines in steps (1), (2), and (4). Step (3) requires no queries. It is somewhat surprising that a randomized algorithm designed for cut queries leads to an optimal quantum query algorithm in the adjacency matrix model. We hope that phrasing the algorithm in this abstract way will make it easy to further apply it to other computational models.

In terms of quantum query complexity in the adjacency matrix model, the cost of the 4 steps are as follows. Item (1) can be done with $O(n^{3/2})$ queries by composing the $O(\sqrt{n})$ query quantum minimum finding algorithm over the n vertices with the n query classical algorithm to evaluate the degree of a vertex. The quantum complexity of (2) was recently studied by Apers and de Wolf [2]. They show that even an ε-*spectral* sparsifier can be found in *time* $\tilde{O}(n^{3/2}/\varepsilon)$ in the adjacency matrix model. For our purposes, we take $\varepsilon = 1/100$ giving an $\tilde{O}(n^{3/2})$ bound here. Item (3) costs no queries as the routine is given an explicit description of H. Item (4) is very similar to the problem that we saw in the lower bound: we have to learn up to M edges in a search space of size $O(n^2)$ which can be done with $O(n\sqrt{M})$ queries. By Lemma 1 we can take $M = O(n)$ resulting in an $O(n^{3/2})$ quantum query bound for this step.

These bounds when taken together imply a quantum algorithm for edge connectivity making $\tilde{O}(n^{3/2})$ queries in the adjacency matrix model.

Extension to weighted graphs

The query complexity of steps 1–3 does not change for weighted graphs. The complexity of step 4, however, depends on the upper bound M on the number of edges in the graph $\mathrm{Contract}(G, \mathcal{P})$, which does depend on the edge weights. To extend the above algorithm to weighted graphs, we prove the following generalization of Lemma 1.

▶ **Lemma 2.** *Let $G = (V, w)$ be a weighted graph with $|V| = n$ and where every edge has weight at most τ. Let $d = \min_{u \in V} w(\Delta_G(\{u\}))$. For a nonnegative $\varepsilon < 1$, let $\mathcal{T} = \{X : |X|, |\overline{X}| \geq 2$ and $w(\Delta_G(X)) \leq \lambda(G) + \varepsilon d\}$ and let $G' = \mathrm{Contract}(G, \mathrm{atoms}(\mathcal{T}))$. Then*

$$w(E(G')) \leq \frac{68\tau n}{(1-\varepsilon)^2} \enspace .$$

This lemma is tight as can be seen from the cycle graph with all edge weights τ. Because the bound necessarily depends on τ, applying this lemma back to the cut query or sequential models does not seem to lead to good algorithms.[4] For quantum algorithms, however, it is exactly what is needed.

If the edge-weight ratio is τ, for constant ε Lemma 2 implies an $O(\tau n)$ upper bound on the number of edges in the contracted graph $\mathrm{Contract}(G, \mathcal{P})$. This means that the LearnContraction step can be performed with $O(n^{3/2}\sqrt{\tau})$ queries. Together with the $\Omega(n^{3/2}\sqrt{\tau})$ query lower bound mentioned above we obtain the following tight characterization of the query complexity of minimum cut in the adjacency matrix model in terms of the edge-weight ratio.

▶ **Theorem 3.** *Let $G = (V, w)$ be an n-vertex weighted graph with edge-weight ratio τ. There is a quantum algorithm that finds the weight and shores of a minimum cut of G with probability at least $3/4$ after $\tilde{O}(n^{3/2}\sqrt{\tau})$ queries to the adjacency matrix of G. Moreover, there is a family of graphs with edge-weight ratio τ for which computing the weight of a minimum cut with bounded-error requires $\Omega(n^{3/2}\sqrt{\tau})$ quantum queries to the adjacency matrix.*

The upper bound for this theorem is given in Theorem 21, and the lower bound in Theorem 35.

Upper bound on the quantum time complexity

Let us now consider the time complexity of the above algorithm, corresponding to the total number of queries and elementary gates in the quantum circuit model that the algorithm uses. Steps (1) and (4) are ultimately applications of Grover's algorithm and can be implemented in time which is just a $O(\log(n))$ factor more than their query complexity. For step (2), Apers and de Wolf already give a time complexity upper bound of $\tilde{O}(n^{3/2}/\varepsilon)$. Thus to get an upper bound on the time complexity it suffices to analyze the routine $\mathrm{LearnCutAtoms}(H, \lambda, \delta)$ from step (3). Given a graph H, this subroutine requires us to output the atoms of $\mathcal{T}$, where $\mathcal{T}$ is the set of shores of all near-minimum cuts of H. For this discussion, one should take near-minimum cuts to mean cuts of weight at most $(1 + 1/100)\lambda(H)$. It is known that an n-vertex graph H has at most $O(n^2)$ cuts of weight $< 3\lambda(H)/2$ [22]. Thus we know that $|\mathcal{T}|$ is not too large. However, we still need to efficiently find these near-minimum cuts.

To do this we build on Karger's seminal work [25] that connects near-minimum cuts with tree packings. Consider a spanning tree T of H, as in Figure 1. A cut in H with shore X is said to *2-respect* T if it cuts at most 2 edges of T, that is $|\Delta_T(X)| \leq 2$. Karger showed how

[4] The randomized cut query complexity of minimum cut for weighted graphs was recently resolved using different techniques by Mukhopadhyay and Nanongkai [30].

to efficiently construct a set of $O(\log n)$ spanning trees in H so that every near-minimum cut 2-respects at least one of them. As each tree has at most $n-1+\binom{n-1}{2}=\binom{n}{2}$ 2-respecting cuts, this family of trees defines a set of shores $\mathcal{T}'$ of cardinality $O(n^2\log n)$ which necessarily contains $\mathcal{T}$. A graph can potentially contain $\binom{n}{2}$ minimum cuts, as witnessed by the cycle graph, thus this bound is nearly tight. Unfortunately, iterating over $\mathcal{T}'$ is still too costly for us.

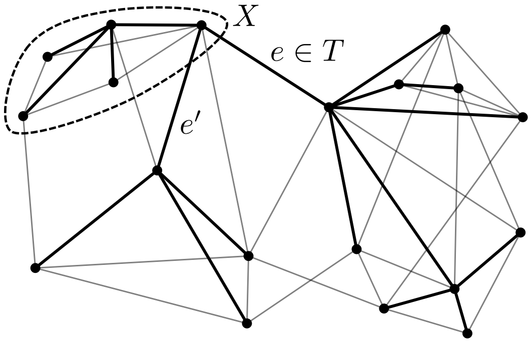


Figure 1 Graph H (thin grey edges) with spanning tree T (thick black edges). The cut with shore X 2-respects T since $|\Delta_T(X)|=|\{e,e'\}|\leq 2$. There are at most $\binom{n}{2}$ such cuts.

As we are only interested in atoms($\mathcal{T}$), and not $\mathcal{T}$ itself, it suffices for us to find a set $\mathcal{S}$ such that atoms($\mathcal{S}$) = atoms($\mathcal{T}$). We call such an $\mathcal{S}$ a *generating set* for atoms($\mathcal{T}$). Our next observation is that there necessarily exists a generating set for atoms($\mathcal{T}$) of size $O(n)$. This follows by a greedy argument: set $\mathcal{S}=\emptyset$ and iterate over all cut shores $X\in\mathcal{T}$, adding X to $\mathcal{S}$ iff atoms($\mathcal{S}\cup X$) $\neq$ atoms($\mathcal{S}$). The resulting $\mathcal{S}$ has the same atoms as $\mathcal{T}$. Moreover, $|\mathcal{S}|\leq n-1$ since every element added to $\mathcal{S}$ creates at least one new atom, there are at most n atoms in total, and $\mathcal{S}=\emptyset$ has 1 atom. While a good start, this still leaves the problem of efficiently finding a small generating set.

We are able to give an explicit description of an $O(n\log(n))$ size generating set. First consider a single spanning tree T of H. For any $f\in E(T)\cup E(T)^{(2)}$ we let shore(f) denote the cut shore such that $\Delta_T(\text{shore}(f))=f$. [5] Now define an unweighted graph $L(T)$ whose vertex set is $E(T)$ and where $f\in E(T)^{(2)}$ is an edge of $L(T)$ iff shore(f) is a near-minimum cut of H (i.e., shore(f) $\in\mathcal{T}$). We show an example in Figure 2. Further, let $O(T)=\{e\in E(T):\exists X\in\mathcal{T}:\Delta_T(X)=\{e\}\}$ index the set of near-minimum cuts that 1-respect T. We prove the following lemma.

▶ **Lemma 4.** *Let $\mathcal{T}'=\{X\in\mathcal{T}:|\Delta_T(X)|\leq 2\}$ be the shores in $\mathcal{T}$ whose corresponding cuts 2-respect T. If F is a spanning forest of $L(T)$ then $\mathcal{S}(T)=\{\text{shore}(f)\mid f\in E(F)\cup O(T)\}$ is a generating set for* atoms($\mathcal{T}'$).

Moreover, since $|E(F)|\leq n-2$ and $|O(T)|\leq n-1$ we have $|\mathcal{S}(T)|\leq 2n-3$. Taking the union of $\mathcal{S}(T)$ over all of the $\log(n)$ spanning trees T of Karger's tree packing gives a generating set $\mathcal{S}$ for $\mathcal{T}$ of size $O(n\log(n))$.

We cannot explicitly write down the graph $L(T)$, but using an efficient data structure for evaluating 2-respecting cuts [30, 17] we can in $O(\log(n))$ time determine whether or not $\{e,e'\}$ is an edge of $L(T)$. This essentially gives us adjacency matrix access to $L(T)$, and

[5] As $\Delta_T(X)=\Delta_T(\overline{X})$, for uniqueness we define a root r in T and choose shore(f) so that it does not contain r.

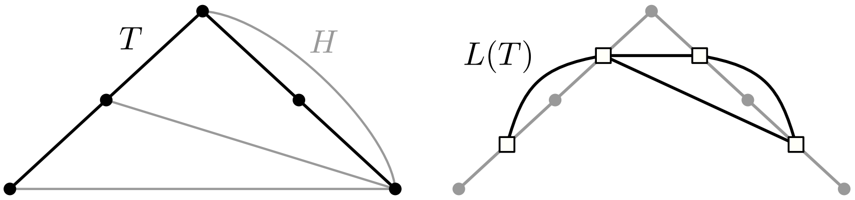


Figure 2 Left: A spanning tree T (thick black edges) of the graph H (thin grey edges) with minimum cut $\lambda(H) = 2$. Right: The associated graph $L(T)$ with vertex set $E(T)$ and $f \in E(T)^{(2)}$ an edge of $L(T)$ iff shore(f) is the shore of a near-minimum cut in H (in this case, a near-minimum cut is a cut of weight $\le \frac{3}{2}\lambda(H)$).

hence we can use the $\tilde{O}(n^{3/2})$ time quantum algorithm from [11] to construct a spanning forest F of $L(T)$. We note that it is conceivable that there exists an efficient classical algorithm to do this. However this would require using further properties of $L(T)$ since classically computing a spanning forest in the adjacency matrix model requires $\Omega(n^2)$ queries.

Once we have the $O(n \log n)$ size generating set $\mathcal{S}$, we still cannot naively compute the atoms of $\mathcal{S}$ because this would again be too costly. Rather, we find the atoms of $\mathcal{S}$ in $\tilde{O}(n)$ time by combining a random hashing scheme with an efficient data structure based on Euler tour trees [20]. This shows that a quantum algorithm can implement step (3), LearnCutAtoms, in time $\tilde{O}(n^{3/2})$. Note that this running time is independent of the kind of oracle access we have to G. This gives the following theorem.

▶ **Theorem 5.** *Let $G = (V, w)$ be an n-vertex weighted graph with m edges and edge-weight ratio τ. There is a quantum algorithm that finds the weight and shores of a minimum cut of G with probability at least $2/3$ in query and time complexity $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model and $\tilde{O}(\sqrt{mn\tau} + n^{3/2})$ in the adjacency array model.*

1.3 Open problems

A few open problems remain from this work.

1. In the adjacency array model there remains a significant gap between the upper and lower bounds we are able to show. For dense graphs the upper bound is $\tilde{O}(n^{3/2}\sqrt{\tau})$ and we have the lower bounds $\Omega(n^{3/2})$ for $\tau > 1$ and $\Omega(\tau n)$ for $1 \le \tau \le n$. We suspect that the quantum query complexity of the minimum cut problem in the adjacency array model is $\tilde{\Theta}(n)$ for simple graphs ($\tau = 1$) and $\tilde{\Theta}(\sqrt{mn\tau})$ for weighted graphs ($1 < \tau \le m/n$), but were unable to prove this.
2. We have given a quantum algorithm with running time $\tilde{O}(m + n^{3/2})$ for the subroutine LearnCutAtoms. By building on our insights we believe that this routine can even be performed by a classical randomized algorithm in near-linear time $\tilde{O}(m)$. This would improve the running time of our quantum algorithm for the minimum cut problem in the adjacency array model from $\tilde{O}(\sqrt{mn\tau} + n^{3/2})$ to $\tilde{O}(\sqrt{mn\tau})$. It also seems of more general interest, giving a weighted (but potentially randomized) generalization of the algorithm by Kawarabayashi and Thorup [26] for finding a contraction of G that preserves all near-minimum cuts and only has $O(\tau n)$ total weight of edges.
3. What is the quantum complexity of determining a $(1+\varepsilon)$-approximation of the minimum cut weight? Apers and de Wolf [2] gave a $(1+\varepsilon)$-approximation algorithm with time and query complexity $\tilde{O}(\sqrt{mn}/\varepsilon)$ in the adjacency array model. For the unweighted case, our algorithm improves this in terms of query complexity by exactly computing the minimum cut with $\tilde{O}(\sqrt{mn})$ queries. Can one approximate the weight of a minimum cut in an unweighted graph with even fewer queries?

2 Preliminaries

For a natural number $n \geq 1$ we let $[n] = \{1, \ldots, n\}$. For a real number x we let $\lfloor x \rceil$ denote the closest integer to x.

2.1 Graph basics and notation

Let V be a finite set and $V^{(2)}$ the set of all subsets of V of cardinality 2. We represent a weighted undirected graph as a pair $G = (V, w)$ where $w : V^{(2)} \to \mathbb{R}$ is a non-negative function. We let $V(G)$ be the vertex set of a graph G and $E(G) = \{e \in V^{(2)} : w(e) > 0\}$ be the set of edges of G. We extend the weight function to sets $S \subseteq V^{(2)}$ by $w(S) = \sum_{e \in S} w(e)$. We say that G is *simple* if $w : V^{(2)} \to \{0, 1\}$ and in this case also denote G as $G = (V, E)$, where E is the set of edges. We call the ratio of the largest edge weight of G to the smallest the *edge-weight ratio* of G.

For a subset $X \subseteq V$ we use the shorthand $\overline{X} = V \setminus X$, and we say X is *non-trivial* if $\emptyset \neq X \subsetneq V$. For disjoint sets $X, Y \subseteq V$ we use $E(X, Y)$ for the set of edges with one endpoint in X and one endpoint in Y. For a non-trivial set X, let $\Delta_G(X) = \{\{i, j\} \in E(G) : i \in X, j \in \overline{X}\}$ be the set of edges of G with one endpoint in X and one endpoint in $\overline{X}$. A *cut* of G is a set of the form $\Delta_G(X)$ for some non-trivial set X. We call X and $\overline{X}$ the *shores* of the cut $\Delta_G(X)$. We call a cut of the form $\Delta_G(\{u\})$ a *star* cut, and refer to all other cuts as *non-star* cuts. The *weight* of a cut S is $w(S)$, which in the case of a simple graph equals $|S|$. We let $\lambda(G) = \min_{\emptyset \neq X \subsetneq V} w(\Delta_G(X))$ be the minimum weight of a cut in G. We call a cut realizing this bound a *minimum cut.* We call a cut $\Delta_G(X)$ satisfying $w(\Delta_G(X)) \leq \alpha\lambda(G)$ an α-near minimum cut. In the case where G is simple we call $\lambda(G)$ the *edge connectivity* of G. We will only use the term edge connectivity in the context of unweighted graphs.

▶ **Definition 6** (Vertex Contraction). *Let $G = (V, w)$ be a weighted graph and $\mathcal{P} = \{S_1, \ldots, S_k\}$ be a partition of V. Define* Contract$(G, \mathcal{P})$ *to be the k-vertex weighted graph $G' = (\mathcal{P}, w')$ where $w'(\{S_i, S_j\}) = w(E(S_i, S_j))$ for each $\{S_i, S_j\} \in \mathcal{P}^{(2)}$.*

Note that as long as $|\mathcal{P}| \geq 2$ it will hold that $\lambda(\text{Contract}(G, \mathcal{P})) \geq \lambda(G)$.

We will also need to make use of graph sparsifiers.

▶ **Definition 7** (Cut sparsifier). *For a weighted graph $G = (V, w)$ and $\varepsilon > 0$ an ε-*cut sparsifier *$H = (V, w')$ of G satisfies*

1. *H is a reweighted subgraph of G, that is $w'(e) > 0$ only if $w(e) > 0$.*
2. *It holds that $(1 - \varepsilon)w(\Delta_G(X)) \leq w'(\Delta_H(X)) \leq (1 + \varepsilon)w(\Delta_G(X))$ for all $\emptyset \neq X \subsetneq V$.*

Cut sparsifiers were first defined by Benczúr and Karger [5] who showed that a weighted graph G has an ε-cut sparsifier H with $O(n \log(n)/\epsilon^2)$ edges, and H can be constructed by a randomized algorithm in time $O(m \log^3(n))$. Fung, Hariharan, Harvey and Panigrahi [14] have since shown that a cut sparsifier with the same bound on the number of edges can be constructed by a randomized algorithm in time $O(m) + \tilde{O}(n/\varepsilon^2)$, and Batson, Spielman and Srivastava [3] have given a deterministic polynomial time construction of sparsifiers with only $O(n/\varepsilon^2)$ edges.

2.2 Atoms

A family of subsets $\mathcal{T} = \{X_1, \ldots, X_k\}$ of V induces a partition of V given by the regions in the Venn diagram of $\mathcal{T}$. We call the resulting sets of this partition the *atoms* of $\mathcal{T}$:

▶ **Definition 8** (Atoms). *Let V be a finite set and let $\mathcal{T} = \{X_1, \ldots, X_k\}$ where each $X_i \subseteq V$. Define* $\mathrm{atoms}(\mathcal{T}) = \{A_1, \ldots, A_\ell\}$ *to be a partition of V such that*

1. *For any $A_j \in \mathrm{atoms}(\mathcal{T})$ and $u, v \in A_j$ it holds that for all $X_i \in \mathcal{T}$ either $u, v \in X_i$ or $u, v \in \overline{X}_i$.*
2. $\mathrm{atoms}(\mathcal{T})$ *is the coarsest partition with property (1).*

▶ **Definition 9** (Generating set). *Let V be a finite set and $\mathcal{T}$ a set of subsets of V. We say that $\mathcal{S} \subseteq \mathcal{T}$ is a* generating set *for* $\mathrm{atoms}(\mathcal{T})$ *if* $\mathrm{atoms}(\mathcal{S}) = \mathrm{atoms}(\mathcal{T})$.

▶ **Proposition 10.** *Let V be a finite set and $\mathcal{T}_1, \mathcal{T}_2$ two sets whose elements are subsets of V. Let $\mathcal{S}_1, \mathcal{S}_2$ be generating sets for* $\mathrm{atoms}(\mathcal{T}_1), \mathrm{atoms}(\mathcal{T}_2)$ *respectively. Then $\mathcal{S}_1 \cup \mathcal{S}_2$ is a generating set for* $\mathrm{atoms}(\mathcal{T}_1 \cup \mathcal{T}_2)$.

Proof. As $\mathcal{S}_1 \subseteq \mathcal{T}_1, \mathcal{S}_2 \subseteq \mathcal{T}_2$ by the definition of a generating set, $\mathcal{S}_1 \cup \mathcal{S}_2 \subseteq \mathcal{T}_1 \cup \mathcal{T}_2$ and $\mathrm{atoms}(\mathcal{T}_1 \cup \mathcal{T}_2)$ is a refinement of $\mathrm{atoms}(\mathcal{S}_1 \cup \mathcal{S}_2)$. Now we show that for any u, v that are in different sets of $\mathrm{atoms}(\mathcal{T}_1 \cup \mathcal{T}_2)$ there is a set $S \in \mathcal{S}_1 \cup \mathcal{S}_2$ which separates them. This will imply that in fact $\mathrm{atoms}(\mathcal{T}_1 \cup \mathcal{T}_2) = \mathrm{atoms}(\mathcal{S}_1 \cup \mathcal{S}_2)$.

If u, v are in different sets of $\mathrm{atoms}(\mathcal{T}_1 \cup \mathcal{T}_2)$ then there must be a $T \in \mathcal{T}_1 \cup \mathcal{T}_2$ which separates them. Suppose without loss of generality that $T \in \mathcal{T}_1$. Then since $\mathrm{atoms}(\mathcal{S}_1) = \mathrm{atoms}(\mathcal{T}_1)$ and u, v are in different sets of $\mathrm{atoms}(\mathcal{T}_1)$, there must be an $S \in \mathcal{S}_1$ which separates u and v. This completes the proof. ◀

2.3 Quantum query and computational models

For general background on the quantum query model we refer the reader to [23]. Here we restrict ourselves to describing the quantum implementation of the input oracles in the adjacency matrix and adjacency array models.

In the adjacency matrix model, on input a weighted graph $G = (V, w)$, classically one can query any $\{u, v\} \in V^{(2)}$ and receive the answer $w(\{u, v\})$. We now describe how to model this by a quantum query. We will assume that the edge weights are given as binary decimal numbers with M_1 bits before the decimal and M_2 bits after the decimal for a total of $M = M_1 + M_2$ bits. The state of the quantum query algorithm will have three registers, a query register, an answer register, and a workspace register. The state of the algorithm will in general be in a superposition of the basis states $|\{u, v\}\rangle|b\rangle|a\rangle$ where $\{u, v\} \in V^{(2)}, b \in \{0, 1\}^M$ and $a \in \mathcal{A}$ for an arbitrary finite set $\mathcal{A}$. On input graph $G = (V, w)$, the input oracle O_G acts on a basis state $|\{u, v\}\rangle|b\rangle|a\rangle$ as

$$\mathsf{O}_G|\{u, v\}\rangle|b\rangle|a\rangle = |\{u, v\}\rangle|b \oplus w(\{u, v\})\rangle|a\rangle \ .$$

In the adjacency array model, on input a weighted n-vertex graph $G = (V, w)$ one can make two types of queries. In the first type, one can query a vertex $v \in V$ and receive its degree $\deg(v)$. The second type is specified by a family of functions $\{f_v : [\deg(v)] \to V\}_{v \in V}$ such that $f_v(i)$ corresponds to the i^{th} neighbor of vertex v (according to some arbitrary but fixed ordering). A query consists of a pair (v, i) for $i \in [\deg(v)]$ and the returned answer is the pair $(f_v(i), w(\{v, f_v(i)\}))$. In this paper we will only need to model the second type of query quantumly. This is because our upper bounds are larger than n so we can let the algorithm classically query all degrees at the start of the algorithm, and in our lower bound on the query complexity of edge connectivity for weighted graphs we assume the algorithm already knows the degree of every vertex. The state of the quantum query algorithm will again have a query register, an answer register, and a workspace register, with the state of the algorithm in general being in a superposition of the basis states $|(v, i)\rangle|x\rangle|b\rangle|a\rangle$ where

$v \in V, i \in [\deg(v)], x \in \{0, \dots, n-1\}, b \in \{0,1\}^M$, and $a \in \mathcal{A}$ for an arbitrary finite set $\mathcal{A}$. We further let $\tau : V \to \{0, 1, \dots, n-1\}$ be a bijection where $|V| = n$. Then the input oracle O_G acts on a basis state in the following way:

$$\mathsf{O}_G|(v,i)\rangle|x\rangle|b\rangle|a\rangle = |(v,i)\rangle|x + \tau(f_v(i)) \bmod n\rangle|b \oplus w(\{v, f_v(i)\})\rangle|a\rangle \ .$$

In Section 5 we will further show that our query algorithms can be implemented in a time efficient manner. We analyze the time complexity in terms of the standard quantum circuit model augmented with two types of oracles. One is the oracle for the input, either in the adjacency matrix or array model, and the second is an oracle to a classical memory of $\tilde{O}(n)$ bits. The latter corresponds to a *quantum random-access-memory* or *QRAM*. We further assume that we can classically update a value in this $\tilde{O}(n)$ bit classical memory in time $\tilde{O}(1)$. The assumption of QRAM access is also required for the time efficiency of the sparsifier construction in [2] which our algorithms build on, and in fact is a necessary (but sometimes inexplicit) assumption in the time analysis of many quantum algorithms for graph problems, e.g. [11, 1, 4].

2.4 Quantum algorithmic primitives

We now go over the quantum subroutines we will need. We need several variants of quantum search.

▶ **Theorem 11** (Quantum search [19]). *Given oracle access to a string $x \in \{0,1\}^N$ such that $|x| > 0$, there is a quantum algorithm that with probability at least $9/10$ returns an i such that $x_i = 1$. The algorithm makes $O(\sqrt{N})$ queries to x and has time complexity $O(\sqrt{N}\log(N))$.*

▶ **Theorem 12** (Exact quantum search, [9, Theorem 4]). *Given a positive integer k and oracle access to a string $x \in \{0,1\}^N$ with $|x| = k$, there is a quantum algorithm that returns an i such that $x_i = 1$ with certainty. The algorithm makes $O(\sqrt{N/k})$ queries to x and has time complexity $O(\sqrt{N/k}\log(N))$.*

▶ **Theorem 13** (Based on [10, Theorem 3]). *Given $t, N \in \mathbb{N}$ with $1 \le t \le N$ and oracle access to $x \in \{0,1\}^N$, there is a quantum algorithm such that*

- *if $|x| \le t$ then the algorithm outputs x with certainty, and*
- *if $|x| > t$ then the algorithm reports so with probability at least $9/10$.*

The algorithm makes $O(\sqrt{tN})$ queries to x and has time complexity $O(\sqrt{tN}\log(N))$.

Proof. Initialize $S = \emptyset$. For $k = t$ down to 1, do: (i) run exact quantum search (from Theorem 12) on x with parameter k, returning an index i, (ii) query x_i and if $x_i = 1$ then add i to S and "unmark" x_i for all future iterations, i.e. implicitly return $x_i = 0$ to future queries of the algorithm.

Finally, run normal quantum search (from Theorem 11) on the indices of x outside of S to check that there are no more solutions. If this returns an $i \notin S$ such that $x_i = 1$, then report $|x| > t$, otherwise return the string y where $y_i = 1$ if $i \in S$ and $y_i = 0$ otherwise.

The query complexity of the algorithm is

$$O\left(\sum_{k=1}^{t} \sqrt{\frac{N}{k}}\right) + O(\sqrt{N}) = O(\sqrt{tN}) \ ,$$

and its time complexity is similarly $O(\sqrt{tN}\log(N))$, as claimed. For correctness, first note that if $|x| > t$ then necessarily an index i such that $x_i = 1$ is remaining in the final step. Quantum search Theorem 11 will find such an index with probability at least $9/10$. It remains to prove that x is learned with certainty if $|x| \le t$. To this end, assume for contradiction that

$|S| < |x|$. Then necessarily there was an iteration k' between t and 1 such that $k' = |x|$. In such case, however, the remaining k' runs of exact quantum search will each return a nonzero index, and so all nonzero indices will be found. This proves that necessarily all indices are found in the first t iterations of exact quantum search, and hence the final quantum search step cannot find an additional nonzero index. ◀

▶ **Theorem 14** (Quantum minimum finding [12]). *Let $N, M \in \mathbb{N}$ be positive integer and $f : [N] \to \mathbb{R}$. There is a quantum algorithm that with probability at least $2/3$ outputs an element of $\operatorname{argmin}_{i\in[N]} f(i)$. The algorithm makes $O(\sqrt{N})$ oracle calls to f and has time complexity $\tilde{O}(\sqrt{N})$.*

▶ **Theorem 15** ([2, Theorem 1]). *Let G be a weighted n-vertex graph with m edges. There is a quantum algorithm that with high probability outputs an explicit description of an ε-cut sparsifier H of G with $\tilde{O}(n/\varepsilon^2)$ edges in query and time complexity $\tilde{O}(\sqrt{mn}/\varepsilon)$ in the adjacency array model or $\tilde{O}(n^{3/2}/\epsilon)$ in the adjacency matrix model.*

Apers and de Wolf actually show a stronger theorem than this in that their algorithm can output a spectral sparsifier instead of just a cut sparsifier. We will not need this additional property, however.

2.5 Problems related to minimum cuts

Let $G = (V, w)$ be a weighted graph. There are three outputs related to a minimum cut of G that one could want from an algorithm: the weight of a minimum cut, the shores of a minimum cut, or the edges in a minimum cut. The relationship between the complexity of these problems is not always obvious, and can depend on the computational model one is studying. All the upper and lower bounds we prove in this paper apply to all three problems.

Say the edge-weight ratio of G is τ. As an example of how we can apply the quantum search algorithm Theorem 13, we show that, given the shores of a minimum cut in G, a quantum algorithm can also find the edges of the cut with $O(n^{3/2}\sqrt{\tau})$ and $O(\sqrt{mn\tau})$ queries in the adjacency matrix and array models respectively. As this matches the complexity of our upper bounds, we will only explicitly mention finding the weight and shores of a minimum cut in Theorem 21.

▶ **Proposition 16.** *Let $G = (V, w)$ be an n-vertex weighted graph with edge-weight ratio τ. Let $\Delta_G(X)$ be a minimum cut of G. Given X, a quantum algorithm can with probability at least $3/4$ output $\Delta_G(X)$ with $O(n^{3/2}\sqrt{\tau})$ queries and time complexity $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model, and $O(\sqrt{mn\tau})$ queries and time complexity $\tilde{O}(\sqrt{mn\tau})$ in the adjacency array model.*

Proof. Consider the adjacency matrix model first. With $O(n)$ queries and time $O(n\log(n))$ we can identify the smallest and largest edge weights of G except error probability at most $1/8$. Thus by rescaling we will henceforth assume that the smallest edge weight is 1 and largest edge weight is at most τ.

Let $x \in \{0,1\}^{\binom{n}{2}}$ denote a bit string labeled by elements of $V^{(2)}$ and set $x(\{u,v\}) = 1$ iff $\{u,v\} \in E(G)$ and u and v are not both in X or both in $\overline{X}$. Given X, a query to x can be answered by a single query to the adjacency matrix of G. As the largest weight of an edge of G is at most τ and $\Delta_G(X)$ is a minimum cut, $w(\Delta_G(X)) \leq \tau(n-1)$. As every edge of G has weight at least 1 we also have $|x| \leq \tau(n-1)$. Thus by Theorem 13, except with error probability $1/8$, we can learn x, and therefore also $\Delta_G(X)$, with $O(n^{3/2}\sqrt{\tau})$ queries and time $\tilde{O}(n^{3/2}\sqrt{\tau})$.

The statement for the adjacency array model follows from Theorem 13 by a similar argument. ◀

3 Number of edges in near-minimum cuts

In this section, we generalize Lemma 1 to weighted graphs. Our proof follows that of Rubinstein, Schramm, and Weinberg [35].

▶ **Lemma 2.** *Let $G = (V, w)$ be a weighted graph with $|V| = n$ and where every edge has weight at most τ. Let $d = \min_{u \in V} w(\Delta_G(\{u\}))$. For a nonnegative $\varepsilon < 1$, let $\mathcal{T} = \{X : |X|, |\overline{X}| \geq 2$ and $w(\Delta_G(X)) \leq \lambda(G) + \varepsilon d\}$ and let $G' = \mathrm{Contract}(G, \mathrm{atoms}(\mathcal{T}))$. Then*

$$w(E(G')) \leq \frac{68\tau n}{(1-\varepsilon)^2} \ .$$

Before proving this lemma we first state and prove a claim.

▷ Claim 17. Let V be a finite set of cardinality n and $r \leq n$ be a positive integer. Let $\mathcal{T} = \{X_1, \ldots, X_k\}$ where each $X_i \subseteq V$. Let $\mathcal{T}_0 = V$ and for $i = 1, \ldots, k$ let $\mathcal{T}_i = \{X_1, \ldots, X_i\}$. Suppose that $\mathcal{T}$ has the property that for all $i = 0, \ldots, k-1$ there is a set $A_j \in \mathrm{atoms}(\mathcal{T}_i)$ that is refined into two sets each of cardinality $\geq r$ in $\mathrm{atoms}(\mathcal{T}_{i+1})$. Then $|\mathcal{T}| \leq \frac{n}{r} - 1$.

Proof. To each $\mathcal{T}_i$ for $i = 1, \ldots, k$ we associate a binary tree B_i. Each vertex of B_i has a label, which will be an element of $\cup_{j=0}^{i} \mathrm{atoms}(\mathcal{T}_j)$. The tree B_1 has root v, labeled by V, and two children v_0, v_1 labeled by the two elements $X_1, \overline{X}_1 \in \mathrm{atoms}(\mathcal{T}_1)$. Note that by definition $|X_1|, |\overline{X}_1| \geq r$.

In general, the tree B_{i+1} is formed from B_i as follows. Initially, set $B_{i+1} = B_i$. Then for every leaf u of B_i which is labeled by a set $Y \in \mathrm{atoms}(\mathcal{T}_i)$ of size $\geq 2r$, if Y is refined into sets Y_1, Y_2 in $\mathrm{atoms}(\mathcal{T}_{i+1})$, then in B_{i+1} the node u is given two children labeled by Y_1 and Y_2, respectively. Note that this construction has the property that only internal vertices of B_i that are labeled by sets of size $\geq 2r$ have children. Call a vertex *big* if it is labeled by a set of size $\geq r$ and *small* otherwise. By construction, every internal vertex of B_i has at least one big child.

Let b_i be the number of big leaves in B_i. We now show by induction that $i \leq b_i - 1$. This will prove the claim as the leaves of B_i partition V and therefore $b_i \leq n/r$.

For $i = 1$ we have that $b_i = 2$ since $|X_1|, |\overline{X}_1| \geq r$, thus the base case holds. Now suppose that $i \leq b_i - 1$, we will show that $i + 1 \leq b_{i+1} - 1$. By definition of $\mathcal{T}$, there must be some set $Y \in \mathrm{atoms}(\mathcal{T}_i)$ which is refined into two sets Y_1, Y_2 both of cardinality at least r in $\mathrm{atoms}(\mathcal{T}_{i+1})$. Further, Y will label some leaf of u of B_i and u will have two children which are big in B_{i+1}. Any other big leaf of B_i which becomes an internal vertex of B_{i+1} must have at least one child which is big. This shows that $b_{i+1} \geq b_i + 1$ and gives the inductive step. ◁

Now we are ready for the proof of Lemma 2.

Proof of Lemma 2. Let $\alpha = \beta = \frac{1}{4}(1-\varepsilon)$ so that $\alpha + \beta \leq \frac{1}{2}(1-\varepsilon)$. Let $K \subseteq \mathcal{T}$ be formed as follows. Initialize K to be empty. Then do the following: while there is an $X \in \mathcal{T}$ such that there is an $A \in \mathrm{atoms}(K), A_1, A_2 \in \mathrm{atoms}(K \cup X)$ such that $A = A_1 \cup A_2$ and $|A_1|, |A_2| \geq \frac{\beta d}{\tau}$, add X to K. By Claim 17, at the end of this process $|K| \leq \frac{\tau n}{\beta d}$. Let $\mathcal{K} = \cup_{X \in K} \Delta_G(X)$ be the set of edges of cuts with shores in K. Throughout this proof, cuts will always be with respect to G and we will henceforth drop the subscript to simply write $\Delta(X)$.

Let $S \subseteq V$ be the set of vertices v such that $w(E(v, V \setminus \{v\}) \cap \mathcal{K}) \geq \alpha \cdot w(v)$. We say that $v \in V$ is *small* if for the $A \in \mathrm{atoms}(K)$ with $v \in A$ there is an $X \in \mathcal{T}$ such that $\mathrm{atoms}(K \cup X)$ refines A into A_1, A_2 with $v \in A_1$ and $|A_1| < \frac{\beta d}{\tau}$.

▷ Claim 18. If v is small then $v \in S$.

Proof. Let $X \in \mathcal{T}$ be the shore of a cut which witnesses that v is small. Let us assume without loss of generality that $v \in X$. Suppose for contradiction that $v \notin S$. There are three possibilities for an edge $\{u, v\}$: either $\{u, v\} \in \mathcal{K}$, or $u \in A_1$, or $u \in A_2$. Let the total weight of these kind of edges be $w_{\mathcal{K}}, w_1, w_2$, respectively. Thus $w(v) = w_{\mathcal{K}} + w_1 + w_2$. We further know that $w_{\mathcal{K}} < \alpha w(v)$ by the assumption that $v \notin S$ and that $w_1 < \beta d$ since $|A_1| < \frac{\beta d}{\tau}$ and the maximum edge weight is τ. This means $w_2 > w(v) - \alpha w(v) - \beta d$. Further note that v contributes weight at least w_2 to the weight of $\Delta(X)$.

As $\Delta(X)$ is not a star cut, we can consider the cut $\Delta(X')$ where $X' = X \setminus \{v\}$. We claim that $w(\Delta(X')) < \lambda$, which is a contradiction. The only difference between $w(\Delta(X))$ and $w(\Delta(X'))$ is the contribution of v. The weight of edges involving v in $\Delta(X')$ is at most $w_{\mathcal{K}} + w_1 < \alpha w(v) + \beta d$. Thus

$$\begin{aligned} w(\Delta(X)) - w(\Delta(X')) &\geq w_2 - (w_{\mathcal{K}} + w_1) \\ &> w(v) - 2\alpha w(v) - 2\beta d \\ &\geq d(1 - 2\alpha - 2\beta) \\ &\geq \varepsilon d \ , \end{aligned}$$

implying that $w(\Delta(X')) < \lambda$. ◁

Let $G' = \mathrm{Contract}(G, \mathrm{atoms}(\mathcal{T}))$. We now bound $w(E(G'))$. We claim that every edge in G' is either in $\mathcal{K}$ or is incident to a vertex in S. For if $\{u, v\} \in E(G')$ but $\{u, v\} \notin \mathcal{K}$, then for a cut $\Delta(Y)$ for $Y \in \mathcal{T}$ with $\{u, v\} \in \Delta(Y)$ it must be the case that there is an $A \in \mathrm{atoms}(K)$ such that $u, v \in A$ and that for the $A_1, A_2 \in \mathrm{atoms}(K \cup Y)$ with $A = A_1 \cup A_2$, one of A_1, A_2 has size $< \frac{\beta d}{\tau}$. This means that either u or v is small and so by Claim 18, $\{u, v\}$ is incident to S.

The number of sets in K is at most $\frac{\tau n}{\beta d}$ and for each $X \in K$ we have $w(\Delta(X)) \leq \lambda + \varepsilon d \leq (1 + \varepsilon)d$. Thus we have that $w(\mathcal{K}) \leq (1 + \varepsilon)\frac{\tau n}{\beta}$.

Let us now bound the weight of edges incident to S. As each vertex $v \in S$ has weight at least $\alpha w(v)$ amongst edges in $\mathcal{K}$ we have that $\frac{\alpha}{2} \sum_{v \in S} w(v) \leq w(\mathcal{K})$. Thus overall we find

$$\begin{aligned} w(E(G')) &\leq w(\mathcal{K}) \left(1 + \frac{2}{\alpha}\right) \\ &\leq (1 + \varepsilon)(\alpha + 2)\frac{\tau n}{\alpha \beta} \\ &\leq \frac{68 \tau n}{(1 - \varepsilon)^2} \ . \end{aligned}$$ ◀

The bound in Lemma 2 is tight up to constant factors. To see this, consider a cycle graph with uniform edge weight τ. Every edge participates in some minimum cut, and hence $G = G'$ and $w(E(G')) = \tau n$.

4 Query-efficient quantum algorithm for minimum cut

We first describe a query-efficient quantum algorithm to find the weight and shores of a minimum cut. In Section 5 we make this algorithm time-efficient. Our quantum query algorithm for minimum cut mainly relies on Lemma 2, and is inspired by a classical randomized algorithm for edge connectivity in the cut query model by Rubinstein, Schramm, and Weinberg (RSW) [35]. The RSW cut query algorithm is based on 4 subroutines whose input/output behavior we describe in Algorithms 1–4 below. For weighted graphs, we need an additional subroutine to compute the maximum weight of an edge in the graph which is stated in Algorithm 5. We describe all these subroutines in an abstract way to make it easy to (i) describe the time-efficient algorithm in the next section, and (ii) to instantiate this algorithm for other query models in the future. We indicate oracle access to G by square brackets and put the parameters explicitly given to the routines in parentheses.

Algorithm 1 $\mathrm{FindMinStar}[G](\delta)$.

Input: Oracle access to a weighted graph G, error parameter δ.

Output: With probability at least $1-\delta$ output $v \in \mathrm{argmin}_{u\in V}\, w(\Delta_G(\{u\}))$ and $d_{\min} = \min_{u\in V} w(\Delta_G(\{u\}))$.

Algorithm 2 $\text{Cut-Sparsifier}[G](\varepsilon,\delta)$.

Input: Oracle access to a weighted graph G, sparsifier accuracy parameter ε, error parameter δ.

Output: With probability at least $1-\delta$ output an integer-weighted ε-cut sparsifier H of G with $\tilde{O}(n/\epsilon^2)$ edges.

Algorithm 3 $\mathrm{LearnCutAtoms}(H,\lambda,\delta)$.

Input: Adjacency array description of H, cut threshold λ, and error parameter δ.

Output: Define the set $\mathcal{T} = \{X : |X|, |\overline{X}| \geq 2, w(\Delta_H(X)) \leq \lambda\}$. With probability at least $1-\delta$ output $\mathrm{atoms}(\mathcal{T})$.

Algorithm 4 $\mathrm{LearnContraction}[G](\mathcal{P},M,\delta)$.

Input: Oracle access to a weighted graph G, a partition $\mathcal{P}$ of $V(G)$, a natural number M, and error parameter δ.

Output: Let $G' = \mathrm{Contract}(G,\mathcal{P})$. With probability at least $1-\delta$ return G' if the number of edges of G' is at most M, and otherwise return NULL.

Algorithm 5 $\mathrm{FindMaxWeight}[G](\delta)$.

Input: Oracle access to a weighted graph G, error parameter δ.

Output: With probability at least $1-\delta$ output τ, the maximum weight of an edge of G.

We combine these subroutines in Algorithm 6 to give a template for solving the minimum cut problem in an abstract query model.

Algorithm 6 Query algorithm for minimum cut.

```
   Input: Oracle access to a weighted graph $G$
   Output: $\lambda(G)$ and the shores of a minimum cut of $G$.
1: $(v, d_{\min}) \leftarrow \mathrm{FindMinStar}[G](\frac{1}{20})$.
2: $\tau \leftarrow \mathrm{FindMaxWeight}[G](\frac{1}{20})$.
3: $H = (V, w') \leftarrow \text{Cut-Sparsifier}[G](\frac{1}{100}, \frac{1}{20})$.
4: Compute $\lambda(H)$.
5: $\mathcal{P} = \{S_1, \dots, S_k\} \leftarrow \mathrm{LearnCutAtoms}(H, (1 + \frac{1}{100})\lambda(H), \frac{1}{20})$.
6: $G' \leftarrow \mathrm{LearnContraction}[G](\mathcal{P}, 100\tau n, \frac{1}{20})$. If $G' = \mathrm{NULL}$ then abort.
7: Compute the weight $\lambda(G')$ and shores $(Y, V(G') \setminus Y)$ of a minimum cut in $G'$.
8: If $d_{\min} \leq \lambda(G')$ output $(d_{\min}, (\{v\}, V \backslash \{v\}))$. Otherwise, let $Z = \cup_{S_i \in Y} S_i$ and output
   $(\lambda(G'), (Z, \overline{Z}))$.
```

▶ **Theorem 19.** *Let G be a weighted graph with n vertices, minimum edge weight at least 1, and maximum edge weight τ. Algorithm 6 finds the weight and shores of a minimum cut of G with probability at least $3/4$. The number of queries of the algorithm is the sum of the number of queries of the subroutines FindMinStar$[G](\frac{1}{20})$, FindMaxWeight$[G](\frac{1}{20})$, Cut-Sparsifier$[G](\frac{1}{100}, \frac{1}{20})$, and LearnContraction$[G](\mathcal{P}, 100\tau n, \frac{1}{20})$.*

Proof. Queries to the input graph G are only made in steps $1, 2, 3$, and 6. This gives the statement about the complexity of the algorithm.

Next let us deal with the error probability. With probability at least $16/20$ steps 1–5 return correctly by the definition of these subroutines and the error parameter provided. Let us now assume this is the case. Then $H = (V, w')$ is a valid ε-sparsifier of G for $\varepsilon = 1/100$. Let $X \in \mathcal{T}$. Then we have

$$w(\Delta_G(X)) \leq (1 + \varepsilon) w'(\Delta_H(X)) \leq (1 + \varepsilon)(1 + 3\varepsilon)\lambda(H) \leq (1 + \varepsilon)^2 (1 + 3\varepsilon)\lambda(G) \ .$$

We have $(1+\varepsilon)^2(1+3\varepsilon) \leq \frac{11}{10}$ by the choice of ε, and so $w(\Delta_G(X)) \leq \frac{11}{10}\lambda(G) \leq \lambda(G) + \frac{1}{10} d_{\min}$ since $\lambda(G) \leq d_{\min}$. Thus by Lemma 2, the total weight of edges in Contract$(G, \mathcal{P})$ will be at most $100\tau n$. As we assume the minimum weight of an edge is at least 1, the number of edges in Contract$(G, \mathcal{P})$ will also be at most $100\tau n$. Hence except with probability at most $1/20$, LearnContraction will correctly return Contract$(G, \mathcal{P})$ in step 5.

We have now argued that with probability at least $3/4$ all subroutines will correctly return. We now argue correctness assuming that this is the case. In this case, G' will be a valid contraction of G and so $\lambda(G') \geq \lambda(G)$. Thus if $\lambda(G)$ is achieved by a star cut the algorithm will return correctly.

Let us now assume that $d_{\min} > \lambda(G)$ and let $\Delta_G(X)$ be a non-star cut with $w(\Delta_G(X)) = \lambda(G)$. We have

$$w'(\Delta_H(X)) \leq (1 + \varepsilon) w(\Delta_G(X)) = (1 + \varepsilon)\lambda(G) \leq \frac{1 + \varepsilon}{1 - \varepsilon}\lambda(H) \leq (1 + 3\varepsilon)\lambda(H) \ ,$$

where the last step holds as $\varepsilon \leq \frac{1}{3}$. This means $X \in \mathcal{T}$ and therefore no edge of $\Delta_G(X)$ will be contracted in $G' = \text{Contract}(G, \mathcal{P})$. Thus $\lambda(G') \leq \lambda(G)$ and as the edge connectivity cannot decrease in a contraction in fact $\lambda(G') = \lambda(G)$. Hence the algorithm returns correctly in step 8. ◀

▶ **Lemma 20.** *Let $G = (V, w)$ be a weighted graph with n vertices and m edges. Subroutines FindMinStar$[G](\frac{1}{20})$, FindMaxWeight$[G](\frac{1}{20})$, Cut-Sparsifier$[G](\frac{1}{100}, \frac{1}{20})$ can be implemented by a quantum algorithm with query and time complexity $\tilde{O}(n^{3/2})$ in the adjacency matrix and $\tilde{O}(\sqrt{mn})$ in the adjacency array model.*

LearnContraction$[G](\mathcal{P}, 100\tau n, \frac{1}{20})$ can be implemented by a quantum algorithm with query and time complexity $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix and $\tilde{O}(\sqrt{mn\tau})$ in the adjacency array model.

Proof. First note that in the adjacency array model we may assume that $m \geq n$. Otherwise, $\sqrt{mn} \geq m$ and we can perform each task classically in $\tilde{O}(m)$ time and queries. We consider each of the subroutines in turn:

FindMinStar$[G](\frac{1}{20})$**:** In the adjacency matrix model we can compute $w(\Delta_G(\{v\})$ with $n-1$ classical queries to the adjacency matrix. We can compose this with quantum minimum finding to find the minimum weight of a star cut and a vertex realizing this in query and time complexity $\tilde{O}(n^{3/2})$ by Theorem 14.

In the adjacency array model we first classically query the degrees of all the vertices with n queries. In a simple graph this suffices to determine the minimum weight of a star cut. In a weighted graph we continue as follows. For $1 \leq \ell \leq \lceil \log n \rceil$, define the bucket $B_\ell \subseteq V$ as the subset of nodes v that have degree in $[2^{\ell-1}, 2^\ell)$. As the sum of the degrees is $2m$ we have that $|B_\ell| \leq 2m/2^{\ell-1}$. Finding the minimum $\min_{v \in B_\ell} w(\Delta_G(\{v\}))$ over a single bucket has quantum query and time complexity $\tilde{O}(\sqrt{mn})$: we can compute $w(\Delta_G(\{v\}))$ for a single $v \in B_\ell$ using at most 2^ℓ classical queries, and then do quantum minimum finding over the $|B_\ell| \leq 2m/2^{\ell-1}$ nodes in B_ℓ. This has total query and time complexity $\tilde{O}(2^\ell \sqrt{2m/2^{\ell-1}}) \in \tilde{O}(\sqrt{m2^\ell}) \in \tilde{O}(\sqrt{mn})$. We do this for each of the $\lceil \log n \rceil$ buckets and we output the minimum overall weight and a vertex realizing this. This yields a total time and query complexity $\tilde{O}(\sqrt{mn})$.

FindMaxWeight$[G](\frac{1}{20})$**:** This amounts to finding the maximum of a set of n^2 numbers in the adjacency matrix model, or m numbers in the adjacency list model. By Theorem 14 this has query and time complexity $\tilde{O}(n)$ and $\tilde{O}(\sqrt{m})$, respectively.

Cut-Sparsifier$[G](\frac{1}{100}, \frac{1}{20})$**:** A $\frac{1}{100}$-cut sparsifier with $\tilde{O}(n/\epsilon^2)$ edges can be constructed with high probability in query and time complexity $\tilde{O}(n^{3/2})$ in the adjacency matrix model or $\tilde{O}(\sqrt{mn})$ in the adjacency array model by Theorem 15.

LearnContraction$[G](\mathcal{P}, 100\tau n, \frac{1}{20})$**:** First we handle a trivial case. If $\tau \geq n$ then we can classically learn the input in time $n^2 = O(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model and time $m = O(\sqrt{mn\tau})$ in the adjacency array model. Thus we can assume $\tau < n$.

First we do the adjacency matrix case. Let $x \in \mathbb{R}^{\binom{n}{2}}$ be a vector whose entries are labeled by elements of $V^{(2)}$ and where $x(e) = w(e)$ if the endpoints of e are in distinct elements of $\mathcal{P}$ and $x(e) = 0$ otherwise. A query to an entry of x can be answered with one query to the adjacency matrix of G. Let $\hat{x} \in \{0,1\}^{\binom{n}{2}}$ be defined by $\hat{x}(e) = 1$ if $w(e) > 0$ and $\hat{x}(e) = 0$ otherwise. We can also answer a query to $\hat{x}$ with one query to the adjacency matrix of G. By Theorem 13 in query and time complexity $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model we can with probability at least $9/10$ output $\hat{x}$ if $|\hat{x}| \leq 100\tau n$ and otherwise output NULL. We can then classically query x in the non-zero locations of $\hat{x}$ with $100\tau n = O(n^{3/2}\sqrt{\tau})$ more classical queries to output x. This fulfils the specification of LearnContraction.

Similarly, in the adjacency array model let $x \in \mathbb{R}^m$ be labeled by entries of the adjacency array of G and define $x(e) = w(e)$ if the endpoints of e are in distinct elements of $\mathcal{P}$ and $x(e) = 0$ otherwise. Let $\hat{x}(e) = 1$ if $x(e) > 0$ and $\hat{x}(e) = 0$ otherwise as before. A query to an entry of x or $\hat{x}$ can be answered with one query to the adjacency array of G. Again by Theorem 13, in query and time complexity $\tilde{O}(\sqrt{mn\tau})$ in the adjacency array model we can with probability at least $9/10$ output $\hat{x}$ if $|\hat{x}| \leq 100\tau n$ and otherwise output NULL. With $100\tau n = O(\sqrt{mn\tau})$ more queries we can then output x. ◀

▶ **Theorem 21.** *Let $G = (V, w)$ be an n-vertex weighted graph with m edges and edge-weight ratio τ. There is a quantum algorithm that finds the weight and shores of a minimum cut of G with probability at least $3/4$ after $\tilde{O}(n^{3/2}\sqrt{\tau})$ queries to the adjacency matrix of G or $\tilde{O}(\sqrt{mn\tau})$ queries to the adjacency array.*

Proof. First we use the minimization analogue of FindMaxWeight to find the minimum edge weight α. Then by normalizing by $1/\alpha$ we may assume that all edge weights are at least 1 and apply Theorem 19. The bound on the quantum query complexities then follows from Lemma 20. ◀

5 Time-efficient quantum algorithm for minimum cut

In this section we describe a quantum algorithm for computing the weight of a minimum cut of a weighted graph with time complexity $\tilde{O}(\sqrt{mn\tau} + n^{3/2})$ in the adjacency array model and $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model. In the adjacency matrix model this is optimal up to polylogarithmic factors. Our algorithm is a time-efficient implementation of Algorithm 6. The running time of this algorithm is the sum of the running time of its 4 subroutines, and we have already analyzed the complexity of 3 of those subroutines in Lemma 20. Thus it now suffices to give a time-efficient implementation of the subroutine LearnCutAtoms, as formalized in the next lemma.

▶ **Lemma 22.** *Let $\kappa(n)$ denote the maximum time complexity of a quantum algorithm for the subroutine LearnCutAtoms$(H, (1 + \frac{1}{100})\lambda(H), \frac{1}{20})$ over weighted n-vertex graphs H with $\tilde{O}(n)$ edges. Let G be a weighted graph with n vertices, m edges, and edge-weight ratio τ. There is a quantum algorithm to compute the weight and shores of a minimum cut of G with probability at least $2/3$ that runs in time $\kappa(n) + \tilde{O}(\sqrt{mn\tau})$ in the adjacency array model and $\kappa(n) + \tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model.*

Proof. First we use minimum finding Theorem 14 to determine the minimum α and maximum β edge weights with error probability at most $1/12$. This requires time $\tilde{O}(\sqrt{m})$ in the adjacency array model and $\tilde{O}(n)$ in the adjacency matrix model and so will be low order to the time bounds stated in the lemma. From α, β we compute the edge-weight ratio $\tau = \beta/\alpha$. By multiplying all edge weights by $1/\alpha$ we may assume that the minimum edge weight is 1 and the maximum edge weight is τ.

If $\tau > m/n$ (in the adjacency array model) or $\tau > n$ (in the adjacency matrix model), then we simply run a randomized near-linear time algorithm (e.g., [25]) for calculating the weight and shores of a minimum cut of G. This then takes time $\tilde{O}(m) \in \tilde{O}(\sqrt{mn\tau})$ in the array model and $\tilde{O}(n^2) \in \tilde{O}(n^{3/2}\sqrt{\tau})$ in the matrix model. We can hence assume that $\tau \leq m/n$ in the array model and $\tau \leq n$ in the matrix model.

We use a quantum implementation of Algorithm 6. By Theorem 19 this algorithm has error probability at most $1/4$, thus our overall error probability will be at most $1/3$ as desired. For the running time it suffices to analyze the quantum time complexity of all 8 steps. In Lemma 20 we show that the time complexity of steps 1–3 and 6 is $\tilde{O}(\sqrt{mn\tau})$ in the adjacency array model and $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model. For step 4, we can use a randomized near-linear time algorithm (e.g., [25]) for calculating the weight and shores of a minimum cut of H. As H has $\tilde{O}(n)$ edges this takes time $\tilde{O}(n)$. In step 7, we compute the weight and shores of a minimum cut in G' which has at most $100\tau n$ edges by the definition of LearnContraction. This takes time $\tilde{O}(\tau n)$, which is $\tilde{O}(\sqrt{mn\tau})$ in the array model (by the assumption $\tau \leq m/n$) or $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the matrix model (by the assumption $\tau \leq n$). Finally, step 8 is trivial and the quantum time complexity of step 5 is exactly $\kappa(n)$. ◀

This section is hence devoted to proving the following theorem.

▶ **Theorem 23.** *Let H be an n-vertex weighted graph with m edges. There is a quantum algorithm that implements LearnCutAtoms$(H, (1+\frac{1}{100})\lambda(H), \frac{1}{20})$ in time $\tilde{O}(m + n^{3/2})$.*

In particular, Theorem 23 implies that $\kappa(n) \in \tilde{O}(n^{3/2})$, and hence we find a time-efficient quantum algorithm.

▶ **Theorem 5.** *Let $G = (V, w)$ be an n-vertex weighted graph with m edges and edge-weight ratio τ. There is a quantum algorithm that finds the weight and shores of a minimum cut of G with probability at least $2/3$ in query and time complexity $\tilde{O}(n^{3/2}\sqrt{\tau})$ in the adjacency matrix model and $\tilde{O}(\sqrt{mn\tau} + n^{3/2})$ in the adjacency array model.*

Proof. Follows from Lemma 22 and Theorem 23. ◀

5.1 Tools

Our time efficient algorithm builds on a number of tools, which we first introduce here.

5.1.1 2-respecting cuts and Karger's theorem

In his seminal work on a near-linear time randomized algorithm for minimum cut [25], Karger combined sparsification with the notion of *tree-respecting cuts*. Consider an n-vertex graph $G = (V, w)$, a spanning tree T and a cut with shore X. We say that the cut *2-respects* T if it cuts at most 2 edges of T, i.e., $|\Delta_T(X)| \leq 2$, and *strictly 2-respects* T if $|\Delta_T(X)| = 2$. Note that the set of cuts which 2-respect T depends only on $E(T)$ and not the weight of edges in T. Note also that there are $n - 1 + \binom{n-1}{2} = \binom{n}{2}$ cuts that 2-respect T.

Karger proved that we can efficiently construct a set of $O(\log n)$ spanning trees of G such that every minimum cut of G will 2-respect a constant fraction of them. This effectively reduces the exponentially large search space for finding a minimum cut to the set of merely $O(n^2 \log n)$ cuts that 2-respect one of the spanning trees. For our purpose, we will use these spanning trees as an efficient representation of the near-minimum cuts of the graph. For this, we need a slight generalization of Karger's theorem on tree-respecting cuts. This shows we can efficiently find $O(\log n)$ spanning trees such that any $(1 + 1/16)$-near-minimum cut 2-respects a constant fraction of them, while Karger's statement was only for minimum cuts. This only requires a minor modification of Karger's proof, but for completeness we provide a proof in Appendix A.

Throughout this section we will use the phrase "with high probability" to mean with probability at least $1 - 1/n^c$ for an arbitrary constant c.

▶ **Theorem 24** ([25, Theorem 4.1])**.** *Let $G = (V, w)$ be a weighted graph with n vertices and m edges. There is a randomized algorithm that in time $O(m \log^2(n) + n \log^4(n))$ time constructs a set of $O(\log n)$ spanning trees such that every $(1 + 1/16)$-near minimum cut of G 2-respects $1/4$ of them with high probability.*

Karger states the runtime of the algorithm in this theorem as $O(m + n\log^3(n))$, but we opt for a simpler proof rather optimizing log factors.

5.1.2 Data structures

We will frequently need to refer to a 2-respecting cut both by its shores and the edges of the tree it cuts. We develop some notation to make this easier.

▶ **Definition 25** (Notation for 2-respecting cuts). *Let T be a tree on vertex set V with root r. Define $N(T) = E(T) \cup E(T)^{(2)}$. For $f \in N(T)$ define* $\mathrm{shore}(f)$ *to be the set $X \subseteq V$ such that $\Delta_T(X) = f$ and X does not contain r. For $X \subseteq V$ such that $|\Delta_T(X)| \leq 2$, let* $\mathrm{cutedges}(X) = \Delta_T(X)$*. We overload both these notations to sets so that* $\mathrm{shore}(Q) = \{\mathrm{shore}(f) : f \in Q\}$ *for $Q \subseteq N(T)$ and similarly* $\mathrm{cutedges}(\mathcal{T}) = \{\Delta_T(X) : X \in \mathcal{T}\}$ *for a set $\mathcal{T}$ of shores of 2-respecting cuts of T.*

With some preprocessing time, we can efficiently evaluate the weight of 2-respecting cuts. The following lemma is very useful.

▶ **Lemma 26** ([17, Lemma 1]). *Given a weighted graph $G = (V, w)$ with n vertices and m edges, and a spanning tree T of G, we can construct in $O(m\log n)$ time a data structure that, for any $f \in N(T)$, reports the weight $w(\Delta_G(\mathrm{shore}(f)))$ of the corresponding 2-respecting cut in $O(\log n)$ time.*

Another data structure that we use is based on the *Euler tour technique* [36, 20]. This is a way of representing a tree that is useful to access and modify data in subtrees. Consider an undirected tree $T = (V_T, E_T)$ with root $r \in V_T$. To T we associate the directed graph $\vec{T} = (V_T, \vec{E}_T)$ obtained by replacing every edge in E_T by a pair of directed edges in opposite directions. Now let $\mathcal{E}_T \in (\vec{E}_T)^{2(n-1)}$ denote an Euler tour in $\vec{T}$, starting and ending in root r. $\mathcal{E}_T$ is a sequence of $2(n-1)$ edges as each directed edge is traversed exactly once.

For every node u in V_T, let $f(u)$ be the index in $\mathcal{E}_T$ of the edge that points toward u, and let $\ell(u)$ be the index of the last edge that points toward u. Now if $T(u)$ is the subtree of T induced by vertex u and all of its descendants, then the subsequence of $\mathcal{E}_T$ starting at $f(u)$ and ending at $\ell(u)$ (both included) is an Euler tour representation of $T(u)$. Hence any subtree corresponds to a subsequence of $\mathcal{E}_T$. We can use this to prove the lemma below, which will be useful to compute $\mathrm{atoms}(\mathcal{T})$ from a set $\mathcal{T}$ of shores of cuts that 2-respect a given tree.

Given a tree whose nodes have some key value, we call a *subtree-add* the increasing or decreasing of the key value in a subtree by some fixed amount.

▶ **Lemma 27.** *Let $T = (V_T, E_T)$ be a tree with key values $\{k_u \mid u \in V_T\}$ of $O(\log n)$ bits. There is a data structure that implements M subtree-adds in time $\tilde{O}(n + M)$.*

Proof. Fix a root node r. Represent T by an Euler tour $\mathcal{E}_T \in (\vec{E}_T)^{2(n-1)}$ and define $f(u), \ell(u)$ for each $u \in V$ as above. Associate to $\mathcal{E}_T$ a list A of length $2(n-1)$ to store the key values, setting $A(i) = k_u$ if the i-th entry of $\mathcal{E}_T$ is an edge whose tail is u. Adding value α to the keys of nodes in subtree $T(u)$ amounts to adding α to every entry in the subsequence in A starting with $f(u)$ and ending with $\ell(u)$ (both included). Call such an operation $\mathtt{ADD}(\alpha, f(u), \ell(u))$.

To implement M `ADD` operations, create a second emtpy list B with length $2(n-1)$. For every operation $\mathtt{ADD}(\alpha, f(u), \ell(u))$, set $B(f(u)) = B(f(u)) + \alpha$ and if $\ell(u) < 2(n-1)$ set $B(\ell(u)+1) = B(\ell(u)+1) - \alpha$. Now do a partial sum transformation of B:

```
Create list $s_B$ of length $2(n-1)$ with $s_B(1) = B(1)$ and $s_B(i) = 0$ for all $i \in [2, 2(n-1)]$.
for $i = 2, 3, \dots, 2(n-1)$ do
    Set $s_B(i) = s_B(i-1) + B(i)$.
end for
```

In total this has time complexity $\tilde{O}(n + M)$ (assuming $\tilde{O}(1)$ cost for arithmetic operations). The final key values are now given by setting $k_u = A(f(u)) + B(f(u))$. ◀

5.2 Generating set for a single tree

Let $G = (V, w)$ be an n-vertex weighted graph and T be a spanning tree of G. Let $Q \subseteq E(T)^{(2)}$ and $\mathcal{M} = \mathrm{shore}(Q)$. In words, $\mathcal{M}$ is an arbitrary set of shores of cuts that *strictly* 2-respect T. The next lemma gives an explicit generating set $\mathcal{S}$ for atoms($\mathcal{M}$) with $|\mathcal{S}| \leq n-2$. We first make a definition that will be used throughout this section.

▶ **Definition 28** (separate). *Let V be a finite set and $X \subseteq V$. For $u, v \in V$ we say that X separates u, v if exactly one of them is in X.*

▶ **Lemma 29.** *Let T be a tree on a vertex set V of cardinality n. Let $\mathcal{M} \subseteq 2^V$ be a set of shores that strictly 2-respect T and let $Q = \mathrm{cutedges}(\mathcal{M})$. Define the graph $L = (E(T), Q)$ and let F be a spanning forest of L. Then $\mathcal{S} = \mathrm{shore}(E(F))$ is a generating set for* atoms($\mathcal{M}$).

Proof. Clearly $E(F) \subseteq Q$ thus $\mathcal{S} \subseteq \mathcal{M}$. This means that atoms($\mathcal{M}$) is a refinement of atoms($\mathcal{S}$). Thus to show atoms($\mathcal{S}$) = atoms($\mathcal{M}$) it suffices to show that any $u, v \in V$ that are in different sets of atoms($\mathcal{M}$) are also in different sets of atoms($\mathcal{S}$).

The key fact we need is that if $\Delta_T(X) = \{e, e'\}$ then X separates u, v iff exactly one of e, e' is on the path from u to v in T. Suppose that u, v are in different sets of atoms($\mathcal{M}$), that is there is an $X \in \mathcal{M}$ which separates them. Say that $\Delta_T(X) = \{e_{\text{in}}, e_{\text{out}}\}$ where e_{in} is on the $u - v$ path in T and e_{out} is not. Then $\{e_{\text{in}}, e_{\text{out}}\} \in Q$ and therefore there must be a path between e_{in} and e_{out} in the spanning forest F. Let $(e_0, e_1, e_2, \dots, e_k)$, where $e_0 = e_{\text{in}}, e_k = e_{\text{out}}$, be the sequence of vertices on this path in F. As e_{in} is on the $u - v$ path in T and e_{out} is not, there must be consecutive vertices e_i, e_{i+1} where e_i is on the $u - v$ path in T and e_{i+1} is not. As $\{e_i, e_{i+1}\} \in E(F)$ there is an $X \in \mathcal{S}$ which separates u and v. ◀

▶ **Lemma 30.** *Let $G = (V, w)$ be an n-vertex weighted graph with m edges and T a spanning tree of G. For a real number $\alpha \geq 1$, let $\mathcal{T} = \{X \subseteq V : w(\Delta_G(X)) \leq \alpha\lambda(G), |\Delta_T(X)| \leq 2\}$. There is a quantum algorithm that outputs with high probability a set $Q \subseteq N(T)$ in time $\tilde{O}(m + n^{3/2})$ such that $|Q| \leq 2n - 3$ and $\mathcal{S} = \mathrm{shore}(Q)$ is a generating set for* atoms($\mathcal{T}$).

Proof. Let $\mathcal{T}_1 = \{X \in \mathcal{T} : |\Delta_T(X)| = 1\}$ and $\mathcal{T}_2 = \{X \in \mathcal{T} : |\Delta_T(X)| = 2\}$. Let $Q_1 = \mathrm{cutedges}(\mathcal{T}_1)$ and $Q_2 = \mathrm{cutedges}(\mathcal{T}_2)$. Let F be a spanning tree for $L = (E(T), Q_2)$. By Lemma 29, $\mathcal{R} = \mathrm{shore}(E(F))$ is a generating set for atoms($\mathcal{T}_2$) and $|\mathcal{R}| \leq n - 2$ as F is a spanning tree of an $n-1$-vertex graph. Thus by Proposition 10, $\mathcal{S} = \mathcal{T}_1 \cup \mathcal{R}$ is a generating set for $\mathcal{T}$ of size at most $2n - 3$. Thus taking $Q = Q_1 \cup E(F)$ satisfies the conditions of the lemma.

Now we must show how to efficiently output Q. We can first run a near-linear time classical randomized algorithm to compute $\lambda(G)$ [25]. We then in near-linear time set up the data structure given by Lemma 26. For an $f \in N(T)$ this lets us check in time $O(\log(n))$ if $f \in Q_1 \cup Q_2$. We can then cycle over the edges $e \in E(T)$ to create the set Q_1 classically in time $\tilde{O}(n)$. It now remains to construct a spanning tree of $L = (E(T), Q_2)$. For any $f \in E(T)^{(2)}$ we can use the data structure to check in $O(\log n)$ time if $f \in Q_2$. This gives us adjacency matrix access to L with $O(\log n)$ overhead for each query. Now we can use the quantum algorithm from [11] that with high probability outputs a spanning forest of an n vertex graph in the adjacency matrix model with $\tilde{O}(n^{3/2})$ queries and time. Thus we can use this algorithm to construct a spanning forest F of L. We then output $Q = Q_1 \cup E(F)$ as desired. ◀

Now we have an implicit representation cutedges($\mathcal{S}$) of a generating set $\mathcal{S}$ for atoms($\mathcal{T}$), where $\mathcal{T}$ is the set of near-minimum cuts of a graph G that 2-respect a tree T. What we need, however, is to actually output atoms($\mathcal{T}$). In the following lemma we show how to do this efficiently by combining random hashing with Euler tour trees.

▶ **Lemma 31.** *Let T be a tree on a vertex set V of size n, $Q \subseteq N(T)$, and $\mathcal{S} = \mathrm{shore}(Q)$. Given input Q there is a classical algorithm that with probability at least $1 - 1/n$ outputs* atoms($\mathcal{S}$) *in time $\tilde{O}(n + |Q|)$.*

Proof. Let M be a large integer to be chosen later and consider the following algorithm. Pick $\ell \in \mathbb{Z}_M$ uniformly at random and give every vertex $u \in V$ the key value $k_u = \ell$. For every $f \in Q$, do:

- Pick $\ell \in \mathbb{Z}_M$ uniformly at random and set $k_u = k_u + \ell \pmod M$ for all $u \in \mathrm{shore}(f)$.

Now if u and v are in the same set of atoms($\mathcal{S}$), that is no set of $\mathcal{S}$ separates them, then $k_u = k_v$. On the other hand, if u and v are in different sets of atoms($\mathcal{S}$) then there is some $f \in Q$ such that $u \in \mathrm{shore}(f)$ and $v \notin \mathrm{shore}(f)$, or vice versa. In this case, k_u and k_v are pairwise independent and distributed uniformly at random in $\mathbb{Z}_M$. Hence $k_u = k_v$ with probability $1/M$. Taking a union bound over all pairs u, v, we see that with probability at least $1 - \binom{n}{2}/M$ we have that $k_u \neq k_v$ for all u, v in different sets of atoms($\mathcal{S}$). If we set $M = n^3$ and we let $\mathcal{P}(\{k_u\})$ denote the partition induced by gathering nodes with the same key value, then $\mathcal{P}(\{k_u\}) = \mathrm{atoms}(\mathcal{S})$ with probability at least $1 - 1/n$.

The cost of actually implementing this algorithm is dominated by sequentially updating for every $f \in Q$ the key value for all nodes in shore(f). This amounts to changing the key value in at most 2 subtrees of T:

- If $f = e \in E(T)$, then shore(f) is the subtree $T(u)$ of some node u and we have to change the key value in $T(u)$.
- If $f = \{e, e'\} \in E(T)^{(2)}$, then we distinguish two cases. If one of the two cut edges is a descendant of the other then shore(f) is of the form $T(u) \setminus T(v)$ for two nodes $u, v \in V$. In this case we can update the key values by adding ℓ to $T(u)$ and subtracting ℓ from $T(v)$. If neither of the edge is a descendant of the other then shore(f) is of the form $T(u) \cup T(v)$, and we can update the key values by adding ℓ to $T(u)$ and $T(v)$.

In Lemma 27 we show how to change the key values in $|Q|$ subtrees in total time $\tilde{O}(n + |Q|)$ using Euler tour trees. ◀

We can now put all these pieces together into the following algorithm.

Algorithm 7 Algorithm for finding atoms of the shores of 2-respecting near-minimum cuts.

```
Input: Explicit description of $G = (V, w)$, a spanning tree $T$ of $G$, a real number $\alpha \geq 1$.
Output: atoms($\mathcal{T}$) where $\mathcal{T} = \{X : w(\Delta_G(X)) \leq \alpha\lambda(G)$ and $|\Delta_T(X)| \leq 2\}$.
1: Compute $\lambda(G)$.
2: Create data structure as in Lemma 26 for evaluating the weight of cuts in $G$ that 2-respect $T$.
3: Compute $Q$ such that shore($Q$) is a generating set for atoms($\mathcal{T}$) by Lemma 30.
4: Use Lemma 31 to find and return atoms(shore($Q$)) = atoms($\mathcal{T}$).
```

▶ **Lemma 32.** *Let $G = (V, w)$ be an n-vertex weighted graph with m edges and T a spanning tree of G. Let $\alpha \geq 1$ be a real number and $\mathcal{T} = \{X : w(\Delta_G(X)) \leq \alpha\lambda(G)$ and $|\Delta_T(X)| \leq 2\}$. Algorithm 7 outputs* atoms$(\mathcal{T})$ *with high probability and can be implemented by a quantum algorithm in time $\tilde{O}(m + n^{3/2})$.*

5.3 Time-Efficient quantum algorithm for LearnCutAtoms

We now describe a time-efficient quantum algorithm for outputting atoms$(\mathcal{T})$, where $\mathcal{T}$ is the set of shores of all $(1 + 1/100)$-near-minimum cuts of a weighted graph H. This algorithm combines Karger's tree packing Theorem 24 with the algorithm that produces the atoms of shores of cuts that 2-respect a tree from the previous section (Lemma 32).

Algorithm 8 LearnCutAtoms(H, λ, δ).

```
Input: Explicit description of an $n$-vertex weighted graph $H = (V, w)$ with $m$ edges, a cut threshold $\lambda \leq (1 + 1/16)\lambda(H)$, and an error parameter $\delta$.
Output: atoms$(\mathcal{T})$ where $\mathcal{T} = \{X \subseteq V : w(\Delta_G(X)) \leq \lambda\}$.
1: Construct set of $K \in O(\log n)$ spanning trees $\{T_i\}$ using Theorem 24.
2: for $i = 1, 2, \ldots, K$ do
3:     Use Algorithm 7 to find atoms$(\mathcal{T}_i)$ where $\mathcal{T}_i = \{X \subseteq V : w(\Delta_G(X)) \leq \lambda$ and $|\Delta_{T_i}(X)| \leq 2\}$.
4: end for
5: Output atoms$(\cup_i$atoms$(\mathcal{T}_i))$.
```

▶ **Theorem 23.** *Let H be an n-vertex weighted graph with m edges. There is a quantum algorithm that implements LearnCutAtoms$(H, (1 + \frac{1}{100})\lambda(H), \frac{1}{20})$ in time $\tilde{O}(m + n^{3/2})$.*

Proof. We use Algorithm 8. First let us argue correctness. As $\lambda \leq (1 + 16)\lambda(H)$, by Theorem 24 for every $X \in \mathcal{T}$ there will be a tree T_i such that $\Delta_{T_i}(X) \leq 2$. This means that $\mathcal{T} = \cup_{i=1}^{K} \mathcal{T}_i$. Hence atoms$(\mathcal{T})$ = atoms$(\cup\mathcal{T}_i)$ = atoms$(\cup$atoms$(\mathcal{T}_i))$. By Lemma 32, step (3) correctly outputs atoms$(\mathcal{T}_i)$ for $i = 1, \ldots, K$ with high probability, and thus step (5) will output atoms$(\mathcal{T})$ with high probability.

Now let us analyze the complexity. Step (1) can be done in $\tilde{O}(m)$ time by a classical randomized algorithm by Theorem 24. Step (3) can be done by a quantum algorithm in time $\tilde{O}(m + n^{3/2})$ by Lemma 32, and thus the for loop has the same time bound as $K = O(\log n)$.

Finally, we need to explain how to (classically) implement step (5). First we give every node $v \in V$ a key value $k_v = 0$. Then, for each $i = 1, \ldots, K$, we iterate over the node set and append a $\log n$-bit string to the key value of every node, indicating the component of atoms$(\mathcal{T}_i)$ of which it is part. At the end of this routine every node has a $O(\log^2 n)$-bit key value that indicates its component in atoms$(\mathcal{T})$. The total runtime for this step is $\tilde{O}(n)$. Thus overall the running time is $\tilde{O}(m + n^{3/2})$. ◀

6 Lower bounds

In this section we present lower bounds on the complexity of edge connectivity and weighted minimum cut.

First we describe some existing lower bounds for the case of simple graphs. Let CON$_n$ be the problem of deciding if an input *simple* graph on n vertices is connected or not. This is a special case of edge connectivity, where one wants to decide if the edge connectivity is zero or positive. Dürr, Heiligman, Høyer and Mhalla [11] proved the following quantum query lower bounds on the complexity of CON$_n$.

▶ **Theorem 33** ([11]). *The bounded-error quantum query complexity of* CON_n *is* $\Theta(n^{3/2})$ *in the adjacency matrix model and* $\Theta(n)$ *in the adjacency array model.*

This theorem shows that, in the adjacency matrix model, Theorem 21 is tight up to polylogarithmic factors for simple graphs. For the adjacency array model there is still a gap between the $\Omega(n)$ lower bound from Theorem 33 and the $\tilde{O}(\sqrt{mn})$ upper bound for simple graphs given by Theorem 21.

For the minimum cut problem in a weighted graph we prove separate and distinct lower bounds for the adjacency matrix model and the adjacency array model. All our lower bounds essentially follow by forcing the algorithm to solve a counting problem in order to compute the weight of a minimum cut. We then use the following theorem by Nayak and Wu that gives a lower bound on the quantum query complexity of exact counting.

▶ **Theorem 34** ([32, Corollary 1.2]). *Let* $k, N \in \mathbb{N}$ *with* $2k+1 \leq N$. *Assume query access to* $x \in \{0,1\}^N$ *with the promise that* $|x| = k+1$ *or* $|x| = k-1$. *Any quantum algorithm that correctly decides whether* $|x| = k+1$ *or* $|x| = k-1$ *with probability at least* $2/3$ *must make* $\Omega(\sqrt{Nk})$ *queries.*

6.1 Adjacency matrix model

In the adjacency matrix model we show that for any integer $1 \leq \tau \leq (\lfloor n/2 \rfloor - 1)/2$, in the worst case $\Omega(n^{3/2}\sqrt{\tau})$ adjacency matrix queries are needed to compute the weight of a minimum cut of a graph with edge weights in $\{1, \tau\}$. This matches the upper bound in Theorem 21, and hence settles the quantum query complexity of weighted minimum cut in the adjacency matrix model. For $\tau = 1$ this reproduces the aforementioned $\Omega(n^{3/2})$ bound which follows from [11].

▶ **Theorem 35.** *Let* $n, \tau \in \mathbb{N}$ *satisfy* $1 \leq \tau \leq (\lfloor n/2 \rfloor - 1)/2$. *There is a family of* n*-vertex graphs* $\mathcal{G}$ *all of which have edge weights in* $\{0, 1, \tau\}$ *such that any quantum algorithm that for every graph* $G \in \mathcal{G}$ *computes with probability at least* $2/3$ *the weight of a minimum cut in* G *must make* $\Omega(n^{3/2}\sqrt{\tau})$ *queries in the adjacency matrix model. Similarly, any quantum algorithm that for every graph* $G \in \mathcal{G}$ *computes with probability at least* $2/3$ *the shores* $(X, \overline{X})$ *of a cut realizing the minimum weight must make* $\Omega(n^{3/2}\sqrt{\tau})$ *queries in the adjacency matrix model.*

Proof. Let V be an n-element set and partition V into disjoint sets $V = V_0 \sqcup V_1$ where $|V_0| = \lfloor n/2 \rfloor$, $|V_1| = \lceil n/2 \rceil$. Choose a distinguished vertex $v_0 \in V_0$, and let $V_0' = V_0 \setminus \{v_0\}$. Let $N = |V_0' \times V_1|$ and let $g : V_0' \times V_1 \to [N]$ be a bijection. For every $x \in \{0,1\}^N$ we define a weighted graph $G_x = (V, w_x)$ where

- $w_x(\{u, v\}) = \tau$ if $u, v \in V_0$ or $u, v \in V_1$,
- $w_x(\{u, v\}) = x(g(\{u, v\}))$ if $(u \in V_0', v \in V_1)$ or $(u \in V_1, v \in V_0')$,
- $w_x(\{u, v\}) = 0$ otherwise.

Let $k = \tau(\lfloor n/2 \rfloor - 1)$. In the following $\Delta_{G_x}(\cdot)$ will always be with respect to G_x and we drop the subscript. For any x it holds that $w_x(\Delta(V_0)) = |x|$ and $w_x(\Delta(\{v_0\})) = k$. Now consider any x and a subset $\emptyset \neq Y \subsetneq V$ different from V_0 or V_1. We can prove that $w_x(\Delta(Y)) \geq k$. To this end note that either $\emptyset \neq Y \cap V_0 \subsetneq V_0$ or $\emptyset \neq Y \cap V_1 \subsetneq V_1$. First assume that the former is the case. Then

$$w_x(\Delta(Y)) = \sum_{u \in Y, v \notin Y} w_x(\{u, v\}) \geq \sum_{u \in Y \cap V_0, v \in V_0 \setminus Y} w_x(\{u, v\}) \geq k,$$

as k is the weight of a minimum cut in the complete weighted graph over $\lfloor n/2 \rfloor$ nodes with all edge weights τ. If instead $\emptyset \neq Y \cap V_1 \subsetneq V_1$ then a similar argument shows that $w_x(\Delta(Y)) \geq \tau(\lceil n/2 \rceil - 1) \geq k$.

Thus if $|x| < k$ then $\Delta(V_0)$ will be the unique minimum cut of G_x, and the weight of a minimum cut in G_x will be $w_x(\Delta(V_0)) = |x|$. On the other hand, if $|x| > k$ then the weight of a minimum cut in G_x will be k, which is realized by the star cut $\Delta(\{v_0\})$ (and potentially other cuts in G_x) but not by $\Delta(V_0)$ as $w_x(\Delta(V_0)) = |x| > k$.

Let $\mathcal{S} = \{x \in \{0,1\}^N : |x| \in \{k-1, k+1\}\}$ and $\mathcal{G} = \{G_x : x \in \mathcal{S}\}$. Suppose there was a T query algorithm in the adjacency matrix model that for any $G_x \in \mathcal{G}$ with probability at least $2/3$ output the weight of a minimum cut in G_x. If the output is $< k$ then we know that $|x| = k-1$ and if the output is k then we know that $|x| = k+1$. Moreover, any query to the adjacency matrix of G_x can be simulated by a query to x, thus such an algorithm gives a T query algorithm to determine if $|x| = k-1$ or $|x| = k+1$ when we are promised one of these is the case. Since $\tau \leq (\lfloor n/2 \rfloor - 1)/2$ we have $2k+1 \leq N$ and therefore we may apply Theorem 34 to obtain $T \in \Omega(\sqrt{Nk}) = \Omega(n^{3/2}\sqrt{\tau})$.

Similarly, a T query algorithm in the adjacency matrix model that for any $G_x \in \mathcal{G}$ with probability at least $2/3$ outputs the shores of a cut realizing the minimum weight also implies a T query algorithm to determine if $|x| = k-1$ or $|x| = k+1$. In this case, if $|x| = k-1$ then the output must be $(V_0, \overline{V}_0)$ as these are the shores of the unique minimum cut in G_x. On the other hand, if $|x| = k+1$ then $(V_0, \overline{V}_0)$ is not a correct output. Thus the output of the algorithm lets us determine with probability at least $2/3$ if $|x| = k-1$ or $|x| = k+1$ and we again have $T \in \Omega(\sqrt{Nk}) = \Omega(n^{3/2}\sqrt{\tau})$. ◀

6.2 Adjacency array model

Given adjacency array access to a graph with edge-weight ratio τ, we showed an upper bound of $\tilde{O}(\sqrt{mn\tau})$ on the quantum query complexity of computing the weight of a minimum cut. In this section we prove two distinct lower bounds, each of which is tight in a specific regime. First we show that for any $\tau > 1$ there exists a family of dense graphs on n vertices with edge-weight ratio τ for which computing the weight of a minimum cut requires $\Omega(n^{3/2})$ queries to the adjacency array. This shows that the adjacency array upper bound of Theorem 21 is tight for dense weighted graphs with constant (but non-unit) edge-weight ratio. Secondly and using a different approach, for any $1 \leq \tau \in O(n)$ we prove an $\Omega(\tau n)$ lower bound for a family of dense graphs with edge-weight ratio τ. This shows that we cannot get a quantum speedup when $\tau \in \Omega(n)$.

6.2.1 Constant edge-weight ratio

For the first bound we first need a claim about the minimum cuts of a complete weighted bipartite graph.

▷ Claim 36. Let $n \geq 8$ be a multiple of 4 and $G = (L \sqcup R, w)$ be a weighted bipartite graph with bipartition L, R where $|L| = 3n/4, |R| = n/4$. Further suppose that for every $x \in L, y \in R$ it holds that $w(\{x,y\}) \geq 1$. Then any cut of G that is not of the form $\Delta_G(\{x\})$ for $x \in L$ has weight at least $n/2$.

Proof. First consider a star cut $\Delta_G(\{y\})$ for $y \in R$. This has weight at least $3n/4$, since this is the degree of y and all edges have weight at least 1.

It now remains to show the claim holds for non-star cuts. Consider a general non-star cut with shore $X \cup Y$ with $X \subseteq L, Y \subseteq R$. Let $k = |X|, \ell = |Y|$. As it is a non-star cut we have $k + \ell \geq 2$. By complementing as needed we may also assume that $k \leq 3n/8$. We also have the obvious constraints that $\ell \leq n/4$ and $k, \ell \geq 0$.

As G is a complete weighted bipartite graph with every edge weight at least one we have

$$w(\Delta_G(X \cup Y)) \geq k(n/4 - \ell) + \ell(3n/4 - k) \enspace .$$

As $k \leq 3n/8$ the term $\ell(3n/4 - k)$ is greater than $n/2$ whenever $\ell \geq 2$. Thus we can focus on $\ell \in \{0, 1\}$. If $\ell = 0$ then $k \geq 2$ and so the weight of the cut is at least $k(n/4) = n/2$ as desired. If $\ell = 1$ then the weight of the cut is $k(n/4 - 1) + 3n/4 - k$ which is always at least $3n/4$ as long as $n \geq 8$. ◁

This claim means that if $\min_{x \in L} w(\Delta_G(\{x\})) < n/2$ then this value will be the weight of a minimum cut in G. We can leverage this to show a lower bound as follows. In the next proof, for a function $f : \{0,1\}^n \to \{0,1\}$ we will use $Q_{1/3}(f)$ to denote the quantum query complexity of computing f with error at most 1/3.

▶ **Theorem 37.** *Let $n \geq 8$ be a multiple of 4 and $0 < \varepsilon \leq 1$. There is a family of n-vertex graphs $\mathcal{G}$ all of which have edge weights in $\{1, 1+\varepsilon\}$ such that any quantum algorithm that for every graph $G \in \mathcal{G}$ computes with probability at least 2/3 the weight of a minimum cut in G must make $\Omega(n^{3/2})$ queries in the adjacency array model. Similarly, any quantum algorithm that for every graph $G \in \mathcal{G}$ computes with probability at least 2/3 the shores $(X, \overline{X})$ of a cut realizing the minimum weight must make $\Omega(n^{3/2})$ queries in the adjacency array model.*

Proof. Let $X = \{x \in \{0,1\}^{n/4} : |x| = \lfloor n/8 \rfloor - 1\}$ and $Y = \{y \in \{0,1\}^{n/4} : |y| = \lfloor n/8 \rfloor + 1\}$. For every $x = (x^{(1)}, \ldots, x^{(3n/4)}) \in (X \cup Y)^{3n/4}$ we associate a bipartite graph $G_x = (L \sqcup R, w_x)$ where $L = \{1, \ldots, 3n/4\}, R = \{3n/4 + 1, \ldots, n\}$ and $w_x(\{i, j\}) = 1 + \varepsilon \cdot x^{(i)}(j - 3n/4)$ for every $i \in L, j \in R$. We set $\mathcal{G} = \{G_x : x \in (X \cup Y)^{3n/4}\}$.

Define the function $g : X \cup Y \to \{0, 1\}$ where $g(x) = 0$ iff $x \in X$. We have $Q_{1/3}(g) \in \Omega(n)$ by Theorem 34. Let $f : \{0,1\}^{3n/4} \to \{0,1\}$ be the AND function, for which $Q_{1/3}(f) \in \Omega(\sqrt{n})$. By the composition theorem for quantum query complexity [23, 34], we have $Q_{1/3}(h) \in \Omega(n^{3/2})$ for the composed function $h = f \circ g^{3n/4}$.

Let $x = (x^{(1)}, \ldots, x^{(3n/4)}) \in (X \cup Y)^{3n/4}$. If $h(x) = 1$ then $x^{(i)} \in Y$ for all $i \in [3n/4]$ and the weight of the star cut $\Delta_{G_x}(\{i\}) = n/4 + \varepsilon \cdot (\lfloor n/8 \rfloor + 1)$. As $\varepsilon \leq 1$ this will be the weight of a minimum cut in G_x by Claim 36. On the other hand if $h(x) = 0$ then some $x^{(i)} \in X$ and $\Delta_{G_x}(\{i\}) = n/4 + \varepsilon \cdot (\lfloor n/8 \rfloor - 1)$ and this will be the weight of a minimum cut of G_x. Thus computing the weight of a minimum cut of G_x lets us evaluate $h(x)$. Further, given oracle access to x we can simulate queries to G_x in the adjacency array model. Let A_x be a $3n/4$-by-$n/4$ matrix whose i^{th} row is the vector $1 + \varepsilon x^{(i)}$. Then the vertical concatenation of A with A^T is a valid adjacency array for G_x. To a degree query on vertex i we simply answer $n/4$ if $1 \leq 3n/4$ and $3n/4$ if $3n/4 + 1 \leq i \leq n$. We can also answer a query to the name and weight of the j^{th} neighbor of i with one query to x. This shows that the (1/3)-error quantum query complexity of computing the weight of a minimum cut on graphs in $\mathcal{G}$ in the adjacency array model is at least $Q_{1/3}(f \circ g^{3n/4}) \in \Omega(n^{3/2})$.

Finally, suppose a quantum query algorithm can compute a shore of a minimum cut in G_x with T queries. We know that this shore must be of the form $\{v\}$ for a vertex $v \in L$. Thus with with $O(n)$ more queries the algorithm can classically compute the weight of a minimum cut by querying the weight of the neighbors of v. Thus $T + O(n) \in \Omega(n^{3/2})$, which means $T \in \Omega(n^{3/2})$. This completes the proof. ◀

6.2.2 Large edge-weight ratio

Let $n \in \mathbb{N}$ be a multiple of 4 and V be a vertex set with $|V| = n$. Partition V into four sets V_1, V_2, V_3, V_4.

Now consider an integer τ such that $1 \leq \tau \leq 5n/8$ and $\tau n/10$ is an integer. Fix a set S of $\tau n/10$ "edge disjoint" quadruples $(u_1, u_2, u_3, u_4) \in V_1 \times V_2 \times V_3 \times V_4$. By edge disjoint we mean no pair of consecutive elements (u_i, u_{i+1}) or (u_4, u_1) appears in more than one quadruple. We fix an enumeration of S and refer to the vertices in the ℓ^{th} quadruple as $u_1^\ell, u_2^\ell, u_3^\ell, u_4^\ell$.

For every $x \in \{0,1\}^{\tau n/10}$ we define an n-vertex weighted graph $G_x = (V, w_x)$ where $w_x(\{u, v\}) = \tau$ if $u \neq v \in V_i$ for some $i \in [4]$, and for $\ell \in [\tau n/10]$ we set

$$w_x(\{u_1^\ell, u_2^\ell\}) = w_x(\{u_3^\ell, u_4^\ell\}) = x_\ell,$$
$$w_x(\{u_2^\ell, u_3^\ell\}) = w_x(\{u_4^\ell, u_1^\ell\}) = 1 - x_\ell \ .$$

Otherwise, $w_x(\{u, v\}) = 0$. In words, on each V_i we have a complete graph with all edge weights τ, and for each $\ell \in [\tau n/10]$ we either add unit weight edges $\{u_1^\ell, u_2^\ell\}, \{u_3^\ell, u_4^\ell\}$ or $\{u_2^\ell, u_3^\ell\}, \{u_4^\ell, u_1^\ell\}$ depending on x_ℓ. The construction is depicted in Figure 3.

There are a few important points to note about this definition. First, the edge-weight ratio of G_x is τ. Second, for any X that nontrivially intersects some V_i we have that $w(\Delta_{G_x}(X)) \geq \tau(n/4 - 1)$. This means that such an X cannot be the shore of a minimum cut of G_x. Third, by construction the degree of every vertex of G_x is independent of x. This means that degree queries to G_x can be trivially answered and give us no information about x.

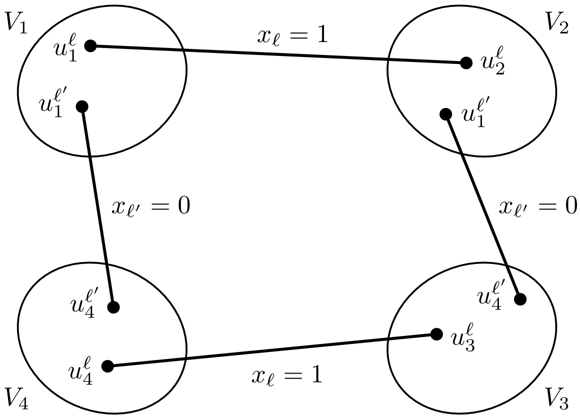


Figure 3 Figure of graph G_x. If $x_\ell = 1$ then we add edges $\{u_1^\ell, u_2^\ell\}$ and $\{u_3^\ell, u_4^\ell\}$. If $x_\ell = 0$ then we add edges $\{u_2^\ell, u_3^\ell\}$ and $\{u_4^\ell, u_1^\ell\}$.

▶ **Lemma 38.** *We can simulate a single query to G_x in the adjacency array model using a single query to x.*

Proof. We first handle degree queries. This can be answered with no queries to x as the degree of a vertex is independent of x.

Now consider a query $(v, k) \in V \times [\deg(v)]$ to which we must answer the name u of the k-th neighbor of v and the edge weight $w_x(\{u, v\})$. For clarity of exposition, we assume $v = u_1^t \in V_1$; the other cases are handled similarly.

- If $k \leq n/4-1$ then return the k-th neighbor u of v inside V_1 and edge weight $w_x(\{u,v\}) = \tau$.
- If $k \geq n/4$ then let $j = k - n/4 + 1$. Letting ℓ denote the index of the j^{th} quadruple of S containing v we query x_ℓ.
  - If $x_\ell = 1$ then return neighbor u_2^ℓ and edge weight $w(\{v, u_2^\ell\}) = 1$.
  - If $x_\ell = 0$ then return neighbor u_4^ℓ and edge weight $w(\{v, u_4^\ell\}) = 1$.

In total this takes a single query to x, which proves the lemma. ◀

Now we can prove the following lemma.

▶ **Lemma 39.** *Fix integers n and τ such that $1 \leq \tau \leq 5n/8$ and $\tau n/10 \in \mathbb{N}$. Consider a string $x \in \{0,1\}^{\tau n/10}$ and the corresponding graph G_x. If $|x| < \tau n/20$ then G_x has a unique minimum cut with shores $(X, \overline{X}) = (V_1 \cup V_2, V_3 \cup V_4)$ and weight $w(\Delta_{G_x}(X)) = 2|x|$. If $|x| > \tau n/20$ then G_x has a unique minimum cut with shores $(X, \overline{X}) = (V_1 \cup V_4, V_2 \cup V_3)$ and weight $w(\Delta_{G_x}(X)) = 2(\tau n/10 - |x|)$.*

Proof. First consider any cut shore that nontrivially intersects some V_i. Since the subgraph $G_x[V_i]$ induced on V_i is a complete graph with edge weights τ, this implies that such a cut has weight at least $\tau(|V_i| - 1) = \tau(n/4 - 1)$. Now consider the small set of remaining cut shores that trivially intersect the V_i's. The weight of each one of these cuts can be easily expressed as a function of the Hamming weight $|x|$ of the input:

$$\begin{aligned} w(\Delta_{G_x}(V_i)) &= \tau n/10, \\ w(\Delta_{G_x}(V_1 \cup V_3)) = w(\Delta_{G_x}(V_2 \cup V_4)) &= 2\tau n/10, \\ w(\Delta_{G_x}(V_1 \cup V_2)) = w(\Delta_{G_x}(V_3 \cup V_4)) &= 2|x|, \\ w(\Delta_{G_x}(V_1 \cup V_4)) = w(\Delta_{G_x}(V_2 \cup V_3)) &= 2(\tau n/10 - |x|). \end{aligned}$$

It is clear that all minimum weight cuts will be among these cuts, and the lemma easily follows. ◀

Using this lemma we can prove the following theorem.

▶ **Theorem 40.** *Let $\tau, n \in \mathbb{N}$ be such that $1 \leq \tau \leq 5n/8$ and $\tau n/20 \in \mathbb{N}$. There exists a family of n-vertex graphs $\mathcal{G}'$ with $\Omega(n^2)$ edges, all of which have edge weights in $\{1, \tau\}$, such that any quantum algorithm that for every graph $G' \in \mathcal{G}'$ computes with probability at least $2/3$ the weight of a minimum cut in G' must make $\Omega(n\tau)$ queries in the adjacency array model. Similarly, any quantum algorithm that for every graph $G' \in \mathcal{G}'$ computes with probability at least $2/3$ the shores $(X, \overline{X})$ of a cut realizing the minimum weight must make $\Omega(n\tau)$ queries in the adjacency array model.*

Proof. First consider the set of strings $\mathcal{X} \subseteq \{0,1\}^{\tau n/10}$ with Hamming weight

$$|x| = \lfloor \tau n/100 \rceil \pm 1 < \tau n/20.$$

By Lemma 39 the graph G_x, $x \in \mathcal{X}$, has a unique minimum cut with shores $(V_1 \cup V_2, V_3 \cup V_4)$ and weight $2|x|$. Now let $\mathcal{G}' = \{G_x : x \in \mathcal{X}\}$ and assume the existence of a quantum algorithm that for every $G_x \in \mathcal{G}'$ computes with probability at least $2/3$ the weight $2|x|$ of a minimum cut in G' with at most q queries to the adjacency array of G_x. By Lemma 38 this is equivalent to outputting the Hamming weight $|x|$ with probability at least $2/3$ for any $x \in \mathcal{X}$ while making only q queries to x. Using Theorem 34 this implies the lower bound $q \in \Omega(\tau n)$.

Next consider the set of strings $\mathcal{X}' \subseteq \{0,1\}^{\tau n/10}$ that have Hamming weight $|x| = \tau n/20 \pm 1$. By Lemma 39 the graph G_x, $x \in \mathcal{X}'$, again has a unique minimum cut. If $|x| = \tau n/20 - 1$ then its shores are $(V_1 \cup V_2, V_3 \cup V_4)$, while if $|x| = \tau n/20 + 1$ then its shores are $(V_1 \cup V_4, V_2 \cup V_3)$. Now assume that there exists a quantum algorithm that with probability at least $2/3$ returns the shores of a minimum weight cut of G_x with at most q queries to the adjacency array of G_x. By Lemma 38 this is equivalent to distinguishing $|x| = \tau n/20 - 1$ from $|x| = \tau n/20 + 1$ with probability at least $2/3$ for any $x \in \mathcal{X}$ while making only q queries to x. Using Theorem 34 this implies the lower bound $q \in \Omega(\tau n)$. ◀

References

1 Andris Ambainis and Robert Špalek. Quantum algorithms for matching and network flows. In *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 172–183. Springer, 2006.

2 Simon Apers and Ronald de Wolf. Quantum speedup for graph sparsification, cut approximation and Laplacian solving. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 637–648. IEEE, 2020.

3 Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan sparsifiers. *SIAM J. Comput.*, 41(6):1704–1721, 2012. doi:10.1137/090772873.

4 Aleksandrs Belovs, Andrew M Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Time-efficient quantum walks for 3-distinctness. In *International Colloquium on Automata, Languages, and Programming*, pages 105–122. Springer, 2013.

5 András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM J. Comput.*, 44(2):290–319, 2015. doi:10.1137/070705970.

6 Nalin Bhardwaj, Antonio M. Lovett, and Bryce Sandlund. A simple algorithm for minimum cuts in near-linear time. In *Proceedings of the 17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 162 of *LIPIcs*, pages 12:1–12:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.SWAT.2020.12.

7 Arijit Bishnu, Arijit Ghosh, Gopinath Mishra, and Manaswi Paraashar. Query complexity of global minimum cut. *CoRR*, abs/2007.09202, 2020. arXiv:2007.09202.

8 Rodrigo A. Botafogo. Cluster analysis for hypertext systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 116–125, New York, NY, USA, 1993. Association for Computing Machinery. doi:10.1145/160688.160704.

9 Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Quantum computation and quantum information: A millennium volume*, 305, 2002.

10 Harry Buhrman, Richard Cleve, Ronald de Wolf, and Christof Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 358–368. IEEE, 1999.

11 Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM J. Comput.*, 35(6):1310–1328, 2006. doi:10.1137/050644719.

12 Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum. *CoRR*, quant-ph/9607014, 1996. arXiv:quant-ph/9607014.

13 Lester R. Ford and Delbert R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962. URL: http://www.jstor.org/stable/j.ctt183q0b4.

14 Wai Shing Fung, Ramesh Hariharan, Nicholas J. A. Harvey, and Debmalya Panigrahi. A general framework for graph sparsification. *SIAM J. Comput.*, 48(4):1196–1223, 2019. doi:10.1137/16M1091666.

15 Harold N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. *J. Comput. Syst. Sci.*, 50(2):259–273, 1995. doi:10.1006/jcss.1995.1022.

16 Paweł Gawrychowski, Shay Mozes, and Oren Weimann. Minimum cut in $O(m \log^2 n)$ time. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *LIPIcs*, pages 57:1–57:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.57.

17 Paweł Gawrychowski, Shay Mozes, and Oren Weimann. A note on a recent algorithm for minimum cut. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 74–79. SIAM, 2021.

18 Ralph E. Gomory and Te C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961. URL: http://www.jstor.org/stable/2098881.

19 Lov Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 78:325–328, 1997.

20 Monika R. Henzinger and Valerie King. Randomized dynamic graph algorithms with polylogarithmic time per operation. In *Proceedings of the 27th annual ACM symposium on Theory of computing (STOC)*, pages 519–527, 1995.

21 Monika R. Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. *SIAM J. Comput.*, 49(1):1–36, 2020. doi:10.1137/18M1180335.

22 Monika R. Henzinger and David P. Williamson. On the number of small cuts in a graph. *Inf. Process. Lett.*, 59(1):41–44, 1996. doi:10.1016/0020-0190(96)00079-8.

23 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 526–535. ACM, 2007. doi:10.1145/1250790.1250867.

24 David R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999. URL: http://ezproxy.lib.uts.edu.au/login?url=https://www-proquest-com.ezproxy.lib.uts.edu.au/scholarly-journals/random-sampling-cut-flow-network-design-problems/docview/212675010/se-2?accountid=17095.

25 David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, 2000. doi:10.1145/331605.331608.

26 Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic edge connectivity in near-linear time. *J. ACM*, 66(1):4:1–4:50, 2019. doi:10.1145/3274663.

27 Jason Li. Deterministic mincut in almost-linear time. In *Proceedings of the 53rd annual ACM symposium on Theory of computing (STOC)*, 2021.

28 On-Hei S. Lo, Jens M. Schmidt, and Mikkel Thorup. Compact cactus representations of all non-trivial min-cuts. *Discrete Applied Mathematics*, 2020. doi:10.1016/j.dam.2020.03.046.

29 David W. Matula. A linear time 2+epsilon approximation algorithm for edge connectivity. In Vijaya Ramachandran, editor, *Proceedings of the 4th Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms (SODA)*, pages 500–504. ACM/SIAM, 1993. URL: http://dl.acm.org/citation.cfm?id=313559.313872.

30 Sagnik Mukhopadhyay and Danupon Nanongkai. Weighted min-cut: sequential, cut-query, and streaming algorithms. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 496–509, 2020.

31 Hiroshi Nagamochi and Toshihide Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discret. Math.*, 5(1):54–66, 1992. doi:10.1137/0405004.

32 Ashwin Nayak and Felix Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the 31st annual ACM symposium on Theory of computing (STOC)*, pages 384–393, 1999.

33 Jean-Claude Picard and Maurice Queyranne. Selected applications of minimum cuts in networks. *INFOR: Information Systems and Operational Research*, 20(4):394–422, 1982. doi:10.1080/03155986.1982.11731876.

34 Ben Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 560–569. SIAM, 2011. doi:10.1137/1.9781611973082.44.

35 Aviad Rubinstein, Tselil Schramm, and S. Matthew Weinberg. Computing exact minimum cuts without knowing the graph. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 39:1–39:16. LIPICS, 2018. doi:10.4230/LIPIcs.ITCS.2018.39.

36 Robert E. Tarjan and Uzi Vishkin. Finding biconnected components and computing tree functions in logarithmic parallel time. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 12–20. IEEE, 1984.

A Karger's theorem

In this appendix we prove a slight generalization of Karger's theorem [25, Theorem 4.1] which is needed for our time-efficient algorithm. We begin by introducing some needed tools.

A.1 Tools

Matula [29] gave an $O(m/\varepsilon)$ time deterministic algorithm to compute a $(2+\varepsilon)$-approximation to the edge connectivity of a simple graph (or multigraph). The algorithm can also be adapted to give a constant factor approximation to the weight of a minimum cut in an integer-weighted graph in time $O(m\log^2(n))$, see Appendix A of [16].

▶ **Lemma 41** (Matula's approximation algorithm [29, 16]). *Let $G = (V, w)$ be an integer-weighted graph with m edges and n vertices. There is a constant c and a deterministic algorithm that in time $O(m\log^2(n))$ outputs a value $\tilde{\lambda}$ such that $\tilde{\lambda}/c \le \lambda(G) \le \lambda$.*

To efficiently construct a tree-packing we will also need to use random sampling. The following lemma is the heart of Karger's skeleton construction [24]. We recommend the presentation in [6, Lemma 14].

▶ **Lemma 42** ([24]). *Let G be an unweighted multigraph with m edges. For an integer $d \ge 2$ and real numbers ε, γ with $\varepsilon \le 1/3$, let $p = 3d(\ln n)/(\varepsilon\lambda(G))$. In time $O(pm\log(n))$ we can randomly sample $\lceil pm \rceil$ edges of G. With probability $1 - 1/n^d$ the resulting graph H has the properties that*

1. *The minimum cut of H is within a $(1+\epsilon)$ factor of $p\lambda(G) = 3d\ln(n)/\varepsilon^2$.*
2. *For every $X \subseteq V$ we have $(1-\varepsilon)w(\Delta_G(X)) \le w(\Delta_H(X)) \le (1+\varepsilon)w(\Delta_G(X))$.*

Another very useful tool we use is the Nagamochi-Ibaraki construction which shows that for an integer-weighted graph G with m edges, in time $O(m\log(n))$ one can construct a graph G' whose total edge weight is nc and which preserves all cuts of G of weight at most c.

▶ **Lemma 43** ([31]). *Let $G = (V, w)$ be an n-vertex integer-weighted graph with m edges. For any positive integer c there is a deterministic algorithm that in time $O(m\log n)$ produces an integer-weighted graph $G' = (V, w')$ with total edge weight $O(cn)$ such that for all $X \subseteq V$ with $\Delta_G(X) \le c$ it holds that $w(e) = w'(e)$ for all $e \in \Delta_G(X)$. Thus in particular $\Delta_G(X) = \Delta_{G'}(X)$ and $w(\Delta_G(X)) = w'(\Delta_{G'}(X))$ for all X with $\Delta_G(X) \le c$.*

We combine the tools of Matula's approximation algorithm, random sampling, and the sparse certificate of Nagamochi-Ibaraki into the following lemma.

▶ **Lemma 44.** *Let $G = (V, w)$ be an integer-weighted graph and let $0 < \delta < 1$ be a parameter. There is an $O(m\log^2(n) + n\log(n))$ time randomized algorithm to create a weighted graph $H = (V, w_H)$ such that*

1. *H has $O(n\log(n)/\varepsilon^2)$ edges.*
2. *The minimum cut of H has value $\lambda(H) = O(\log n)$.*
3. *If $X \subseteq V$ is such that $w(\Delta(X)) \le (1+\delta)\lambda(G)$ then $w_H(\Delta(X)) \le (1+3\delta)\lambda(H)$.*

Proof. First, by Lemma 41, in time $O(m \log^2(n))$ we can find a constant factor approximation $\tilde{\lambda}$ satisfying $\tilde{\lambda}/c \leq \lambda(G) \leq \tilde{\lambda}$. Next we apply the Nagamochi-Ibaraki algorithm to G with threshold $t = (1+\delta)\tilde{\lambda}$. In $O(m \log(n))$ time this produces an integer-weighted graph $G_2 = (V, w')$ with total edge weight $O(tn)$ such that for every $X \subseteq V$ with $w(\Delta_G(X)) \leq (1+\delta)\tilde{\lambda}$ it holds that $w(\Delta_G(X)) = w'(\Delta_G(X))$.

We now view G_2 as an unweighted multigraph with $O(tn)$ edges and apply Lemma 42. Let $p = \ln(n)/\tilde{\lambda}$. We randomly choose $\lceil pE(G_2) \rceil = O(n \ln(n))$ edges of G_2 and let the resulting graph be H. This can be done in time $O(n \log(n))$. By Lemma 42 the graph has the stated properties. The total running time is $O(m \log^2(n) + n \log(n))$. ◀

A.2 Tree packing

With these preliminaries in place we now turn to actually constructing a tree packing. We first need the definition, and a lemma of Karger.

▶ **Definition 45** (Weighted tree packing). *Let $G = (V, w)$ be an integer-weighted graph. A* weighted tree packing *is a set of spanning trees of G, each with an assigned weight, such that the total weight of trees containing any edge $e \in E(G)$ is at most $w(e)$. The* value *of the packing is the total weight of trees in it.*

▶ **Lemma 46** ([25, Lemma 2.3]). *Given a weighted tree packing of value βc and a cut of value αc, at least a $(3 - \alpha/\beta)/2$ fraction of the trees by weight 2-constrain the cut.*

Gabow gives an algorithm to construct a near optimal tree packing in an unweighted multigraph. The following is an easy adaptation to an integer-weighted graph.

▶ **Lemma 47** ([15]). *Let $G = (V, w)$ be an integer-weighted graph with n vertices and m edges. There is a deterministic algorithm that finds an integer-weighted tree packing of G of value at least $\lambda(G)/2$ in time $O(m(\lambda(G)^2 \log(n) + \log^2(n)))$.*

Proof. For a multigraph H with n vertices and m' edges, Gabow [15] gives a deterministic algorithm that finds a tree packing of weight $\lambda(H)/2$ in time $m'\lambda(H) \log(n)$. The only difference with our case is that G is an integer-weighted graph instead of a multigraph. We can of course view G as a multigraph but it becomes too expensive to run Gabow's algorithm if this significantly blows up the number of edges.

Thus we first use Lemma 41 to compute $\tilde{\lambda}$ such that $\tilde{\lambda}/c \leq \lambda(G) \leq \tilde{\lambda}$ in time $O(m \log^2(n))$. Then we make a pass through the edges of G and form a graph G' where any edge of weight larger than $\tilde{\lambda}$ in G is thresholded down to $\tilde{\lambda}$. Thus when viewed as a multigraph G' will only have $O(m\lambda(G))$ edges. Any tree packing of G is also a tree packing of G' as the value of any tree packing is at most $\lambda(G) \leq \tilde{\lambda}$. We can then apply Gabow's algorithm to G' to obtain the theorem. ◀

We are finally ready to prove the slight generalization of Karger's theorem that we require.

▶ **Theorem 24** ([25, Theorem 4.1]). *Let $G = (V, w)$ be a weighted graph with n vertices and m edges. There is a randomized algorithm that in time $O(m \log^2(n) + n \log^4(n))$ time constructs a set of $O(\log n)$ spanning trees such that every $(1 + 1/16)$-near minimum cut of G 2-respects $1/4$ of them with high probability.*

Proof. In $O(m)$ time we can find the minimum weight α of an edge of G. Multiplying all edge weights by $1/\alpha$ we obtain a graph where all edge weights are at least 1 and that has the same set of $(1+1/16)$-near minimum cuts as G. Thus without loss of generality now assume that G has all edge weights at least 1.

In $O(m)$ time we create the integer-weighted graph $G' = (V, w')$ where $w'(e) = \lfloor 100w(e) \rceil$. Note that as we assume that every edge of G has weight at least 1, for any $X \subseteq V$ we have

$$0.995w(\Delta_G(X)) \leq \frac{w(\Delta_{G'}(X))}{100} \leq 1.005w(\Delta_G(X)) \ . \tag{1}$$

Thus if $\Delta_G(X)$ is a $(1+\varepsilon)$-near minimum cut of G then $\Delta_{G'}(X)$ is a $(1+\varepsilon)(1.005)^2$-near minimum cut of G'. With $\varepsilon = 1/16$ it follows that $\Delta_{G'}(X)$ is a $1+1/12$-near minimum cut of G'.

Next we apply Lemma 44 to G' to in time $O((m+n)\log^2(n))$ create a graph H with the properties specified there. We then use Lemma 47 to find a tree packing of weight at least $\lambda(H)/2$ and which contains $O(\log(n))$ trees since $\lambda(H) = O(\log(n))$. Now let $\Delta_G(X)$ be a $(1+1/16)$-near minimum cut of G. Then $\Delta_{G'}(X)$ is a $(1+1/12)$-near mincut of G' and by Lemma 44, $\Delta_H(X)$ is a $1+1/4$-near mincut of H. Therefore by Lemma 46 at least $1/4$ of the trees in the packing will 2-respect $\Delta_H(X)$. These trees must also 2-respect $\Delta_G(X)$ since it has the same shore X. ◀

On the Complexity of Evaluating Highest Weight Vectors

Markus Bläser
Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

Julian Dörfler
Saarbrücken Graduate School of Computer Science, Saarland Informatics Campus, Germany

Christian Ikenmeyer[1]
University of Liverpool, UK

Abstract

Geometric complexity theory (GCT) is an approach towards separating algebraic complexity classes through algebraic geometry and representation theory. Originally Mulmuley and Sohoni proposed (SIAM J Comput 2001, 2008) to use occurrence obstructions to prove Valiant's determinant vs permanent conjecture, but recently Bürgisser, Ikenmeyer, and Panova (Journal of the AMS 2019) proved this impossible. However, fundamental theorems of algebraic geometry and representation theory grant that every lower bound in GCT can be proved by the use of so-called highest weight vectors (HWVs). In the setting of interest in GCT (namely in the setting of polynomials) we prove the NP-hardness of the evaluation of HWVs in general, and we give efficient algorithms if the treewidth of the corresponding Young-tableau is small, where the point of evaluation is concisely encoded as a noncommutative algebraic branching program! In particular, this gives a large new class of separating functions that can be efficiently evaluated at points with low (border) Waring rank. As a structural side result we prove that border Waring rank is bounded from above by the ABP width complexity.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Mathematics of computing → Mathematical software

Keywords and phrases Algebraic complexity theory, geometric complexity theory, algebraic branching program, Waring rank, border Waring rank, representation theory, highest weight vector, treewidth

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.29

Related Version *Previous Version*: https://arxiv.org/abs/2002.11594

Funding *Christian Ikenmeyer*: DFG grant IK 116/2-1.

1 Introduction

Geometric complexity theory (GCT) is an approach towards the separation of algebraic complexity classes using algebraic geometry and representation theory [45, 46, 17]. Let $\mathrm{per}_i := \sum_{\pi \in \mathfrak{S}_i} \prod_{j=1}^{i} x_{j,\pi(j)}$ be the permanent polynomial. Valiant asked for the smallest size of a matrix A whose entries are affine linear polynomials such that $\det(A) = \mathrm{per}_i$ and his famous VBP $\neq$ VNP conjecture (also known as the "determinant vs permanent conjecture") states that this size is not polynomially bounded. Mulmuley and Sohoni strengthened the conjecture by allowing per_i to be approximated arbitrarily closely, i.e., $\mathrm{VNP} \not\subseteq \overline{\mathrm{VBP}}$. This question can be attacked with GCT.

In the GCT approach, we set $m := d^2$ and let the group $\mathsf{GL}_m := \mathsf{GL}(\mathbb{C}^m)$ act on a the space of homogeneous degree d polynomials in m variables by linear transformation of the variables. The Mulmuley–Sohoni conjecture can be rephrased as "eventually $x_{11}^{d-i}\mathrm{per}_i \notin \overline{\mathsf{GL}_m \det_d}$" if

[1] part of this research was done when CI was at the Max Planck Institute for Software Systems, Germany, and the Simons Institute for the Theory of Computing, United States

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 29; pp. 29:1–29:36
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

d grows polynomially in i. Now we try to attack this problem by representation theoretic methods, so-called *obstructions*. A first crucial insight is that $x_{11}^{d-i}\mathrm{per}_i \in \overline{\mathsf{GL}_m \det_d}$ iff $\overline{\mathsf{GL}_m(x_{11}^{d-i}\mathrm{per}_i)} \subseteq \overline{\mathsf{GL}_m \det_d}$. Thus, we compare two varieties and we want to disprove that the orbit closure of the padded permanent is contained in the orbit closure of the determinant for polynomially large d. To to so, an important object to study are so-called *highest weight vectors* (HWVs) of weight $\lambda \in \mathbb{N}^m$. They are homogeneous degree n polynomials in the coefficients of homogeneous degree d polynomials in m variables, satisfying two properties (see Sec. 5). Their dimension is called the plethysm coefficient. The dimension of their restriction to a GL_m-variety X is called the *multiplicity* $\mathsf{mult}_\lambda\mathbb{C}[X]$ *of* λ *in the coordinate ring* $\mathbb{C}[X]$ [2]. They are important, because if $\mathsf{mult}_\lambda\mathbb{C}[X] > \mathsf{mult}_\lambda\mathbb{C}[Y]$, then Schur's lemma implies that $X \not\subseteq Y$. In this case, λ is called a *multiplicity obstruction*. If additionally $\mathsf{mult}_\lambda\mathbb{C}[X] > 0 = \mathsf{mult}_\lambda\mathbb{C}[Y]$, then λ is called an *occurrence obstruction*. Even more fundamentally, the properties of the representation theory of GL_m imply that if $x_{11}^{d-i}\mathrm{per}_i \notin \overline{\mathsf{GL}_m \det_d}$, then there exists a HWV f such that $f(\overline{\mathsf{GL}_m \det_d}) = \{0\}$ and for a random $g \in \mathsf{GL}_m$ we have $f(g(x_{11}^{d-i}\mathrm{per}_i)) \neq 0$. So this separation is always provable by HWVs. This follows from the fact that HWVs uniquely classify the irreducible representations of GL_m.

Bürgisser et al. [16] proved that occurrence obstructions are not sufficient to prove Mulmuley and Sohoni's conjecture. Hence, multiplicity obstructions are a focus of recent research [25, 36]. To compute multiplicities, it is import to understand the complexity of the evaluation of highest weight vectors.

To calculate a multiplicity $\mathsf{mult}_\lambda\mathbb{C}[X]$, a common approach is to generate a basis of all HWVs of weight λ and evaluate them at enough points from X (points from all GL_m-varieties in GCT are efficiently samplable) and observe the dimension of their linear span, which equals $\mathsf{mult}_\lambda\mathbb{C}[X]$. For this to work, one needs an algorithm to evaluate HWVs at points. An evaluation algorithm is even more important to make the following approach work: We know that if $X \not\subseteq Y$, then there exists a HWV f of some weight λ such that $f(Y) = \{0\}$ and $f(x) \neq 0$ for almost all points $x \in X$ [8, Cor. 11.4.2]. This evaluation is a challenging problem in algebraic geometry that is related to deep combinatorics, see [40, 20, 2].

The reader unfamiliar with Young tableaux or highest weight vectors finds their definitions at the beginning of Section 5, see also [16, §4].

2 Our contributions

To our best knowledge, we systematically study the complexity of evaluating highest weight vectors for the first time. In Section 5 we first present a known combinatorial method of exactly evaluating HWVs without expanding all the monomials explicitly which has been used to to evaluate HWVs at points of small Waring rank as in [2, 15]. Additionally there have been attempts to improve the running time for evaluating at products of linear forms – the so called Chow variety – via dynamic programming [25]. We generalize both approaches in Section 6 to allow evaluation on all points with partial derivative spaces of small dimension, i.e., small noncommutative algebraic branching program width complexity.

▶ **6.2 Theorem (informal).** *The evaluation of a degree n highest weight vector $f_{\hat{T}}$ (given by a Young tableau $\hat{T}$ with r rows) at a homogeneous degree d polynomial p in m variables whose noncommutative ABP width complexity is at most w can be computed in time $O(w^{n+r}\,\mathrm{poly}(n,d,m))$.*

[2] This defines $\mathsf{mult}_\lambda\mathbb{C}[X]$ without defining $\mathbb{C}[X]$. The coordinate ring $\mathbb{C}[X]$ is the polynomial ring quotiened by the ideal of polynomials vanishing on X.

In particular, by Theorem 4.2 this includes for the first time all points of small border Waring rank:

▶ **4.2 Theorem (informal).** *For all polynomials f the noncommutative ABP width of f is less or equal to the border Waring rank of f. This also holds for commutative ABP width complexity.*

Theorem 4.2 is proved using the noncommutative algebraic branching program width complexity as a tool, which shows that it is not just a notion useful for algorithmic purposes, but a natural notion of independent interest. Note that our algorithms are particularly useful, because the noncommutative algebraic branching program width complexity can be determined in polynomial time, whereas determining the Waring rank of a polynomial is NP-hard, even when it is given explicitly as a list of coefficients, see [54].

A HWV can be encoded as a linear combination of Young tableaux, see e.g. [49, §3.9] or [34, Sec. 4.3]. All current evaluation algorithms have a running time exponentially dependent on the size of the Young tableau. We improve this in Section 7 and establish an algorithm that only depends exponentially on the treewidth of the Young tableau:

▶ **7.2 Theorem (informal).** *The evaluation of a degree n highest weight vector $f_{\hat{T}}$ given by a Young tableau $\hat{T}$ at a homogeneous degree d polynomial p in m variables with noncommutative ABP width complexity w can be computed in time $w^{\omega(\tau+1)}\,\mathrm{poly}(n, d, m, |\mathcal{T}|)$, where $\mathcal{T}$ is a tree decomposition of $\hat{T}$ of width τ and size $|\mathcal{T}|$ and ω is the matrix multiplication exponent.*

Our paper is the first that formally connects the running time of algorithms in representation theory with a graph parameter. An implementation of the algorithm in Theorem 7.2 might make it possible to compute the multiplicities for examples that were out of reach before, which is potentially useful for implementing the geometric complexity theory approach.

Lastly we show in Section 8 that this dependency is basically optimal as we show two lower bounds under the exponential time hypothesis. A lower bound of $2^{o(n)}$ for the vanishing evaluation decision problem when the HWV $f \in \mathrm{Sym}^n\mathrm{Sym}^d V$ is given by an arbitrary two row Young tableau and a lower bound of $2^{o(\sqrt{n})}$ when it is given by a semistandard Young tableau. Additionally we show NP-hardness for both versions of the decision problem and even #P-hardness for exact evaluations.

▶ **8.1 Theorem (informal)** (HWVs from two-row tableaux)**.** *Deciding whether a degree n highest weight vector $f_{\hat{T}}$ (given by a two-row Young tableau $\hat{T}$) evaluates to zero at a point of constant degree at least 8 and of Waring rank 3 is NP-hard. Assuming ETH no $2^{o(n)}$ algorithm for this evaluation can exist.*

▶ **8.9 Theorem (informal)** (HWVs from semistandard tableaux)**.** *Deciding whether or not the evaluation of a degree n highest weight vector $f_{\hat{T}}$ (given by a 5-row semistandard Young tableau $\hat{T}$) vanishes at a point of constant degree $d \geq 16$ with $16 \mid d$ and of Waring rank 5 is NP-hard. Additionally this evaluation can not be computed in time $2^{o(\sqrt{n})}$ unless ETH fails.*

▶ **8.2 Theorem (informal)** (#P-hardness)**.** *Evaluating a highest weight vector $f_{\hat{T}}$ (given by a two-row Young tableau $\hat{T}$) at a point of Waring rank 3 and degree $d \geq 18$ is #P-hard.*

We remark that it is quite surprising that these results can be obtained using points of small constant Waring rank.

3 Related work

The approach to lower bounds via evaluating HWVs was used in [13, 14] in the tensor setting to obtain lower bounds on the border rank of matrix multiplication. This also led to multiplicity obstructions (even occurrence obstructions). Our complexity results can be interpreted as limitations on how far such an approach via explicit evaluations can be pushed.

Combinatorics on tableaux for describing highest weight vectors has a rich history dating back to the early invariant theory. This tableau calculus is equivalent to the classical *Feynman diagram calculus* explained in [1], see also [49]. Highest weight vectors of a GL_m-representation W are also called *covariants*, since they correspond to the invariants of $W \otimes (S_\lambda \mathbb{C}^m)^*$, see e.g. [48, Def. 3.9]. Recently, these methods have been applied in various areas, see [39, 3, 50, 24, 44, 2, 15, 21], to name a few. If we restrict ourselves to two-row Young diagrams, then inheritance principles from representation theory [34, Sec. 5.3] let us replace V with $\mathbb{C}^2$. Then $\mathrm{Sym}^d\mathbb{C}^2$ is the space of homogeneous degree d polynomials in 2 variables. This is the Hilbert space corresponding to a system of d indistinguishable photons distributed among two modes, which is used in the study of 2-mode linear optical circuits on d indistinguishable particles.

Waring rank and border Waring rank are classical notions studied in algebraic geometry in the language of higher secant varieties [41]. More generally, border complexity is classically studied in algebraic geometry, see [42]. Bini et al [7] (see also [6]) used it in their construction of fast matrix multiplication algorithms. Studying border complexity in algebraic circuit complexity started with [11, 45] and recently caught momentum [31, 9, 38].

Kronecker coefficients and plethysm coefficients are the dimensions of specific highest weight vector spaces. Algorithms for their computation or theorems about their positivity and value that depend heavily on the shape of the input Young tableau have a long history. For example, if the number of rows of all parameters is constant, then the Kronecker coefficient can be computed in polynomial time [22]. A similar statement is true for plethysm coefficients, see [27]. The software `LiE` [43] performs all representation theoretic computations with a fixed number of rows. In [35], positivity of Kronecker coefficients depends on comparing Young diagrams with respect to the dominance order, and in [4] the main parameter is the so-called *Durfee size* of the Young diagram, which is the side length of largest square that can be embedded into the Young diagram, see also the very recent [5]. The shape of the Young diagram also plays a crucial role in the recent breakthrough proof of Stembridge's stability conjecture [52]. For two-row Young diagrams much additional structure is known, for example Hermite's classical reciprocity law for plethysm coefficients [32], which makes our lower bound for two-row Young tableaux quite surprising.

Treewidth has been intensely studied by Robertson and Seymour and has been applied numerous times to construct faster graph algorithms for cases where the treewidth is bounded by a function $o(n)$, most notably some algorithms for NP-hard problems restricted to planar graphs, for example 3-coloring. See [23] for an introduction to treewidth algorithms.

4 Border Waring rank and Algebraic Branching Programs

In this section we introduce noncommutative ABP width complexity for polynomials and use it to prove Theorem 4.2. Noncommutative ABP width complexity will play a central role in Sections 6 and 7.

An algebraic branching program (ABP) is a layered directed acyclic graph (the vertex set is partitioned into numbered layers and edges only go from the i-th layer to the $(i+1)$-th layer) with two distinguished nodes, the *source* and the *sink*, and the edges are labeled with homogeneous linear polynomials. The weight $w(P)$ of a path P with edge labels $\ell_1, \dots, \ell_d$ is defined as the product $w(P) := \ell_1 \cdots \ell_d$. We say that the ABP *computes* the sum $\sum_{\text{source-sink-path } P} w(P)$. We can view the same ABP both over commuting variables or noncommuting variables. If we interpret it over noncommuting variables, we call it an ncABP. If we want to stress that the variables commute, we call it a cABP. The size of an ABP is the number of its vertices. The width of an ABP is the largest number of vertices in any layer. For a homogeneous degree d polynomial let the *ABP width complexity* $\mathsf{w}(f)$ be defined as the smallest width of a cABP computing f. A sequence (f_n) of polynomials is called a *p-family* if the number of variables and the degree of each f_n are polynomially bounded in n. p-families are the object of study in Valiant's algebraic complexity framework. Let VBP denote the set of all p-families (f_i) with polynomially bounded ABP width complexity $\mathsf{w}(f_i)$.

The Waring rank $\mathsf{WR}(f)$ of a homogeneous degree d polynomial is the smallest r such that f can be written as a sum of r powers of homogeneous linear polynomials. Let $\mathsf{VWaring}$ be the set of all p-families with polynomially bounded Waring rank.

Clearly, $\mathsf{w}(f) \le \mathsf{WR}(f)$, because from a Waring rank r decomposition we can construct a width r cABP that computes f in the straightforward way: The cABP contains exactly r disjoint source-sink-paths (vertex-disjoint up to source and sink) so that on each path all edges have the same label. Therefore $\mathsf{VWaring} \subseteq \mathsf{VBP}$.

There is a natural way to associate to every algebraic complexity measure a corresponding border complexity measure: We define the *border Waring rank* $\underline{\mathsf{WR}}(f)$ as the smallest r such that f can be approximated arbitrarily closely (coefficient-wise) by polynomials with $\mathsf{WR}(f) \le r$, or equivalently, the smallest r such that f lies in the closure (Zariski closure and Euclidean closure coincide) of the set $\{f \mid \mathsf{WR}(f) \le r\}$. Clearly $\underline{\mathsf{WR}}(f) \le \mathsf{WR}(f)$. Let $\overline{\mathsf{VWaring}}$ denote the set of sequences of polynomials with polynomially bounded border Waring rank. Clearly $\mathsf{VWaring} \subseteq \overline{\mathsf{VWaring}}$.

Analogously we can define the *border ABP width complexity* $\underline{\mathsf{w}}(f)$ from w. Clearly $\underline{\mathsf{w}}(f) \le \mathsf{w}(f)$. Let $\overline{\mathsf{VBP}}$ be the set of polynomials with polynomially bounded border ABP width complexity. Clearly $\mathsf{VBP} \subseteq \overline{\mathsf{VBP}}$.

For noncommutative polynomials we define the analogous versions ncw and $\underline{\mathsf{ncw}}$. It follows from Nisan's work [47] that $\mathsf{ncw}(f) = \underline{\mathsf{ncw}}(f)$.

In general, it is unknown by how much an algebraic complexity class grows when applying the closure. In particular, it is open whether $\mathsf{VWaring} = \overline{\mathsf{VWaring}}$ or whether $\mathsf{VBP} = \overline{\mathsf{VBP}}$. But the following result in this direction is known.

▶ **Theorem 4.1.** $\overline{\mathsf{VWaring}} \subseteq \mathsf{VBP}$.

The rest of this section is used to sketch the standard proof of Theorem 4.1, then to introduce the notion of ncABPs computing polynomials, and then to prove our sharp version of Theorem 4.1, Theorem 4.2, with a very short proof. This shows how natural the concepts are that we introduce. The standard proof of Theorem 4.1 is just given for comparison and is not needed in the rest of this paper.

We quickly sketch the standard proof of Theorem 4.1. We will need the following concept only for this proof. A *read-once oblivious ABP* is a layered ABP whose edge labels have univariate polynomials in x_i on each edge in layer i. The first step in the proof is Saxena's duality trick [53, Lemma 1]:

> If $f \in \mathbb{C}[x_1, \dots, x_m]_d$ has $\mathsf{WR}(f) \le s$, then there is a read-once oblivious ABP computing f with width at most $s \cdot (md + d + 1)$.

The proof uses a power series argument. The next crucial step is to use a variant of Nisan's result [47] to see that the *border* read-once oblivious ABP width equals the read-once oblivious ABP width, so approximations can be removed [29, Sec. 4.5.2]:

> If $f \in \mathbb{C}[x_1, \ldots, x_m]_d$ has $\underline{\mathsf{WR}}(f) \leq s$, then there is an read-once oblivious ABP computing f with width at most $s \cdot (md + d + 1)$.

We can unfold this read-once oblivious ABP, i.e., replace each edge (remember, each label is a univariate degree $\leq d$ polynomial) with a (non-layered) ABP computing it, where each edge has an affine linear label. If done properly, this requires $d-1$ additional vertices per edge. Making the ABP layered and homogeneous blows up the ABP's width by a factor of $d+1$. We conclude:

$$\text{For all } f \in \mathbb{C}[x_1, \ldots, x_m]_d \text{ we have } \mathsf{w}(f) \leq \underline{\mathsf{WR}}(f) \cdot (md + d + 1) \cdot (d + 1). \tag{4.1}$$

Eq. (4.1) proves Theorem 4.1 when we assume that m and d are polynomially bounded (which is usually assumed). We now strengthen eq. (4.1) with the following clean statement that is independent of m and d.

▶ **Theorem 4.2.** *For all $f \in \mathbb{C}[x_1, \ldots, x_m]_d$ we have $\mathsf{w}(f) \leq \underline{\mathsf{WR}}(f)$.*

In fact, we prove $\mathsf{w}(f) \leq \mathsf{ncw}(f) \leq \underline{\mathsf{WR}}(f)$, but we have not yet defined what we mean by an ncABP computing a polynomial. The rest of Section 4 is devoted to the proof of Theorem 4.2 and to this definition. We start with introducing several main multilinear algebra concepts of this paper. The actual proof of Theorem 4.2 is then very short and natural.

When talking about homogeneous multivariate noncommutative polynomials, we use the standard language of multilinear algebra: An order d *tensor* in $\otimes^d \mathbb{C}^m$ is a d-dimensional $m \times m \times \cdots \times m$ array of numbers. There is a canonical vector space isomorphism between the vector space of m-variate homogeneous degree d noncommutative polynomials $\mathbb{C}\langle x_1, \ldots, x_m \rangle_d$ and $\otimes^d \mathbb{C}^m$, which is defined on monomials as

$$x_{i_1} x_{i_2} \cdots x_{i_d} \xrightarrow{\sim} E_{i_1, \ldots, i_d},$$

where $E_{i_1,\ldots,i_d}$ is the tensor that is 0 everywhere, but has a single 1 at position $(i_1, \ldots, i_d)$. Let (e_i) be the standard basis of $\mathbb{C}^m$. We use the notation $e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_d} := E_{i_1,\ldots,i_d}$. More generally, for $v_1, \ldots, v_d \in \mathbb{C}^m$, we write $v_1 \otimes v_2 \otimes \cdots \otimes v_m$ to be the tensor whose entry at position $(i_1, \ldots, i_d)$ is the product $(v_1)_{i_1} \cdot (v_2)_{i_2} \cdots (v_d)_{i_d}$.

A tensor T is called *symmetric* if $T_{i_1,\ldots,i_d} = T_{i_{\pi(1)},\ldots,i_{\pi(d)}}$ for all permutations $\pi \in \mathfrak{S}_d$. Let $\mathrm{Sym}^d \mathbb{C}^m \subseteq \otimes^d \mathbb{C}^m$ denote the linear subspace of symmetric tensors. There is a canonical vector space isomorphism between the vector space of m-variate homogeneous degree d *commutative* polynomials $\mathbb{C}[x_1, \ldots, x_m]_d$ and $\mathrm{Sym}^d \mathbb{C}^m$, which is defined on monomials as

$$x_{i_1} x_{i_2} \cdots x_{i_d} \xrightarrow{\sim} \sum_{\pi \in \mathfrak{S}_d} \tfrac{1}{d!} E_{\pi(i_1),\ldots,\pi(i_d)},$$

For example, the polynomial $x_1^2 x_2$ corresponds to the tensor $\frac{1}{3}(e_1 \otimes e_1 \otimes e_2 + e_1 \otimes e_2 \otimes e_1 + e_2 \otimes e_1 \otimes e_1)$. [3] We use e_i and x_i interchangeably.

It is crucial to note that *noncommutative ABPs can compute symmetric tensors.* An example is given in Figure 1, where we used $x := x_1$ and $y := x_2$. As before with cABPs, it is easy to see that every Waring rank r decomposition of f can be converted into a width

[3] This tensor is called the W-state in quantum information theory.

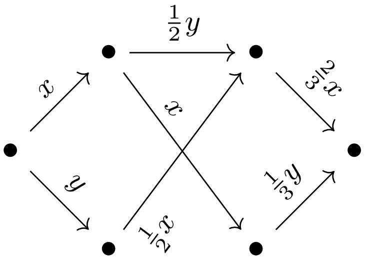


Figure 1 An ncABP computing the symmetric tensor $\frac{1}{3}(x \otimes x \otimes y + x \otimes y \otimes x + y \otimes x \otimes x)$, which corresponds to the polynomial x^2y. If we reinterpret the ncABP as a cABP, it computes $\frac{1}{3}(xxy + xyx + yxx) = x^2y$. Such ncABPs can be efficiently constructed using Nisan's construction technique [47]. Interestingly, in this example the width is only 2, while the Waring rank of x^2y is 3.

r ncABP computing f in the straightforward way: The ncABP contains exactly r disjoint source-sink-paths (vertex-disjoint up to source and sink) so that on each path all edges have the same label. Every ncABP can be reinterpreted as a cABP by letting the variables commute. If the ncABP computes a symmetric tensor, then clearly this cABP computes the corresponding polynomial. Now we can prove Theorem 4.2 in a very natural and short way as follows.

Given f with a border Waring rank s decomposition. We construct the corresponding border ncABP with s many edge-disjoint source-sink-paths, so $\underline{\mathsf{ncw}}(f) \leq s$. Using Nisan's result [47] that $\underline{\mathsf{ncw}} = \mathsf{ncw}$, it follows $\mathsf{ncw}(f) \leq s$. This gives a width s ncABP that computes f. Reinterpreting this ncABP as a cABP finishes our proof of Theorem 4.2.

5 Highest Weight Vectors and their combinatorial evaluation

Let $V = \mathbb{C}^m$ be a finite dimensional complex vector space with standard basis $e_1, e_2, \ldots, e_m$. There is a canonical action of $g \in \mathsf{GL}(V)$ on the tensor power $\otimes^d V$ via $g(p_1 \otimes \cdots \otimes p_d) := (gp_1) \otimes \cdots \otimes (gp_d)$ and linear continuation. This action can be lifted to a linear action on $\mathrm{Sym}^n \otimes^d V$ via

$$(gf)(p) := f(g^t p) \text{ for } f \in \mathrm{Sym}^n \otimes^d V \text{ and } p \in \otimes^d V$$

Note that this makes $\mathrm{Sym}^n \otimes^d V$ a $\mathsf{GL}(V)$-representation. We denote by $\mathrm{Sym}^d V \subseteq \otimes^d V$ the vector space of symmetric tensors over V of order d and by $p_1 \odot \cdots \odot p_d := \sum_{\pi \in \mathfrak{S}_d} \frac{1}{d!} p_{\pi(1)} \otimes \cdots \otimes p_{\pi(d)}$ the symmetric tensor product of $p_1, \ldots, p_d \in V$. The linear subspace $\mathrm{Sym}^d V \subseteq \otimes^d V$ is closed under the action of $\mathsf{GL}(V)$. This action can be lifted to a linear action on $\mathrm{Sym}^n \mathrm{Sym}^d V$ via

$$(gf)(p) = f(g^t p) \text{ for } f \in \mathrm{Sym}^n \mathrm{Sym}^d V \text{ and } p \in \mathrm{Sym}^d V$$

Note that this makes $\mathrm{Sym}^n \mathrm{Sym}^d V$ a $\mathsf{GL}(V)$-representation.

We call a sequence $\lambda = (\lambda_1, \lambda_2, \ldots)$ a *partition of* $N \in \mathbb{N}$ if $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq 0$ and $\sum_{i \geq 1} \lambda_i = N$. In our case we will usually have $N = nd$. We denote the transpose partition λ^t by μ and define it as $\mu_i = |\{j \mid \lambda_j \geq i\}|$. Note that μ is also a partition of N. We will write partitions as finite sequences and omit all the trailing zeros.

For any GL_m representation W, a *highest weight vector* $f \in W$ of type λ is a vector that satisfies

1. f is invariant under the action of any $g \in \mathsf{GL}_m$ when g is upper triangular with 1s on the diagonal.
2. $\mathrm{diag}(\alpha_1, \dots, \alpha_m)f = \alpha_1^{\lambda_1} \cdot \dots \cdot \alpha_m^{\lambda_m} f$ where $\mathrm{diag}(\alpha_1, \dots, \alpha_m)$ is the diagonal matrix with $\alpha_1, \dots, \alpha_m \in \mathbb{C}$ on the diagonal.

The highest weight vectors of type λ form a vector space which we call $\mathsf{HWV}_\lambda(W)$. We denote by $\mathsf{HWV}(W)$ the vector space of all HWVs in W without any weight restriction.

The smallest example is the discriminant polynomial $b^2 - 4ac$ in $\mathrm{Sym}^2\mathrm{Sym}^2\mathbb{C}^2$, see [8, Exa. 9.1.4] for which we have $g(b^2 - 4ac) = \det(g)^2(b^2 - 4ac)$.

We first derive a combinatorial description of the evaluation of highest weight vectors. We follow [20, 15].

We can describe the highest weight vectors of $\mathrm{Sym}^n\mathrm{Sym}^dV$ in terms of so-called Young tableaux (see also [49, §3.9], [34], and [16, §4]).

▶ **Definition 5.1.** *A* Young tableau *T of shape $\lambda = (\lambda_1, \dots, \lambda_r)$ where λ is a partition is a left justified array of boxes where row i contains λ_i boxes and each box contains a positive integer. If the tableau contains the numbers 1 through n each d times it is said to have* (rectangular) content *$n \times d$, for example $\begin{array}{|c|c|c|c|}\hline 1&2&3&1\\ \hline 2&3\\ \cline{1-2}\end{array}$ has content 3×2. A Young tableaux is said to be* semistandard *if the entries are strictly increasing in each column and non-decreasing in each row, for example $\begin{array}{|c|c|c|c|}\hline 1&1&2&3\\ \hline 2&3\\ \cline{1-2}\end{array}$ is semistandard, while $\begin{array}{|c|c|c|c|}\hline 1&2&3&1\\ \hline 2&3\\ \cline{1-2}\end{array}$ is not. A Young tableaux is said to be* standard *if the entries are strictly increasing in each column and row and every entry occurs exactly once. For example, $\begin{array}{|c|c|c|c|c|c|}\hline 1&3&4&6&7&8\\ \hline 2&5&9\\ \cline{1-3}\end{array}$ is standard.*

Fix a tableau T of shape λ with content $(nd) \times 1$ and fix a tensor $p = \sum_{i=1}^r \ell_{i,1} \otimes \cdots \otimes \ell_{i,d} \in \otimes^d\mathbb{C}^m$. We use arithmetic modulo d with the system of representatives $\{1, \dots, d\}$, so $a \bmod d \in \{1, \dots, d\}$. Each of the sets $\{1, \dots, d\}, \{d+1, \dots, 2d\}, \dots$ is called a *block*. We define $k(a) := \lceil a/d \rceil$. We define $j(a) := a \mod d$, which gives the position of the element a in its block. A placement

$$\vartheta : \{1, \dots, nd\} \to \{(i, j) \mid 1 \le i \le r, 1 \le j \le d\}$$

is called *proper* if there is a map $\varphi : \{1, \dots, n\} \to \{1, \dots, r\}$ such that $\vartheta(a) = (\varphi(k(a)), j(a))$. Every ϑ induces a map

$$\vartheta' : \{1, \dots, nd\} \to \{\ell_{i,j} \mid 1 \le i \le r, 1 \le j \le d\}, \quad \vartheta'(a) := \ell_{\vartheta(a)}.$$

We define the determinant of a matrix that has more rows than columns as the determinant of its largest top square submatrix.

We define the polynomial f_T via its evaluation on p:

$$f_T(p) := \sum_{\text{proper } \vartheta} \prod_{c=1}^{\lambda_1} \det{}_{\vartheta,c} \quad \text{with} \quad \det{}_{\vartheta,c} := \det\left(\vartheta'(T(1,c)) \dots \vartheta'(T(\mu_c, c))\right) \tag{5.1}$$

Pictorially φ chooses one of the rank 1 tensors for each block of d numbers and places those onto T. Then we take the product of the columnwise determinants. The evaluation $f(p)$ is now the sum over all possible choices.

All our algorithms for efficient evaluation of HWVs in this paper rely on equation (5.1). We illustrate equation (5.1) with an example. Let $n = 3$, $d = 2$, $T = \begin{array}{|c|c|c|c|}\hline 1&2&3&5\\ \hline 4&6\\ \cline{1-2}\end{array}$, and let $r = 2$, $\ell_{1,1} = e_1$, $\ell_{2,1} = e_1$, $\ell_{1,2} = e_1$, $\ell_{2,2} = e_1 + e_2$, where $e_1 = \binom{1}{0}$ and $e_2 = \binom{0}{1}$. Hence $p = 2e_1 \otimes e_1 + e_1 \otimes e_2$. The three blocks are $\{1, 2\}$, $\{3, 4\}$, and $\{5, 6\}$. The first few proper placements $\vartheta : \{1, \dots, 6\} \to \{(1, 1), (1, 2), (2, 1), (2, 2)\}$, their corresponding maps $\varphi : \{1, 2, 3\} \to \{1, 2\}$, and the determinant calculations are as follows:

proper ϑ	φ	summand in eq. (5.1)
$\big((1,1),(1,2),(1,1),(1,2),(1,1),(1,2)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((1,2),(1,1),(1,1),(1,2),(1,1),(1,2)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((1,1),(1,2),(1,2),(1,1),(1,1),(1,2)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((1,2),(1,1),(1,2),(1,1),(1,1),(1,2)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((1,1),(1,2),(1,1),(1,2),(1,2),(1,1)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((1,2),(1,1),(1,1),(1,2),(1,2),(1,1)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((1,1),(1,2),(1,2),(1,1),(1,2),(1,1)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((1,2),(1,1),(1,2),(1,1),(1,2),(1,1)\big)$	(1,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((2,1),(2,2),(1,1),(1,2),(1,1),(1,2)\big)$	(2,1,1)	$\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\big((2,2),(2,1),(1,1),(1,2),(1,1),(1,2)\big)$	(2,1,1)	$\det\left(\begin{smallmatrix}1&1\\1&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)\cdot\det\left(\begin{smallmatrix}1\end{smallmatrix}\right)=0$
$\vdots$		

It is a classical result from multilinear algebra that this construction yields a well-defined polynomial of weight λ on $\otimes^d\mathbb{C}^m$. If T is the column-standard tableau, then $f_T \in \mathsf{HWV}_\lambda(\mathrm{Sym}^n \otimes^d \mathbb{C}^m)$ is not hard to verify. Schur-Weyl duality states that $\otimes^n \otimes^d \mathbb{C}^m = \bigoplus_\lambda S_\lambda(V) \otimes [\lambda]$, where the sum goes over all partitions λ of nd into at most m parts, and where $S_\lambda(V)$ is the irreducible GL_m-representation of type λ (called the Schur module) and $[\lambda]$ is the irreducible $\mathfrak{S}_{dn}$-representation of type λ (called the Specht module). Since a basis of $[\lambda]$ is given by the standard tableaux of shape λ, this immediately implies that

$$\mathsf{HWV}_\lambda(\mathrm{Sym}^n \otimes^d \mathbb{C}^m) \text{ is the linear span of the } f_T, \text{ where } T \text{ is standard of shape } \lambda. \tag{5.2}$$

See for example [49] or [8, Ch. 19] for a detailed exposition.

The following Lemmas 5.2 and 5.3 follow from eq. (5.1).

▶ **Lemma 5.2.** *Let T and T' be Young tableaux of the same shape with content $(nd) \times 1$ such that T' can be obtained from T by performing permutations within the blocks. The functions f_T and $f_{T'}$ coincide after restricting their domains of definition from $\otimes^d\mathbb{C}^m$ to $\mathrm{Sym}^d\mathbb{C}^m$.*

Proof. If p is symmetric, then p has a Waring rank decomposition, i.e., there exists $r \in \mathbb{N}$ and homogeneous linear forms $p_1, \ldots, p_r$ such that $p = \sum_{i=1}^r p_i^{\otimes d}$. Using this decomposition for p, we see that the summands of $f_T(p)$ and $f_{T'}(p)$ in (5.1) coincide. ◀

Lemma 5.2 implies that in order to define the restriction of f_T to symmetric tensors we only need to define the blocks in T, but not the internal structure of the blocks. Thus for a tableau with content $(nd) \times 1$ we define the tableau $\hat{T}$ by replacing all entries $a \in \{1, \ldots, nd\}$ by $k(a)$. The resulting tableau $\hat{T}$ has content $n \times d$. For example, if $n = 2$, $d = 4$, $T = \begin{array}{|c|c|c|c|c|c|}\hline 1&3&4&6&7&8\\ \hline 2&5 \\ \cline{1-2}\end{array}$, then $\hat{T} = \begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&2&2&2\\ \hline 1&2 \\ \cline{1-2}\end{array}$. For a tableau $\hat{T}$ with content $n \times d$ we define $f_{\hat{T}} \in \mathrm{Sym}^d\mathrm{Sym}^n\mathbb{C}^m$ as the restriction of f_T to $\mathrm{Sym}^n\mathbb{C}^m$.

▶ **Lemma 5.3.** *Let T be a Young tableau that has a column in which there are two or more entries from the same block. Then $f_T = 0$.*

Proof. Let c be the column in T in which there are two or more entries from the same block. As in Lemma 5.2, consider the evaluation of f_T at a point p in its Waring rank decomposition. We observe that every summand in eq. (5.1) is zero, because the determinant corresponding to the column c has a repeated column. ◀

In other words, Lemma 5.3 says that $f_{\hat{T}} = 0$ if $\hat{T}$ contains a column in which a number appears at least twice. Combining this insight with eq. (5.2), we conclude that

$$\mathsf{HWV}_\lambda(\mathrm{Sym}^n\mathrm{Sym}^d\mathbb{C}^m) \text{ is the linear span of the } f_{\hat{T}}, \\ \text{where } \hat{T} \text{ is semistandard of shape } \lambda \text{ with content } n \times d. \tag{5.3}$$

▶ Remark 5.4. From eq. 5.1 and writing p in its Waring rank decomposition, we immediately get an $O(\mathsf{WR}(p)^n \cdot \mathrm{poly}(n, d, m))$ algorithm to evaluate $f_{\hat{T}}(p)$.

6 Non-commutative algebraic branching programs

For an in-depth formal study of ncABPs we now introduce additional notation (cp. Section 4).

▶ **Definition 6.1.** *Let V be a vector space.*

- *A non-commutative algebraic branching program (ncABP) A is an acyclic directed graph with two distinguished nodes s and t and edges labeled with elements from V and every path from s to t having the same length. This makes A layered, with layer k containing all vertices of distance k from s.*
- *The weight $w(P)$ of a path P with edge labels $\ell_1, \dots, \ell_d \in V$ is defined as $w(P) := \ell_1 \otimes \cdots \otimes \ell_d$.*
- *The tensor computed at a node v in A is $\hat{w}(v) = \sum_{s-v \text{ path } P} w(P)$. By convention the tensor computed at s is 1.*
- *The tensor computed by A is the tensor computed at t.*
- *The size of an ncABP is the number of vertices.*
- *The width of an ncABP is the largest number of vertices in any layer.*

In particular we will be looking at ncABPs computing symmetric tensors p and the evaluation of highest weight vectors at p. An example is given in Figure 1.

Each node in layer k computes a tensor in $V^{\otimes k}$. We show in Proposition 6.6 that there is always a minimal ncABP where all these computed tensors are also symmetric and whose size is exactly the size of the partial derivative space of p. An example is given in Figure 1.

We can now use the "overlapping structure of the paths through ncABPs" to our advantage in evaluating HVWs by using dynamic programming.

▶ **Theorem 6.2.** *The evaluation $f_T(p)$ of a highest weight vector $f_T \in \mathrm{Sym}^n\mathrm{Sym}^d\mathbb{C}^m$ given by a Young tableau T with content $(nd) \times 1$ and r rows and a symmetric tensor $p \in \mathrm{Sym}^d\mathbb{C}^m$ given by an ncAPB of width w can be computed in time $O(w^{n+r}\,\mathrm{poly}(n,d,m))$.*

Proof. Let A be an ncABP with edge set $E(A)$, source s, sink t, and width w computing a symmetric tensor $p \in \mathrm{Sym}^d\mathbb{C}^m$. W.l.o.g. let the numbers $i \cdot d + 1, \dots, i \cdot d + d$ occur in order left to right in T for any $i \in \{0, \dots, n-1\}$, see Lemma 5.2. Note that left to right is a unique ordering since if one column contains multiple of these numbers we already know $f_T = 0$, see Lemma 5.3.

Combining eq. (5.1) with $p = \hat{w}(t) = \sum_{s-t \text{ path } P} w(P)$ (see Def. 6.1) we see that

$$f_T(p) := \sum_{\text{proper } \vartheta} \prod_{c=1}^{\lambda_1} \det{}_{\vartheta,c} \quad \text{with} \quad \det{}_{\vartheta,c} := \det\left(\vartheta'(T(1,c)) \dots \vartheta'(T(\mu_c,c))\right), \tag{6.1}$$

where here $\vartheta : \{1, \dots, nd\} \to E(A)$ is called *proper* if there exists $\varphi : \{1, \dots, n\} \to \{s-t \text{ path } P\}$ such that $\vartheta(a) =$ the $j(a)$-th edge of $\varphi(k(a))$ and $\vartheta'(a)$ is the label of $\vartheta(a)$ (see the definitions of j and k in Section 5).

We now calculate partial evaluations in a column by column fashion from right to left. In order to do this we define a partial placement $\vartheta|_{\leq k}$ to be the restriction of ϑ to the boxes in the first k columns of T.

We now observe a common factor for a fixed partial placement $\vartheta|_{\leq k}$:

$$\sum_{\text{proper } \vartheta \text{ extending } \vartheta|_{\leq k}} \prod_{c=1}^{\lambda_1} \det{}_{\vartheta,c} = \left(\prod_{c=1}^{k} \det{}_{\vartheta|_{\leq k},c}\right) \underbrace{\left(\sum_{\text{proper } \vartheta \text{ extending } \vartheta|_{\leq k}} \prod_{c=k+1}^{\lambda_1} \det{}_{\vartheta,c}\right)}_{=:\alpha(\vartheta|_{\leq k})}.$$

Since each proper ϑ corresponds to a set of n many $s-t$ paths, each $\vartheta|_{\leq k}$ defines a set of n partial $s-t$ paths all starting at s (of potentially different lengths, one path for each block), where $\alpha(\vartheta|_{\leq k})$ only depends on the endpoints of these paths. These paths start at s and go up to these endpoints due to the nature of T being ordered from left to right for each block of n numbers. This crucial observation allows us to store and reuse these values of α whenever two partial assignments correspond to lists of n paths ending in the same vertices of A.

We can now calculate the evaluation as $f_T(p) = \alpha(\vartheta|_{\leq 0}) = \alpha(\emptyset)$.

Since the length of each of the paths defined by any $\vartheta|_{\leq k}$ are fixed for fixed k, there are at most w^n possible different values for α that need to be computed. So in total this evaluation algorithm has running time $O(w^{n+r}\operatorname{poly}(n, d, m))$. The w^r term comes from all the possibilities to extend a given $\vartheta|_{\leq k}$ by one column of T. ◀

▶ Remark 6.3. Note that Theorem 6.2 is a generalisation of the dynamic programming used in [25] to evaluate HWVs at the Chow variety Ch^d_m. The Chow variety Ch^d_m consists of products of d linear forms $\ell_1 \odot \cdots \odot \ell_d \in \operatorname{Sym}^d\mathbb{C}^m$. Here the minimal ncABP A of $\ell_1 \odot \cdots \odot \ell_d$ corresponds to having subsets of $\{1, \ldots, d\}$ as vertices where two vertices $U, V \subseteq \{1, \ldots n\}$ are connected by an edge labeled ℓ_i iff $U \setminus V = \{i\}$ and $U \supset V$. Then A has size exactly 2^d and width $\binom{d}{k}$ on layer k while layer k contains all the sets of size k.

We now give the connection between the width of ncABPs, and the dimension of the partial derivative spaces of the symmetric tensors computed by the ncAPB. We additionally show that ncABPs can efficiently compute partial derivatives.

First note that the following equivalence between partial derivatives and polynomial contractions is well known for fields of characteristic 0, see for example [33, Equation 1.1.2] and [16, §5(a)]. We reformulate this as an equivalence between partial derivatives and tensor contractions instead. For $V = \mathbb{C}^m$ with standard basis $e_1, \ldots, e_m$ the tensor contraction $\langle \cdot, \cdot \rangle : \bigotimes^r V \times \bigotimes^s V \to \bigotimes^{s-r} V$ is defined for any $r < s$ on the basis vectors via

$$\langle e_{i_1} \otimes \cdots \otimes e_{i_r}, e_{j_1} \otimes \cdots \otimes e_{j_s} \rangle = \begin{cases} e_{j_{r+1}} \otimes e_{j_{r+2}} \otimes \cdots \otimes e_{j_s} & \text{if } i_k = j_k \text{ for all } 1 \leq k \leq r \\ 0 & \text{otherwise} \end{cases}$$

and extended via linear continuation in both parameters.

▶ **Lemma 6.4.** *Let φ be the canonical isomorphism between $\operatorname{Sym}^d\mathbb{C}^m$ and $\mathbb{C}[x_1, \ldots, x_m]_d$ defined via $\varphi(e_{i_1} \odot \cdots \odot e_{i_d}) = x_{i_1} \cdots x_{i_d}$. The partial derivative $\frac{\partial^k}{\partial \ell_1 \cdots \partial \ell_k} : \mathbb{C}[x_1, \ldots, x_m]_d \to \mathbb{C}[x_1, \ldots, x_m]_{d-k}$ induces a linear map $\operatorname{Sym}^d\mathbb{C}^m \to \operatorname{Sym}^{d-k}\mathbb{C}^m$ via φ that we also call $\frac{\partial^k}{\partial \ell_1 \cdots \partial \ell_k}$. Then the $\frac{\partial^k}{\partial \ell_1 \cdots \partial \ell_k} t$ of a symmetric tensor $t \in \operatorname{Sym}^d V$ is given by the tensor contraction $\frac{d!}{(d-k)!} \langle \ell_1 \otimes \cdots \otimes \ell_k, t \rangle$.*

Since t is symmetric the partial derivative $\frac{\partial^k}{\partial \ell_1 \cdots \partial \ell_k} t$ is also given by $\frac{d!}{(d-k)!} \langle \ell_1 \odot \cdots \odot \ell_k, t \rangle$.

Proof. It suffices to prove this for the case $k = 1$, since repeated tensor contraction is the same as one big tensor contraction and the same holds for partial derivatives. Since both tensor contraction and taking derivatives are linear operations in both parameters we can restrict ourselves to the derivative $\frac{\partial}{\partial e_i}(e_{j_1} \odot \cdots \odot e_{j_d})$ and prove that $\frac{\partial}{\partial e_i}(e_{j_1} \odot \cdots \odot e_{j_d}) = d \cdot \langle e_i, e_{j_1} \odot \cdots \odot e_{j_d} \rangle$. The factor of $\frac{d!}{(d-k)!} = d(d-1)(d-2) \cdots (d-k+1)$ is then the result of repeatedly taking the derivative.

In case e_i is not any of $e_{j_1}, \ldots, e_{j_d}$ clearly

$$\frac{\partial}{\partial e_i}(e_{j_1} \odot \cdots \odot e_{j_d}) = 0 = \frac{d!}{(d-k)!} \langle e_i, e_{j_1} \odot \cdots \odot e_{j_d} \rangle$$

so w.l.o.g. we can now assume due to symmetry $e_{j_1} = e_i$.

We can write $\varphi(e_i \odot e_{j_2} \odot e_{j_3} \odot \cdots \odot e_{j_d}) = x_i^h \cdot q$, $h \geq 1$, for some monomial $q \in \mathbb{C}[x_1, \ldots, x_m]$ not containing x_i.

$$\begin{aligned}\frac{\partial}{\partial e_i}\left(\varphi\left(e_i \odot e_{j_2} \odot e_{j_3} \odot \cdots \odot e_{j_d}\right)\right) &= h \cdot x_i^{h-1} \cdot q \\ &= \varphi\left(h \cdot e_{j_2} \odot e_{j_3} \odot \cdots \odot e_{j_d}\right) \\ &= \varphi\left(\langle e_i, h \cdot e_i \otimes (e_{j_2} \odot e_{j_3} \odot \cdots \odot e_{j_d})\rangle\right) \\ &= \varphi\left(\langle e_i, d \cdot e_i \odot e_{j_2} \odot e_{j_3} \odot \cdots \odot e_{j_d}\rangle\right)\end{aligned}$$

The last equality follows from the fact that all terms of the symmetric tensor not containing e_i as the first component of the tensor vanish under the tensor contraction. ◀

▶ **Lemma 6.5.** *If A is an ncABP computing a symmetric tensor $p \in \mathrm{Sym}^d V$, then the k-th derivatives are linear combinations of the tensors computed at the $(d-k)$-th layer of A.*

Proof. As proven in Lemma 6.4 the derivatives are just tensor contractions. A tensor contraction on an ncABP replaces the last k edges on each s-t path by constants[4], thus directly proving the claim. ◀

We will now characterize the minimal size of ncABPs via the dimension of the partial derivative spaces. For this we denote by $\partial^{=k}(t)$ the partial derivative space of k-th order for $t \in \mathrm{Sym}^d V$:

$$\partial^{=k}(t) := \{\langle q, t\rangle \mid q \in \mathrm{Sym}^k V\}$$

Analogously we define

$$\partial^{\leq k}(t) := \mathrm{span} \bigcup_{i=0}^{k} \partial^{=i}(t) \,.$$

Note that the usage of tensor contractions instead of derivatives is just for simplicity.

For a list $q \in \{1, \ldots, m\}^k$ let $e_q := e_{q_1} \otimes \cdots \otimes e_{q_k}$. For a tensor $p \in \otimes^d \mathbb{C}^m$ we define the $m^k \times m^{d-k}$ matrix $M_k(p)$ whose rows are indexed by elements $q \in \{1, \ldots, m\}^k$ and whose columns are indexed by elements in $q' \in \{1, \ldots, m\}^{d-k}$ via

$$M_k(p)[q, q'] := \text{ the coefficient of } e_q \otimes e_{q'} \text{ in } p. \tag{6.2}$$

These matrices are sometimes called *flattenings* of the tensor p.

▶ **Proposition 6.6.** *If A is an ncABP computing a symmetric tensor $p \in \mathrm{Sym}^d V$, then there is an ncABP B with the following properties:*

1. *B also computes p.*
2. *Each layer of B has at most as many vertices as the same layer in A.*
3. *Each node of B computes a symmetric tensor.*
4. *The k-th layer of B has precisely $\dim \partial^{=k}(p)$ many vertices which is the optimal width.*

[4] due to the symmetry of p we could even choose any k layers and all outgoing edges out of these chosen layers would be replaced by constants for the derivative.

Proof. We mainly follow Nisan [47] who constructed minimal ncABPs and extend this to also compute symmetric tensors at each node and establishing the connection to the dimensions of the partial derivative spaces. For an example of a minimal ncAPB with symmetric tensors computed at each node can be seen in Figure 1.

Let $v_1, \dots, v_t$ be the vertices in a fixed layer k. Let $M_k[q, q'] := M_k(p)[q, q']$ from eq. (6.2). Note that the row of M_k corresponding to q is given precisely by the tensor contraction $\langle e_q, p\rangle$ (columns are indexed by q' and further contraction with $e_{q'}$ gives the matrix entry) and it is thus by Lemma 6.4 a partial derivative of k-th order. Therefore $\operatorname{rank} M_k = \dim \partial^{=k}(p)$.

Now we can construct two matrices L_k and R_k. Here $L_k[q, i]$ for indices $q \in \{e_1, \dots, e_{\dim V}\}^{\otimes k}$ is defined as the coefficient of q in $\hat{w}(v_i)$ and $R_k[i, q']$ for indices $q' \in \{e_1, \dots, e_{\dim V}\}^{\otimes(d-k)}$ is defined as the coefficient of q' in the tensor computed by the restricted ncABP with source v_i. It is easy to verify $M_k = L_k R_k$.

Hence if $t > \operatorname{rank} L_k$ there must be some vertices v_i computing a linear combination of the other vertices in the same layer, thus all outgoing edges of v_i can be replaced by precisely this linear combination, allowing us to remove v_i. In this way we can remove some v_i as long as $t > \operatorname{rank} R_k$.

After this process finishes we have $t = \operatorname{rank} L_k = \operatorname{rank} R_k = \operatorname{rank} M_k = \dim \partial^{=k}(p)$ proving the claims on the width of the layers.

Since by Lemma 6.5 all the $(d - k)$-th partial derivatives are linear combinations of restrictions of the ncABP to the first k levels we can now replace all vertices on the k-th level by t vertices computing a symmetric tensor basis of the k-th partial derivatives thus proving the remaining claim. ◀

From this characterization of ncABP size as the rank of the partial derivative matrices we can also see that ncABP size is preserved under approximation. This was remarked by Michael Forbes [28], but we give a proof for the sake of completeness.

▶ **Corollary 6.7.** *Let $p \in \operatorname{Sym}^d V$ and $(A_i)_{i\in\mathbb{N}}$ be ncABPs s.t. A_i computes $p_i \in \otimes^d V$ and has size $s_i \le s$ and width $w_i \le w$ with $\lim_{i\to\infty} p_i = p$. Then there is an ncABP A computing p with size at most s and width at most w.*

Proof. Let the matrices $M_{k,p_i} := M_k(p_i)$ from eq. (6.2). We have

$$M_{k,p} = \lim_{i\to\infty} M_{k,p_i} .$$

Since each A_i has width at most w, we know that $\operatorname{rank} M_{k,p_i} \le w_i \le w$. This is characterized by all determinants of $(w + 1) \times (w + 1)$ minors of M_{k,p_i} vanishing. So by continuity of the determinant also all $(w + 1) \times (w + 1)$ minors of $M_{k,p}$ vanish and thus $\dim \partial^{=k}(p) = \operatorname{rank} M_{k,p} \le w$ and there is an ncABP A with width at most w by Proposition 6.6.

This constructed A directly has size at most s. For this we note that the partial derivatives of different orders are linearly independent, so $\dim \partial^{\le d}(p) = \sum_{j=0}^{d} \dim \partial^{=j}(p) = s$. This is the same as looking at the rank of the direct sum $\oplus_{j=0}^{d} M_{j,p}$, so the bound on the size of A follows from the same continuity argument. ◀

From this we can conclude an order of inclusion on the sets of symmetric tensors of small Waring rank, small border Waring rank and small non-commutative ncABP size.

▶ **Corollary 6.8.** *Let $k \in \mathbb{N}$ and*

$$\begin{aligned} W_{k,d} &:= \{p \in \mathrm{Sym}^d V \mid \mathsf{WR}(p) \leq k\}\,, \\ \overline{W_{k,d}} &:= \{p \in \mathrm{Sym}^d V \mid \underline{\mathsf{WR}}(p) \leq k\}\,, \\ B_{k,d} &:= \{p \in \mathrm{Sym}^d V \mid \mathsf{ncw}(p) \leq k\}\,. \\ \overline{B_{k,d}} &:= \{p \in \mathrm{Sym}^d V \mid \underline{\mathsf{ncw}}(p) \leq k\}\,. \end{aligned}$$

Then

$$W_{k,d} \subseteq \overline{W_{k,d}} \subseteq B_{k,d} = \overline{B_{k,d}}$$

and there exist k, d for which the inclusions are strict.

Proof. The inclusion $W_{k,d} \subseteq \overline{W_{k,d}}$ is trivial and $B_{k,d} = \overline{B_{k,d}}$ is proven in Corollary 6.7. To show $W_{k,d} \subsetneq \overline{W_{k,d}}$ is strict we refer to [19] showing that $x^{d-1}y$ has Waring rank d while it is known[5] that $x^{d-1}y = \lim_{\varepsilon\to 0} \frac{1}{\varepsilon d}((x+\varepsilon y)^d - x^d)$ and thus $x^{d-1}y$ has border Waring rank at most 2. For the inclusion $W_{k,d} \subseteq B_{k,d}$ we can embed the k summands ℓ_i^d of the Waring rank decomposition as disjoint $s-t$ paths in an ncABP of width k and depth d. Here every edge on the path corresponding to ℓ_i^d has the label ℓ_i. Since $B_{k,d}$ is closed this immediately proves $\overline{W_{k,d}} \subseteq B_{k,d}$. An example for $W_{k,d} \neq B_{k,d}$ is given by the 2×2 matrix multiplication polynomial $p = x_{1,1}^3 + 3x_{1,1}x_{1,2}x_{2,1} + 3x_{1,2}x_{2,2}x_{2,1} + x_{2,2}^3$ that is studied in [21]: We have $\mathsf{ncw}(p) = 4$, but $\underline{\mathsf{WR}}(p) \geq 5$, which can be seen using *Young flattenings*. This representation theoretic technique is explained for example in [26]. The `Macaulay2` code

```
loadPackage "PieriMaps"
MX = pieri ({4,3,2,2} , {1,2,4} , QQ [x11,x12,x21,x22])
p = x11*x11*x11 + 3*x11*x12*x21 + 3*x12*x22*x21 + x22*x22*x22
rank(diff(p,MX))/rank(diff(x11^3,MX))
```

outputs 5, which is the lower bound on the border Waring rank. ◀

Note that the following is still unknown:

▶ **6.9 Question.** *Is there a polynomial q, such that $B_{k,d} \subseteq W_{q(k),d}$ or $B_{k,d} \subseteq \overline{W_{q(k),d}}$?*

7 Treewidth of Young tableaux

Let S be an arbitrary Young tableau containing the numbers $\{1,\ldots,n\}$. We can associate with S the undirected graph $G_S = (V_S, E_S)$ where $V_S = \{1,\ldots,n\}$ and $\{i,j\} \in E_S$ iff i and j are contained in some common column in S, see Figure 2(a) and (b).

We are now going to study how we can use the graph parameter treewidth of G_S to speed up the evaluation of highest weight vectors. Treewidth has been intensely studied by Robertson and Seymour and has been applied numerous times to construct faster graph algorithms for cases where the treewidth is bounded by a function $o(n)$, most notably some algorithms for NP-hard problems restricted to planar graphs, for example 3-coloring. See [23] for an introduction to treewidth algorithms.

[5] Technically we need here that our base field is algebraically closed in order for this to be a border Waring rank decomposition, but $\mathbb{C}$ satisfies this.

▶ **Definition 7.1.** *A* tree decomposition *of a graph $G = (V, E)$ is a tree $\mathcal{T}$ with vertices $X_1, X_2, \dots, X_t$ called bags where $X_i \subseteq V$ and the following properties hold:*

- $\cup_{i=1}^{t} X_i = V$
- *For every edge $\{u, v\} \in E$ there is some bag X_i, s.t. $\{u, v\} \subseteq X_i$.*
- *For every vertex $v \in V$ the bags containing v form a subtree of $\mathcal{T}$.*

The width *of a tree decomposition is the size of the largest bag minus one. The* treewidth *of G is then the smallest possible width of a tree decomposition for G.*

Often solving problems on graphs of bounded treewidth is easier then the general problem and indeed this is also the case for evaluating the highest weight vector corresponding to a graph if the graph $G_{\hat{T}}$ has bounded or low treewidth.

▶ **Theorem 7.2.** *The evaluation $f_{\hat{T}}(p)$ for a highest weight vector $f_{\hat{T}} \in \mathrm{Sym}^n\mathrm{Sym}^d\mathbb{C}^m$ given by a Young tableau $\hat{T}$ with content $n \times d$ and a symmetric tensor $p \in \mathrm{Sym}^d\mathbb{C}^m$ given by an ncABP A of width w can be computed in time $w^{\omega(\tau+1)}\,\mathrm{poly}(n, d, m, |\mathcal{T}|)$ if a tree decomposition $\mathcal{T}$ of $G_{\hat{T}}$ of width τ and size $|\mathcal{T}|$ is given and given that we can multiply two matrices of size $\leq k \times k$ in time $O(k^\omega)$.*

Proof. We generalize the algorithm from Theorem 6.2, by using the tree-decomposition of $G_{\hat{T}}$ to be able to reuse even more partial results in the evaluation of (5.1). Let A be a ncABP with source v_{source} and sink v_{sink}. The label on the edge from v to w shall be called $A_{(v,w)}$.

A tableau $\hat{T}$ with its corresponding graph $G_{\hat{T}}$ is depicted in Figure 2(a) and (b). It is well known that every clique of a graph is fully contained in some bag of its tree decomposition. Every column c of $\hat{T}$ corresponds to a clique in $G_{\hat{T}}$, so there is some bag X_i of $\mathcal{T}$ which contains all the vertices corresponding to the entries of c. We modify $\mathcal{T}$ by adding a new vertex which is only adjacent to X_i. This vertex is from now on associated with the column c and contains all the entries contained in c as its bag. An example is given in Figure 2(c). Without loss of generality we can assume that if a vertex has only one child, then the vertex has the same bag as the child. From now on we only need the structure of the subtree $\mathcal{T}'$ of $\mathcal{T}$ whose leaves are the vertices associated with columns and every vertex removed that is not on the path between two of these vertices. We interpret $\mathcal{T}'$ as an ordered binary tree rooted at an arbitrary internal[6] vertex r. In case any vertex v of $\mathcal{T}'$ has more than two children, we replace v by a binary tree, where each added vertex has the same bag as v, see Figure 2(e). Since $\mathcal{T}'$ is also a tree decomposition, for every vertex v with two children v_{left} and v_{right} we have

$$X_{v_{\text{left}}} \cap X_{v_{\text{right}}} \subseteq X_v. \tag{7.1}$$

We start with a few observations. We sort the leaves of $\mathcal{T}'$ according to when they are visited by depth-first search that always takes the left child first. In this way every leaf $\mathcal{T}'$ gets assigned an index from 1 to λ_1, which we call the *traversal index* of the leaf. Let ν_i denote the length of the column of λ with traversal index i. For any internal vertex v of $\mathcal{T}'$ let $\mathsf{leftmost}(v)$ denote the traversal index of the leftmost leaf of the subtree rooted at v. Analogously, let $\mathsf{rightmost}(v)$ denote the traversal index of the rightmost leaf of the subtree rooted at v. For a leaf v we define $\mathsf{leftmost}(v) = \mathsf{rightmost}(v)$ to be the traversal index of v. For any internal vertex v of $\mathcal{T}'$ with two children $v_{\text{left}}, v_{\text{right}}$ by definition we have

$$\mathsf{leftmost}(v) = \mathsf{leftmost}(v_{\text{left}}) \text{ and } \mathsf{rightmost}(v) = \mathsf{rightmost}(v_{\text{right}}) \tag{7.2}$$

[6] i.e. a non-leaf

and

$$\mathsf{rightmost}(v_{\mathrm{left}}) = \mathsf{leftmost}(v_{\mathrm{right}}) - 1. \tag{7.3}$$

We define leaves(v) to be the set of leaves in $\mathcal{T}'$ with traversal index at least $\mathsf{leftmost}(v)$ and at most $\mathsf{rightmost}(v)$.

For $1 \le i \le n$, $0 \le t \le \lambda_1$, define $\kappa_t(i)$ to be the number of times the number i appears in columns with traversal index at most t. Figure 2(f) shows diamond separators that mark the values for t so that $\kappa_t(i)$ is the number of times the number i appears in columns left of the diamond t. For an internal vertex v with children v_{left} and v_{right} we define $\mathsf{mid}(v) := \mathsf{rightmost}(v_{\mathrm{left}})$. Pictorially, this is the number of the diamond separator between the left and right subtree of v. If v only has one child w, then $\mathsf{mid}(v) = \mathsf{mid}(w)$. Let the ncABP A have layers $L_0, \dots, L_d$, $|L_0| = |L_d| = 1$. We assume that in A all edges between any layers L_i and L_{i+1} exist, hence we allow edges that are labelled with 0. For $0 \le t \le \lambda_1$ define

$$\mathcal{F}_t := L_{\kappa_t(1)} \times \cdots \times L_{\kappa_t(n)}$$

Let $t_{\mathrm{start}} \le t_{\mathrm{end}}$ and let $\Phi_{\mathrm{start}} \in \mathcal{F}_{t_{\mathrm{start}}}$ and $\Phi_{\mathrm{end}} \in \mathcal{F}_{t_{\mathrm{end}}}$. A *$\Phi_{start}$-$\Phi_{end}$-multiwalk* is defined as a finite sequence

$$\mathcal{W} := (\Phi_{t_{\mathrm{start}}}, \Phi_{t_{\mathrm{start}}+1}, \Phi_{t_{\mathrm{start}}+2}, \dots, \Phi_{t_{\mathrm{end}}})$$

such that each $\Phi_t \in \mathcal{F}_t$, and for all t, i with $\kappa_t(i) = \kappa_{t+1}(i)$ we have $\Phi_t(i) = \Phi_{t+1}(i)$. To explain this notion more pictorially, we define a *lazy walk* in a digraph to be a walk that as a step can remain at its vertex instead of advancing over an edge. If a digraph does not have any loops, then for every finite lazy walk there is a corresponding walk on the digraph that is obtained if we add all loops: Remaining at a vertex has the same effect as taking the loop. This is also true in the reverse direction. The i-th walk of a Φ_{start}-Φ_{end}-multiwalk $(\Phi_{t_{\mathrm{start}}}, \Phi_{t_{\mathrm{start}}+1}, \dots, \Phi_{t_{\mathrm{end}}})$ is defined as the sequence $(\Phi_{t_{\mathrm{start}}}(i), \Phi_{t_{\mathrm{start}}+1}(i), \dots, \Phi_{t_{\mathrm{end}}}(i))$, which is a lazy walk in A from $\Phi_{t_{\mathrm{start}}}(i)$ to $\Phi_{t_{\mathrm{end}}}(i)$.

We now define the determinant $\det(\mathcal{W})$. Note that Φ_t and Φ_{t+1} differ in exactly ν_{t+1} positions. We define $\det_{\mathcal{W},t+1}$ as the determinant of the $\nu_{t+1} \times \nu_{t+1}$-matrix obtained from taking the top μ_{t+1} of the edge labels in A that connect Φ_t with Φ_{t+1}. We define $\det(\mathcal{W})$ as

$$\det(\mathcal{W}) := \prod_{c=t_{\mathrm{start}}+1}^{t_{\mathrm{end}}} \det{}_{\mathcal{W},c}$$

For $\Phi_{\mathrm{start}} \in \mathcal{F}_{\mathsf{leftmost}(v)-1}$ and $\Phi_{\mathrm{end}} \in \mathcal{F}_{\mathsf{rightmost}(v)}$ we define

$$D[v, \Phi_{\mathrm{start}}, \Phi_{\mathrm{end}}] := \sum_{\Phi_{\mathrm{start}}\text{-}\Phi_{\mathrm{end}}\text{-multiwalk } \mathcal{W}} \det(\mathcal{W}) \tag{7.4}$$

Note that

$$D[v, \Phi_{\mathrm{start}}, \Phi_{\mathrm{end}}] = \sum_{\Phi_{\mathrm{start}}\text{-}\Phi_{\mathrm{end}}\text{-multiwalk } \mathcal{W}} \prod_{c=\mathsf{leftmost}(v)}^{\mathsf{rightmost}(v)} \det{}_{\mathcal{W},c}. \tag{7.5}$$

Recall that r is the chosen root. We claim that

$$f_{\hat{T}}(p) = D[r, \overrightarrow{v_{\mathrm{source}}}, \overrightarrow{v_{\mathrm{sink}}}], \tag{7.6}$$

where $\overrightarrow{v_{\text{source}}} = (v_{\text{source}}, v_{\text{source}}, \dots, v_{\text{source}})$ and $\overrightarrow{v_{\text{sink}}} = (v_{\text{sink}}, v_{\text{sink}}, \dots, v_{\text{sink}})$. To see this, we observe that the ordering of the leaves of $\mathcal{T}'$ by their traversal index defines an ordering on the columns of tableaux of shape λ. We call this ordering the leaf-ordering. Let T be the following tableau of shape λ and content $(nd) \times 1$ that is a preimage of $\hat{T}$ under the $\hat{\cdot}$-operation (see Section 5): We greedily go through the columns of $\hat{T}$ from left to right in the leaf-order and replace each entry i by the smallest still unused number from $\{(i-1) \cdot d + 1, (i-1) \cdot d + 2, \dots, i \cdot d\}$, see Figure 2(d) for an example. Then $f_{\hat{T}}(p) = f_T(p)$ is given as

$$f_T(p) \overset{(5.1)}{=} \sum_{\text{proper } \vartheta} \prod_{c=1}^{\lambda_1} \det{}_{\vartheta,c} \overset{(*)}{=} D[r, \overrightarrow{v_{\text{source}}}, \overrightarrow{v_{\text{sink}}}]\,.$$

$(*)$ can be seen from the fact that there is a natural 1:1 correspondence between proper ϑ and $\overrightarrow{v_{\text{source}}}$-$\overrightarrow{v_{\text{sink}}}$-multiwalks $\mathcal{W}$: A tensor assigned to the i-th block of T is given by an v_{source}-v_{sink}-path in A, which uniquely specifies the i-th path of the multiwalk $\mathcal{W}$. If ϑ is mapped to $\mathcal{W}$ under this bijection, then $\det(\mathcal{W}) = \prod_{c=1}^{\lambda_1} \det{}_{\vartheta,c}$. This proves (7.6).

We now explain how to compute $D[r, \overrightarrow{v_{\text{source}}}, \overrightarrow{v_{\text{sink}}}]$ recursively over the tree structure of $\mathcal{T}'$. We separate the explanation into several claims. First, we claim that for every internal vertex $v \in \mathcal{T}'$ with only one child v' we have

$$D[v, \Phi_{\text{start}}, \Phi_{\text{end}}] = D[v', \Phi_{\text{start}}, \Phi_{\text{end}}]. \tag{7.7}$$

The right-hand side is well-defined, because $\mathsf{leftmost}(v) = \mathsf{leftmost}(v')$ and $\mathsf{rightmost}(v) = \mathsf{rightmost}(v')$. The equality follows directly from the definition: (7.4). Next, we claim that for every leaf vertex $v \in \mathcal{T}'$ with corresponding column c we have

$$D[v, \Phi_{\text{start}}, \Phi_{\text{end}}] = \det(A_{(\Phi_{\text{start}}(c_1), \Phi_{\text{end}}(c_1))}, A_{(\Phi_{\text{start}}(c_2), \Phi_{\text{end}}(c_2))}, \cdots, A_{(\Phi_{\text{start}}(c_{|c|}), \Phi_{\text{end}}(c_{|c|}))}). \tag{7.8}$$

This follows from the fact that in this case there is exactly one Φ_{start}-Φ_{end}-multiwalk $\mathcal{W}$, and $\mathsf{leftmost}(v) = \mathsf{rightmost}(v)$ is the traversal index of v. The crucial claim is the following. For any inner vertex v of $\mathcal{T}'$ with two children $v_{\text{left}}, v_{\text{right}}$ we claim

$$D[v, \Phi_{\text{start}}, \Phi_{\text{end}}] = \sum_{\Phi_{\text{mid}} \in \mathcal{F}_{\mathsf{mid}(v)}} D[v_{\text{left}}, \Phi_{\text{start}}, \Phi_{\text{mid}}] \cdot D[v_{\text{right}}, \Phi_{\text{mid}}, \Phi_{\text{end}}]. \tag{7.9}$$

Before proving this, first note that $D[v_{\text{left}}, \Phi_{\text{start}}, \Phi_{\text{mid}}]$ on the right-hand side is well-defined, as can be seen by combining (7.2) and (7.3) with the definition of $\mathsf{mid}(v)$. Analogously, $D[v_{\text{right}}, \Phi_{\text{mid}}, \Phi_{\text{end}}]$ is well-defined.

The key tool in the proof of (7.9) is the bijection

$$\{\Phi_{\text{start}}\text{-}\Phi_{\text{end}}\text{-multiwalk}\} \simeq \bigcup_{\Phi_{\text{mid}} \in \mathcal{F}_{\mathsf{mid}(v)}} (\{\Phi_{\text{start}}\text{-}\Phi_{\text{mid}}\text{-multiwalk}\} \times \{\Phi_{\text{mid}}\text{-}\Phi_{\text{end}}\text{-multiwalk}\}) \tag{7.10}$$

given by splitting the multiwalk into two multiwalks, where the inverse map is given by contatenating two multiwalks. The union on the right-hand side is a disjoint union. (7.9) is now proved by a direct calculation as follows.

$$\begin{aligned}
&\sum_{\Phi_{\mathrm{mid}} \in \mathcal{F}_{\mathsf{mid}(v)}} D[v_{\mathrm{left}}, \Phi_{\mathrm{start}}, \Phi_{\mathrm{mid}}] \cdot D[v_{\mathrm{right}}, \Phi_{\mathrm{mid}}, \Phi_{\mathrm{end}}] \\
\overset{(7.5)}{=}\ & \sum_{\Phi_{\mathrm{mid}} \in \mathcal{F}_{\mathsf{mid}(v)}} \left(\sum_{\Phi_{\mathrm{start}}\text{-}\Phi_{\mathrm{mid}}\text{-multiwalk } \mathcal{W}_{\mathrm{left}}} \ \prod_{c_{\mathrm{left}}=\mathsf{leftmost}(v_{\mathrm{left}})}^{\mathsf{rightmost}(v_{\mathrm{left}})} \det_{\mathcal{W},c_{\mathrm{left}}} \right) \\
&\qquad \cdot \left(\sum_{\Phi_{\mathrm{mid}}\text{-}\Phi_{\mathrm{end}}\text{-multiwalk } \mathcal{W}_{\mathrm{right}}} \ \prod_{c_{\mathrm{right}}=\mathsf{leftmost}(v_{\mathrm{right}})}^{\mathsf{rightmost}(v_{\mathrm{right}})} \det_{\mathcal{W},c_{\mathrm{right}}} \right) \\
\overset{(7.10)}{=}\ & \sum_{\Phi_{\mathrm{start}}\text{-}\Phi_{\mathrm{end}}\text{-multiwalk } \mathcal{W}} \left(\prod_{c_{\mathrm{left}}=\mathsf{leftmost}(v_{\mathrm{left}})}^{\mathsf{rightmost}(v_{\mathrm{left}})} \det_{\mathcal{W},c_{\mathrm{left}}} \right) \left(\prod_{c_{\mathrm{right}}=\mathsf{leftmost}(v_{\mathrm{right}})}^{\mathsf{rightmost}(v_{\mathrm{right}})} \det_{\mathcal{W},c_{\mathrm{right}}} \right) \\
\overset{(7.3)}{=}\ & \sum_{\Phi_{\mathrm{start}}\text{-}\Phi_{\mathrm{end}}\text{-multiwalk } \mathcal{W}} \left(\prod_{c=\mathsf{leftmost}(v_{\mathrm{left}})}^{\mathsf{rightmost}(v_{\mathrm{right}})} \det_{\mathcal{W},c} \right) \overset{(7.2),(7.5)}{=} D[v, \Phi_{\mathrm{start}}, \Phi_{\mathrm{end}}].
\end{aligned}$$

This proves (7.9).

Equations (7.7), (7.8), and (7.9) give us a procedure to compute $D[r, \vec{s}, \vec{t}]$ by induction over the structure of the tree $\mathcal{T}'$. But we can improve the running time significantly as follows (the procedure is illustrated in Figure 3).

We arbitrarily order the vertices within each layer such that every vertex $v \in L_j$ has an index $\iota(v) \in \{1, \ldots, |L_j|\}$. Of course the vertex v with $\iota(v) = 1$ plays no special role, but v is useful to find a normal form for paths from L_j to L_j of length 0: They go from v to v. A vertex v with $\iota(v) = 1$ is called a *principal* vertex.

For a set of $X \subseteq \mathbb{N}$ define the subset $\mathcal{F}_t^X \subseteq \mathcal{F}_t$ as

$$\mathcal{F}_t^X := \{\Phi \in \mathcal{F}_t \mid \text{ for all } i \notin X : \ \Phi(i) \text{ is principal}\}$$

For every $\Phi \in \mathcal{F}_t$ define $\Phi^X \in \mathcal{F}_t^X$ via

$$\Phi^X(i) := \begin{cases} \Phi(i) & \text{if } i \in X \\ \text{a principal vertex } v & \text{otherwise.} \end{cases} \tag{7.11}$$

Clearly $\overrightarrow{v_{\mathrm{source}}} = \overrightarrow{v_{\mathrm{source}}}^{X_r}$ and $\overrightarrow{v_{\mathrm{sink}}} = \overrightarrow{v_{\mathrm{sink}}}^{X_r}$, hence we see $D[r, \overrightarrow{v_{\mathrm{source}}}, \overrightarrow{v_{\mathrm{sink}}}] = D[r, \overrightarrow{v_{\mathrm{source}}}^{X_r}, \overrightarrow{v_{\mathrm{sink}}}^{X_r}]$. We claim that

$$D[v, \Phi_{\mathrm{start}}, \Phi_{\mathrm{end}}] = \begin{cases} 0 & \text{if there exists } i \notin X_v \text{ with } \iota(\Phi_{\mathrm{start}}(i)) \neq \iota(\Phi_{\mathrm{end}}(i)) \\ D[v, \Phi_{\mathrm{start}}^{X_v}, \Phi_{\mathrm{end}}^{X_v}] & \text{otherwise.} \end{cases} \tag{7.12}$$

To see this, first assume that there is $i \notin X_v$ with $\iota(\Phi_{\mathrm{start}}(i)) \neq \iota(\Phi_{\mathrm{end}}(i))$. Since $i \notin X_v$, we either have *all* instances of i in leaves(v_{left}) or *all* instances of i in leaves(v_{right}) or *no* instance of i in leaves(v). The first two cases are impossible, because in those cases we have $\iota(\Phi_{\mathrm{start}}(i)) = 1 = \iota(\Phi_{\mathrm{end}}(i))$, because the first and last layer only have one vertex each. In the third case, there is no Φ_{start}-Φ_{end}-multiwalk, because the i-th path in the multiwalk would have to start at a vertex and end at a different vertex without using any edge. This proves the first case of (7.12). Now, assume that for all $i \notin X_v$ we have $\iota(\Phi_{\mathrm{start}}(i)) = \iota(\Phi_{\mathrm{end}}(i))$. We have a distinction into the same three cases as above. If $i \notin X_v$ has all instances of i in leaves(v_{left}) or in leaves(v_{right}), then $\iota(\Phi_{\mathrm{start}}(i)) = 1 = \iota(\Phi_{\mathrm{end}}(i))$, which implies $\Phi^{X_v}_{\text{start}}(i) = \Phi_{\text{start}}(i)$ and $\Phi^{X_v}_{\text{end}}(i) = \Phi_{\text{end}}(i)$. If $i \notin X_v$ has no instance of i in leaves(v), then the value of $\iota(\Phi_{\text{start}}(i))$ and $\iota(\Phi_{\text{end}}(i))$ does not affect $D[v, \Phi_{\text{start}}, \Phi_{\text{end}}]$, as long as $\iota(\Phi_{\text{start}}(i)) = \iota(\Phi_{\text{end}}(i))$, because the i-th path in any Φ_{start}-Φ_{end}-multiwalk is unique and of length 0. This proves (7.12).

We use the short notation $\iota(\Phi_{\text{start}}|_{\overline{X_v}}) \neq \iota(\Phi_{\text{end}}|_{\overline{X_v}})$ for the first condition in (7.12): The ι-values of the vectors Φ_{start} and Φ_{end} are different when restricted to the complement of X_v.

To compute $D[v, \Phi^{X_v}_{\text{start}}, \Phi^{X_v}_{\text{end}}]$ recursively we observe the following.

$$\begin{aligned} D[v, \Phi^{X_v}_{\text{start}}, \Phi^{X_v}_{\text{end}}] &= \sum_{\Phi_{\text{mid}} \in \mathcal{F}_{\mathsf{mid}(v)}} D[v_{\text{left}}, \Phi^{X_v}_{\text{start}}, \Phi_{\text{mid}}] \cdot D[v_{\text{right}}, \Phi_{\text{mid}}, \Phi^{X_v}_{\text{end}}] \qquad (7.13) \\ &\overset{(7.12)}{=} \sum_{\Phi_{\text{mid}}} D[v_{\text{left}}, \Phi^{X_v}_{\text{start}}, \Phi_{\text{mid}}] \cdot D[v_{\text{right}}, \Phi_{\text{mid}}, \Phi^{X_v}_{\text{end}}], \end{aligned}$$

where the second sum is over all those $\Phi_{\text{mid}} \in \mathcal{F}_{\mathsf{mid}(v)}$ that satisfy both

- $\Phi^{X_v}_{\text{start}}(i) = \Phi_{\text{mid}}(i)$ for all $i \notin X_{v_{\text{left}}}$ and
- $\Phi^{X_v}_{\text{end}}(i) = \Phi_{\text{mid}}(i)$ for all $i \notin X_{v_{\text{right}}}$.

It follows from (7.11) that all these summation indices Φ_{mid} satisfy $\iota(\Phi_{\text{mid}}(i)) = 1$ for all $i \in (\overline{X_v} \cap \overline{X_{v_{\text{left}}}}) \cup (\overline{X_v} \cap \overline{X_{v_{\text{right}}}})$, where the bar denotes the set complement. But $(\overline{X_v} \cap \overline{X_{v_{\text{left}}}}) \cup (\overline{X_v} \cap \overline{X_{v_{\text{right}}}}) = \overline{X_v \cup (X_{v_{\text{left}}} \cap X_{v_{\text{right}}})}$, which equals $\overline{X_v}$ by (7.1). This implies that $\Phi_{\text{mid}} \in \mathcal{F}^{X_v}_{\mathsf{mid}(v)}$. Therefore we can rewrite (7.13) as

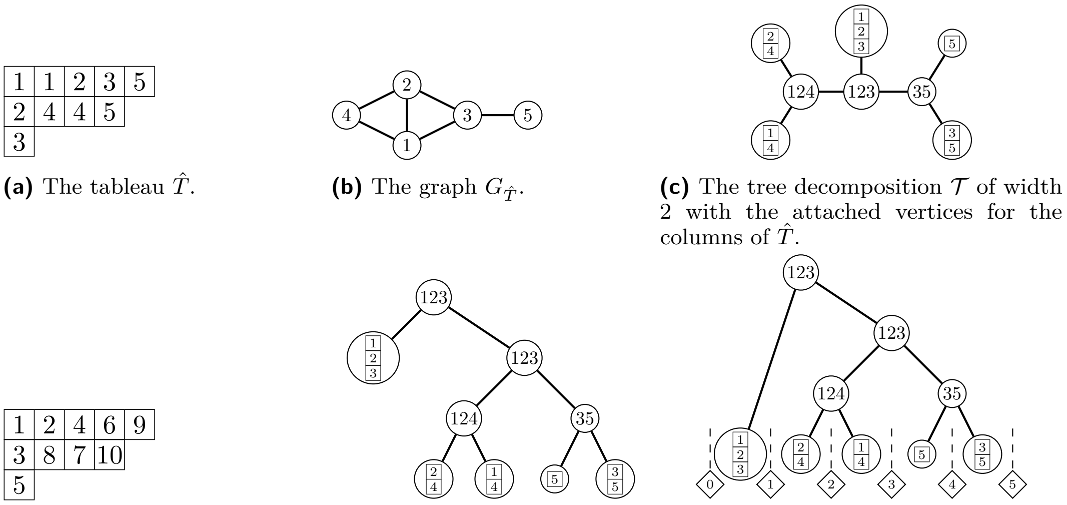
$$\begin{aligned} D[v, \Phi^{X_v}_{\text{start}}, \Phi^{X_v}_{\text{end}}] &= \sum_{\Phi_{\text{mid}} \in \mathcal{F}^{X_v}_{\mathsf{mid}(v)}} D[v_{\text{left}}, \Phi^{X_v}_{\text{start}}, \Phi_{\text{mid}}] \cdot D[v_{\text{right}}, \Phi_{\text{mid}}, \Phi^{X_v}_{\text{end}}] \\ &\overset{(7.12)}{=} \sum_{\Phi_{\text{mid}} \in \mathcal{F}^{X_v}_{\mathsf{mid}(v)}} \begin{pmatrix} 0 \;\text{ if }\; \iota(\Phi^{X_v}_{\text{start}}|_{\overline{X_{v_{\text{left}}}}}) \neq \iota(\Phi_{\text{mid}}|_{\overline{X_{v_{\text{left}}}}}), \\ D[v_{\text{left}}, (\Phi^{X_v}_{\text{start}})^{X_{v_{\text{left}}}}, \Phi^{X_{v_{\text{left}}}}_{\text{mid}}] \text{ otherwise} \end{pmatrix} \\ &\qquad \cdot \begin{pmatrix} 0 \;\text{ if }\; \iota(\Phi_{\text{mid}}|_{\overline{X_{v_{\text{right}}}}}) \neq \iota(\Phi^{X_v}_{\text{end}}|_{\overline{X_{v_{\text{right}}}}}), \\ D[v_{\text{right}}, \Phi^{X_{v_{\text{right}}}}_{\text{mid}}, (\Phi^{X_v}_{\text{end}})^{X_{v_{\text{right}}}}] \end{pmatrix} \end{aligned}$$

Using this equality we can compute the $|\mathcal{F}^{X_v}| \times |\mathcal{F}^{X_v}|$ matrix $D[v, \mathcal{F}^{X_v}, \mathcal{F}^{X_v}]$ as the product of two matrices of dimensions $|\mathcal{F}^{X_v}| \times |\mathcal{F}^{X_v}|$ whose entries can be computed recursively: They are either 0 or an entry in $D[w, \mathcal{F}^{X_w}, \mathcal{F}^{X_w}]$, where w is a child of v. The entries in $D[w, \mathcal{F}^{X_w}, \mathcal{F}^{X_w}]$ are not computed individually, but recursively as a product of matrices. For vertices that have only one child we use the assumption that they have the same bag, so that we can apply (7.7). Leaves are treated via (7.8).

If we can multiply two matrices of size $\leq k \times k$ in time $O(k^\omega)$, then the total running time to compute $D[r, \mathcal{F}^{X_r}, \mathcal{F}^{X_r}]$ is $w^{\omega(\tau+1)}\,\text{poly}(n, d, m, |\mathcal{T}|)$. Note that $D[r, \mathcal{F}^{X_r}, \mathcal{F}^{X_r}]$ is a 1×1 matrix whose entry is the desired $D[r, \overrightarrow{v_{\text{source}}}, \overrightarrow{v_{\text{sink}}}]$. ◀

▶ Remark 7.3. Even though only the size of the largest bag of the tree decomposition influences the asymptotic running time, it is advisable for an actual implementation of this algorithm to minimize the size of the individual bags. This can be achieved by removing each number from any bag which is not on a direct path between columns that contain that particular number or even splitting bags in some cases.

This dependency on the treewidth instead of n is significant, since for example the graphs of semistandard Young tableaux with only two rows are planar and thus have a treewidth of $O(\sqrt{n})$. Additionally this dependency is tight: we can construct semistandard Young tableaux with two rows and rectangular content which induce multigraph versions of the $n \times n$ grid-graphs and thus have treewidth $\Omega(\sqrt{n})$. We prove both these observations in Proposition 7.5.



(a) The tableau $\hat{T}$.

(b) The graph $G_{\hat{T}}$.

(c) The tree decomposition $\mathcal{T}$ of width 2 with the attached vertices for the columns of $\hat{T}$.

(d) The computed tableau T.

(e) The computation tree $\mathcal{T}'$ used for the dynamic program.

(f) $\mathcal{T}'$ with diamond separators for $0 \leq t \leq 5$.

Figure 2 An example execution of the preparation of the algorithm of Theorem 7.2 for the Young tableau $\hat{T}$.

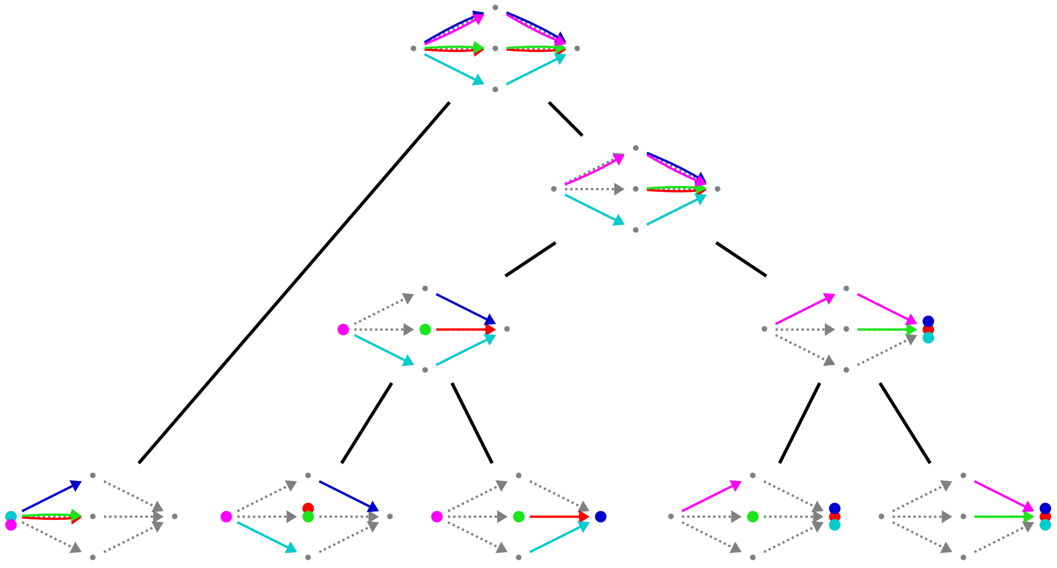


Figure 3 An example of an ABP of width 3 and depth 2 and the multi-walks that are obtained from a 5-tuple of s-t-paths (visible at the root) when decomposed according to Figure 2. Formally, the fat colored vertices correspond to principal vertices, even though they are drawn in different heights in layers.

Let p be given as a Waring rank decomposition of rank r. From this we can easily construct an ncABP of width $w = r$, in the same way we did to prove Theorem 4.2. Therefore the evaluation algorithm in Theorem 7.2 now takes time $O(w^{\omega(\tau+1)}) \cdot \mathrm{poly}(n, d, m) = O(r^{\omega(\tau+1)}) \cdot \mathrm{poly}(n, d, m)$. Comparing this to the naive algorithm in Remark 5.4, we get a faster evaluation in the case $\tau \in o(n)$, which for example is achieved for all semistandard tableaux with two rows which we will now prove.

As a first step, we will prove that in this case the corresponding graphs are always planar.

▶ **Proposition 7.4.** *Let S be a semistandard Young tableau with two rows. Then G_S is planar.*

Proof. Let S contain the numbers $\{1, \ldots, n\}$. We first start by constructing a different graph $G'_S = (L_S \dot{\cup} R_S, E'_S)$ which is a bipartite graph consisting of two copies of vertices $L_S = \{1_L, \ldots, n_L\}$, $R_S = \{1_R, \ldots, n_R\}$. L_S and R_S can be realized in the plane on two parallel vertical line segments, where the vertex indices increase from top to bottom. Now $\{i_L, j_R\} \in E'_S$ iff $\begin{array}{|c|}\hline i\\\hline j\\\hline\end{array}$ is a column in S. Here the vertical order in S matters, so due to S being semistandard we know $i < j$. We will now prove that G'_S is outerplanar and can be drawn with straight lines. So let $\{i, j\}, \{k, l\} \in E'_S$ be two different edges where the column $\begin{array}{|c|}\hline i\\\hline j\\\hline\end{array}$ appears to the left of the column $\begin{array}{|c|}\hline k\\\hline l\\\hline\end{array}$ in T. Due to T being semistandard this implies $i \leq k$ and $j \leq l$, which means those two edges do not cross. Since the edges were arbitrary no two edges intersect and G'_S is outerplanar.

Because both sets of vertices are ordered in ascending order we can now continuously rotate both vertex sets by 180 degrees and move them on top of each other, in this way unifying both copies of each vertex while still keeping the graph planar (the edges are not straight lines anymore, but they have the shape of a spiral). This resulting graph is precisely G_S, thus proving the claim. ◀

Now we can commence to prove the upper bound on the treewidth of Young tableaux with two rows. Additionally we prove that this bound is tight.

▶ **Proposition 7.5.**

1. *Let S_n be a semistandard Young tableau with two rows containing the numbers $\{1, \ldots, n\}$. Then G_{S_n} has treewidth at most $O(\sqrt{n})$.*
2. *Additionally there is a family (S'_n) of semistandard Young tableaux with two rows containing the numbers $\{1, \ldots, n\}$ exactly 4 times each and $G_{S'_n}$ having treewidth $\Omega(\sqrt{n})$.*

Proof. Let S_n be a semistandard Young tableau with 2 rows containing the numbers $\{1, \ldots, n\}$. Then G_{S_n} is a planar graph with n vertices by Proposition 7.4. The fact that planar graphs on n vertices have treewidth bounded by $O(\sqrt{n})$ follows directly from the famous planar excluded grid theorem [51].

We now prove the second part. W.l.o.g. we can restrict n to be of the form $(2k)^2$ with $k \in \mathbb{N} \setminus \{0\}$, since we can always extend the tableau without increasing the treewidth by appending four columns containing only a single cell with the number $i + 1$ to the end of S'_i to get S'_{i+1}. This change corresponds to adding a new isolated vertex to $G_{S'_i}$. We repeat this until $N = (2k)^2$, which scales n up by at most a factor of 8.

Every layered multigraph $G = (V, E)$ with the following properties is the graph G_S corresponding to some two row semistandard tableaux S where each number i appears exactly as often as the degree of i in G:

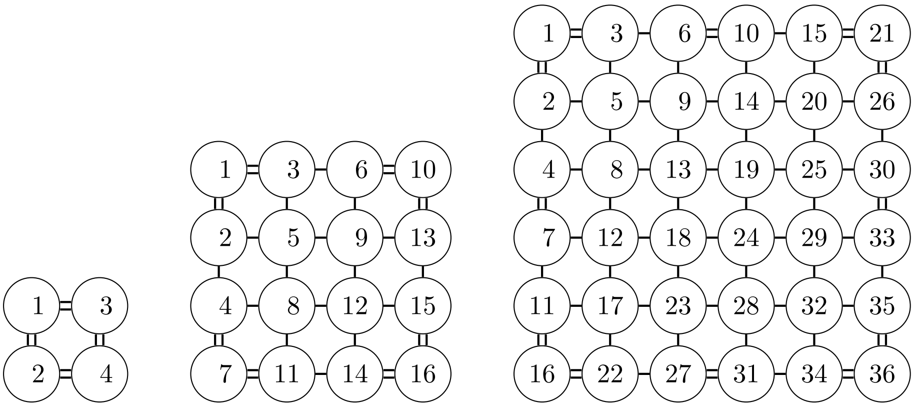


Figure 4 The grid graphs $\boxplus_2, \boxplus_4$ and $\boxplus_6$ after doubling the correct edges around the border and relabeling the vertices. The layers in $\boxplus_2$ are $\{1\}$, $\{2,3\}$, $\{4\}$. The layers in $\boxplus_4$ are $\{1\}$, $\{2,3\}$, $\{4,5,6\}$, $\{7,8,9,10\}$, $\{11,12,13\}$, $\{14,15\}$, $\{16\}$. The layers in $\boxplus_6$ are $\{1\}$, $\{2,3\}$, $\{4,5,6\}$, $\{7,8,9,10\}$, $\{11,12,13,14,15\}$, $\{16,17,18,19,20,21\}$, $\{22,23,24,25,26\}$, $\{27,28,29,30\}$, $\{31,32,33\}$, $\{34,35\}$, $\{36\}$.

1. $V = \{1,\ldots,n\}$
2. Edges in G only go from one layer to the next.
3. Edges between any two layers can be drawn with straight lines without crossing when the vertices in each layer are placed in ascending order.
4. All vertices in any layer j are labeled smaller than those in layer $j+1$ and each form a consecutive sequence of integers.

Some examples are provided in Figure 4. This can be shown constructively and separately for every pair of layers j and $j+1$. Since the edges between two layers are not crossing, there is a unique ordering on the set of edges from left to right. Adding columns corresponding to the edges in exactly this order to S forms exactly the wanted semistandard tableaux: For $\{u,v\} \in E$ we add the column $\begin{array}{|c|}\hline u\\\hline v\\\hline\end{array}$ to S. Thus the entries corresponding to layer j are only in the first row while those corresponding to layer $j+1$ only appear in the second row. Because of property (4) the columns of edges from layer j to layer $j+1$ can directly be concatenated to the columns of edges from layer $j+1$ to layer $j+2$ without violating the property of being semistandard. Clearly S contains each number i exactly once for each incident edge of i in G.

We now take the $2k \times 2k$ grid $\boxplus_{2k} = (V_{2k}, E_{2k})$ where

$$V_{2k} = \{(x,y) \mid x,y \in \{1,\ldots,2k\}\}$$
$$E_{2k} = \{\{(x_1,y_1),(x_2,y_2)\} \mid |x_1-x_2| + |y_1-y_2| = 1\}$$

This graph is known to have treewidth exactly $2k$ [23]. We now create a multigraph by doubling all the edges $\{(1,2i-1),(1,2i)\}, \{(2k,2i-1),(2k,2i)\}, \{(2i-1,1),(2i,1)\}$ and $\{(2i-1,2k),(2i,2k)\}$ for every $i \in \{1,\ldots,k\}$ which results in each vertex having degree exactly 4 while not changing the treewidth. To now apply the previous observations we now treat each diagonal $\{(x,y) \mid x+y = j+1\}$ as layer j and label them by increasing x, thus proving the claim of the lower bound. The resulting graphs are also visualized in Figure 4. ◀

▶ **7.6 Question.** *It is open whether the bound of $O(\sqrt{n})$ on the treewidth can be extended to any other constant number of rows, but starting at 3 rows G_S becomes non-planar*[7]*, so another approach to solving this problem would be needed. Additionally, if the number of rows is arbitrary G_S can contain an arbitrarily big clique, so it can have arbitrarily high treewidth. For example for any S with a first column with n distinct entries the graph G_S contains a clique on n vertices and thus has treewidth at least $n-1$.*

8 Hardness of evaluation

We will show that deciding whether a highest weight vector $f_{\hat{T}}$ of $\mathrm{Sym}^n\mathrm{Sym}^d\mathbb{C}^m$ vanishes at a point in $\mathrm{Sym}^d\mathbb{C}^m$ of Waring rank k for suiting parameters n, d, m, k is NP-hard. In particular we prove the NP-hardness of evaluating highest weight vectors given by Young tableaux with two rows in Theorem 8.1.

We can prove a similar – slightly weaker – result in Theorem 8.9, when the tableau $\hat{T}$ is restricted to be semistandard. In this case we have to increase the number of rows, the inner degree of the symmetric tensors and the Waring rank of the points of evaluation. Furthermore we don't prove hardness for all constant d in this case, but only for those divisible by 16. This still rules out polynomial evaluation algorithms which allow d to be part of the input under $\mathsf{P} \neq \mathsf{NP}$. These reductions also yield more explicit lower bounds under the exponential time hypothesis (ETH) in Theorems 8.1 and 8.9. As a reminder, the exponential time hypothesis states, that 3SAT can not be solved in time $2^{o(n)}$. Finally we show in Theorem 8.2 that if we want to calculate the exact value of the evaluation we can even prove #P-hardness for evaluating highest weight vectors given as Young tableaux.

Most of these reductions start with the same base that deciding whether a graph admits a proper 3-coloring a graph is NP-hard even when restricted to planar graphs of maximum degree 4. This was originally proven by Garey, Johnson and Stockmeyer [30] and a modified version can be found in Lemmas 8.6 and 8.7.

▶ **Theorem 8.1.** *Deciding whether a highest weight vector $f_{\hat{T}}$ of $\mathrm{Sym}^n\mathrm{Sym}^d\mathbb{C}^m$ given as a Young tableau $\hat{T}$ evaluates to zero at a fixed point $p = p_d \in \mathrm{Sym}^d\mathbb{C}^m$ of Waring rank 3 is* NP*-hard for constant $d \geq 8, m \geq 2$.*

Assuming ETH *no $2^{o(n)}$ algorithm for this evaluation can exist.*

Proof. We use the NP-hardness of 3-coloring graphs of maximum degree at most 4, see [30] or Lemma 8.6.

Let $G = (V, E)$ be a graph of maximum degree at most 4. Assume w.l.o.g. that $V = \{1, \ldots, n\}$. We now construct a Young tableau T with content $n \times d$ as follows: For every edge $\{u, v\} \in E$ we add two columns of the form $\begin{array}{|c|}\hline u \\ \hline v \\ \hline\end{array}$ to $\hat{T}$. Now for every vertex $v \in V$ add $d - 2 \cdot \deg(v)$ single-box columns $\boxed{v}$ to $\hat{T}$. It is easy to see that $\hat{T}$ has content $n \times d$ and is not necessarily semistandard.

We now choose to evaluate the highest weight vector $f_{\hat{T}}$ at $p = \ell_1^d + \ell_2^d + \ell_3^d$ with $\ell_1 = (1, 0, 0, \ldots), \ell_2 = (1, 1, 0, \ldots), \ell_3 = (1, 2, 0, \ldots) \in \mathbb{C}^m$. Note that the determinant of any two distinct linear forms of these is a real number, so its square is a positive real number.

[7] For example, G_S is the complete graph on 5 vertices for $S = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 3 & 3 \\ \hline 2 & 2 & 2 & 4 & 4 \\ \hline 3 & 4 & 5 & 5 & 5 \\ \hline\end{array}$.

Recall from (5.1) that

$$f_{\hat{T}}(p) = \sum_{\text{proper } \vartheta} \prod_{c=1}^{\lambda_1} \det{}_{\vartheta,c}.$$

We now show a 1-to-1 correspondence between summands of the evaluation and arbitrary – not necessarily proper – 3-colorings of G. A summand will be non-zero iff the corresponding 3-coloring is proper. Due to evaluating at p in its Waring decomposition, ϑ will be proper iff boxes with the same number j get assigned the same ℓ_i. We interpret this as vertex j receiving color i. Additionally every 3-coloring of G corresponds to some placement in this way.

We now take the product of determinants for each column. Since each column with two boxes is repeated twice, this product is a product of squares, and hence will always be positive iff none of the determinants is zero. This idea was first used in [12]. A determinant is non-zero iff different vectors ℓ_i and ℓ_j are chosen for both of the boxes, corresponding to coloring both vertices of this column with different colors. So a summand will be non-zero iff ϑ corresponds to a proper 3-coloring of G.

Note that any algorithm deciding whether $f_{\hat{T}}(p)$ is non-zero in time $2^{o(n)}$ can now be used to decide whether G allows for a proper 3-coloring in time $\text{poly}(|V|)2^{o(|V|)}$ which is a contradiction unless ETH fails as proven in [37]. ◀

Note that our algorithms for evaluation described in Theorems 6.2 and 7.2 both achieve a running time of $2^{O(n)}$ for evaluations at points of constant Waring rank with constant m and d. So Theorem 8.1 gives a matching lower bound under ETH.

The proof for #P-hardness is pretty similar and reduces from counting the number of 3-colorings of a graph with maximum vertex degree 3 which is known to be #P-complete [10]. The main idea is to use a more carefully chosen point of evaluation to ensure that every summand that corresponds to a proper 3-coloring will be exactly 1.

▶ **Theorem 8.2.** *Evaluating a highest weight vector $f_{\hat{T}}$ of $\text{Sym}^n\text{Sym}^d\mathbb{C}^m$ given as a Young tableau $\hat{T}$ at a point $p \in \text{Sym}^d\mathbb{C}^m$ of Waring rank 3 is #P-hard for constant $d \geq 18, m \geq 2$.*

Proof. We reduce from counting the number of 3-colorings of a graph $G = (V, E)$ where every vertex has degree at most 3 which is known to be #P-complete [10]. We proceed in a similar manner as in the NP-hardness proof in Theorem 8.1. We construct $\hat{T}$ by adding the columns $\begin{array}{|c|}\hline u \\ \hline v \\ \hline \end{array}$ for $\{u, v\} \in E$ 6-times each and for every vertex $v \in V$ add $d - 6 \cdot \deg(v)$ columns $\boxed{v}$ to $\hat{T}$. This time we evaluate $f_{\hat{T}}$ at $p = \ell_1^d + \ell_2^d + \ell_3^d$ with $\ell_1 = (1, 0, 0, \ldots), \ell_2 = (1, e^{\frac{i\pi}{3}}, 0, \ldots), \ell_3 = (1, e^{\frac{2i\pi}{3}}, 0, \ldots) \in \mathbb{C}^m$. Note that the determinant of any two distinct linear forms of these is a 6-th root of unity, so its 6-th power is always exactly 1. If we now analyse the summands of the evaluation again we see that each term contributes exactly 1 if it corresponds to a proper 3-coloring and 0 otherwise. Thus the evaluation $f_{\hat{T}}(p)$ counts exactly the number of 3-colorings of G. ◀

Extending this result to semistandard Young tableaux now proceeds in multiple steps, which we devote the rest of this section towards.

We first extend the NP-hardness of 3-coloring to a subclass of planar graphs which we call *grid-like layered* graphs. More specifically we prove NP-hardness for 8-regular, i.e. each vertex has degree exactly 8, grid-like layered graphs in Lemma 8.5, while we show a lower bound of $2^{o\left(\sqrt{|V|}\right)}$ under ETH using Lemma 8.8.

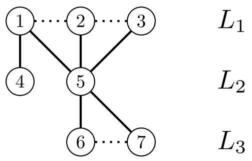


Figure 5 Example of a grid-like layered graph with 3 layers. Edges between layers are drawn as solid lines and edges inside layers as dotted lines.

▶ **Definition 8.3.** *We call a planar multigraph $G = (V, E)$* grid-like layered *if there are disjoint layers $L_1, \dots, L_k \subseteq V$ of vertices and an embedding $e : V \to \mathbb{N} \times \{1, \dots, k\}$, s.t.*

1. *e is injective.*
2. *For every $i \in \{1, \dots, k\}$ we have $e^{-1}(\mathbb{N} \times \{i\}) = L_i$*
3. *Edges between layers only exist between layer L_i and L_{i+1} for all $i \in \{1, \dots, k-1\}$.*
4. *Edges inside layers only exist for vertices $v, u \in L_i$ where $e(v) = e(u) \pm (1, 0)$ for some $i \in \{1, \dots, k\}$.*
5. *All edges can be drawn as straight lines without crossing when vertices are placed according to e in $\mathbb{R}^2$ and the graph is treated as being simple.*
6. *Every vertex has a neighbour in a different layer.*

Note that grid-like layered graphs are not necessarily subgraphs of a grid-graph, see Figure 5 for an example. The crucial property about grid-like layered graphs is, that they can be decomposed into two graphs over the same vertices each corresponding to a semistandard Young tableau with two rows. This decomposition is essential to encode the 3-coloring of such graphs into a single combined semistandard Young tableau.

Recall the definition of the graph $G_{\hat{T}}$ for a semistandard tableau $\hat{T}$ from Section 7.

▶ **Lemma 8.4.** *Let $G = (V, E)$ be a grid-like layered graph. Then $G = (V, E(G_{\hat{T}_{\leftrightarrow}}) \cup E(G_{\hat{T}_{\updownarrow}}))$ for two semistandard tableaux $\hat{T}_{\leftrightarrow}, \hat{T}_{\updownarrow}$ for some relabeling of the vertices V. Additionally $\hat{T}_{\updownarrow}$ contains every number from 1 to $|V|$ at least once.*

Proof. Let e be the embedding of G. We relabel the vertices in increasing order inside each layer according to e and then increasing order from layer L_i to layer L_{i+1} for every i, like in Figure 5. Let $E_{\updownarrow}$ now be the edges between different layers and $E_{\leftrightarrow}$ those inside the layers, see Figure 6. Clearly $E_{\leftrightarrow} \cup E_{\updownarrow} = E$ and every vertex is incident to some edge in $E_{\updownarrow}$ by condition 8.3.6, so if we can construct semistandard tableaux $\hat{T}_{\leftrightarrow}, \hat{T}_{\updownarrow}$ with $E_{\leftrightarrow} = E(G_{\hat{T}_{\leftrightarrow}})$ and $E_{\updownarrow} = E(G_{\hat{T}_{\updownarrow}})$ we are done.

We start with $\hat{T}_{\updownarrow}$. Since the labeling of the vertices is increasing from one layer to the next it suffices to show that we can create $\hat{T}_{\updownarrow}$ for a single pair of consecutive layers and afterwards concatenate them. Condition 8.3.5 gives a unique order of the edges between these layers from left to right. So for the edge $\{u, v\} \in E_{\updownarrow}$ with $u < v$ we add the column $\begin{array}{|c|}\hline u\\\hline v\\\hline\end{array}$ to $\hat{T}_{\updownarrow}$. Assume two columns $\begin{array}{|c|}\hline u\\\hline v\\\hline\end{array}$ and $\begin{array}{|c|}\hline u'\\\hline v'\\\hline\end{array}$ would violate the semistandard property. Then either $u' < u$ in which case the edge $\{u', v'\}$ would start left of $\{u, v\}$ or $v' < v$ in which case the edge $\{u', v'\}$ would end left of $\{u, v\}$, both a contradiction to our unique ordering from left to right. So $\hat{T}_{\updownarrow}$ is semistandard.

We continue with $\hat{T}_{\leftrightarrow}$. Again we only have to consider $\hat{T}_{\leftrightarrow}$ for a single layer as we can just concatenate the resulting tableaux afterwards. If we direct the edges in $E_{\leftrightarrow}$ to only go from the smaller vertex to the larger one we see with condition 8.3.4 that each vertex can

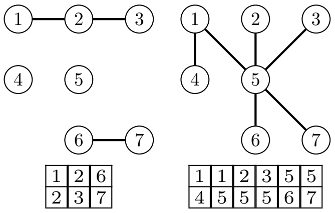


Figure 6 Example of $\hat{T}_{\leftrightarrow}$ and $\hat{T}_{\updownarrow}$ and the two graphs $G_{\hat{T}_{\leftrightarrow}}$ and $G_{\hat{T}_{\updownarrow}}$ according to Lemma 8.4 for the grid-like layered graph given in Figure 5.

only be the first vertex of an edge once, and those edges have the form $\{v, v+1\}$. So the only columns in $\hat{T}_{\leftrightarrow}$ are of the form $\begin{array}{|c|}\hline v\\\hline v{+}1\\\hline\end{array}$. Those can clearly just be combined in order to make $\hat{T}_{\leftrightarrow}$ semistandard.

Note that since G is a multigraph we add every column to the tableaux k times if the edge appears with multiplicity k in G. ◀

We can now give an elegant proof of the NP-hardness of deciding whether a given 8-regular grid-like layered graph $G = (V, E)$ admits a proper 3-coloring. With this elegance comes the caveat, that this proof only yields a lower bound of $2^{o\left(\sqrt[4]{|V|}\right)}$ under ETH, which we improve to $2^{o\left(\sqrt{|V|}\right)}$ with a more technical proof in Lemma 8.8.

For this we need the notion of a graph minor model. We call a collection of subsets of vertices $(V_h)_{h\in V(G)}$ a *graph minor model* of embedding a graph G into a graph H if the $V_h \subseteq V(H)$ induce disjoint non-empty connected subgraphs of H for every $h \in V(G)$ and if for every edge $\{u, v\} \in V(G)$ there is an edge between some vertices of V_u and V_v. See [23, Section 6.3] for a more detailed introduction to graph minors.

▶ **Lemma 8.5.** *Deciding whether a given graph $G = (V, E)$ admits a proper 3-coloring is* NP*-hard, even if the graph is restricted to be grid-like layered and 8-regular.*

Unless ETH fails, 3-coloring doesn't admit an $2^{o\left(\sqrt[4]{|V|}\right)}$ time algorithm for grid-like layered graphs.

Proof. For this we reduce from the decision problem whether a planar graph G admits a proper 3-coloring.

In order to achieve this we find a graph minor model $(V_h)_{h\in V(G)}$ of embedding G into a grid $\boxplus$ with $O(|V(G)|^2)$ vertices in linear time [55]. Let G_1 be the grid $\boxplus$ after removing any vertices and edges which do not correspond to vertices or edges in G, i.e.

$$V(G_1) = \bigcup_{h\in V(G)} V_h$$

and

$$E(G_1) = \bigcup_{h\in V(G)} E(\boxplus[V_h]) \cup \bigcup_{uv\in E(G)} E(\boxplus[V_u \cup V_v])$$

where $\boxplus[V]$ denotes the subgraph of $\boxplus$ induced by the vertices V.

We can now transform any 3-coloring of G into a 3-coloring of G_1 by coloring every vertex in V_h with the same color as h for every $h \in V(G)$. The property of a coloring of G being proper now translates to enforcing that for each $h \in V(G)$ all the vertices inside the component V_h are colored with the same color and vertices in neighbouring components V_u, V_v for $\{u, v\} \in E(G)$ are colored with different colors.

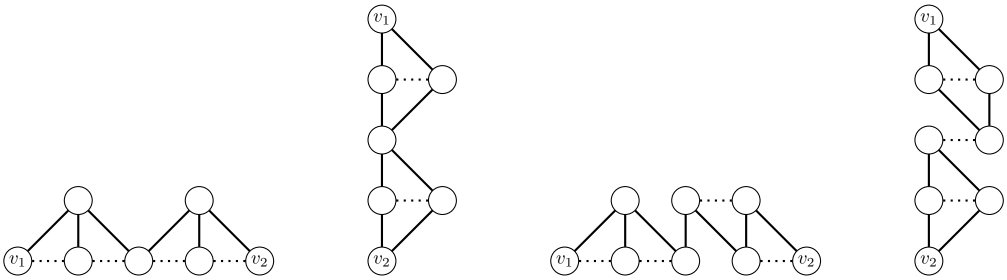


Figure 7 The equality and inequality gadgets $H_1^=, H_2^=$ and $H_1^{\neq}, H_2^{\neq}$ used in Lemma 8.5.

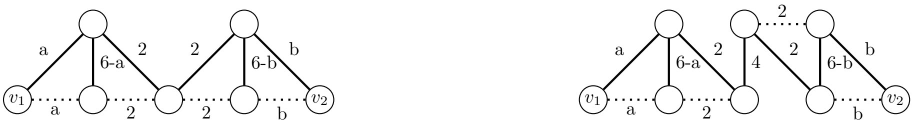


Figure 8 The nearly 8-regular versions of the gadgets $H_1^=$ and $H_1^{\neq}$ used in Lemma 8.5. The edge labels denote the multiplicity of the edges in the multigraph and $2a = \deg(v_1)$ and $2b = \deg(v_2)$.

In order to enforce these constraints on G_1 we construct a new graph G_2 by replacing each edge inside any V_h by the equality gadgets $H_1^=$ or $H_2^=$ and replacing each edge between neighbouring components V_u, V_v by the inequality gadgets $H_1^{\neq}$ or $H_2^{\neq}$. These gadgets are shown in Figure 7. If an edge is horizontal in the canonical embedding of G_1 into the plane we choose variant 1 of the gadgets. If an edge is vertical we choose variant 2.

It can be easily checked that the only way to properly 3-color these gadgets is such that the colors of v_1 and v_2 are the same for the equality gadgets and different for the inequality gadgets.

Clearly G is now properly 3-colorable iff G_2 is properly 3-colorable.

Secondly all those gadgets are designed as grid-like layered graphs. It can be easily checked that replacing all edges in a subgraph of a grid yields a grid-like layered graph, so G_2 is grid-like layered.

So the only thing remaining to do is make the graph 8-regular by adding copies of existing edges to the graph. In order to achieve this it is sufficient to show that multigraph versions of $H_1^=, H_2^=, H_1^{\neq}$ and $H_2^{\neq}$ exist which are 8-regular except for the vertices v_1 and v_2, which can independently have a degree of $2, 4, 6$ or 8 each. This is sufficient since every vertex of the grid graph has a degree between 1 and 4, so it has between 1 and 4 of these gadgets attached to it. The multigraph variations of the gadgets are shown for $H_1^=$ and $H_1^{\neq}$ in Figure 8, for the other two gadgets these are constructed similarly.

Note that $\boxplus$ has $O(|V(G)|^2)$ many vertices so we can conclude that G_2 also has $O(|V(G)|^2)$ many vertices. If we can decide whether the 8-regular grid-like layered graph G_2 allows for a proper 3-coloring in time $2^{o\left(\sqrt[4]{|V(G_2)|}\right)}$ we can decide via this reduction whether the planar graph G allows for a proper 3-coloring in time $\mathrm{poly}(|V(G)|) \cdot 2^{o\left(\sqrt{|V(G)|}\right)}$. This contradicts that planar 3-coloring is not solveable in time $2^{o\left(\sqrt{|V(G)|}\right)}$ unless ETH fails which was essentially observed by Cai and Juedes [18] and is also mentioned in [23, Theorem 14.9]. ◀

Looking at the reduction from 3-satisfiability to 3-coloring more closely we can improve the ETH lower bound of the previous proof. The fourth root was necessary because first embedding the 3SAT formula into a planar graph and then into a grid graph each resulted in quadratic blow-up. By abusing the structure of the intermediate graphs more closely we reduce the size of the grid graph to be only quadratic in the number of variables of the 3SAT formula and thus show a better lower bound in Lemma 8.8.

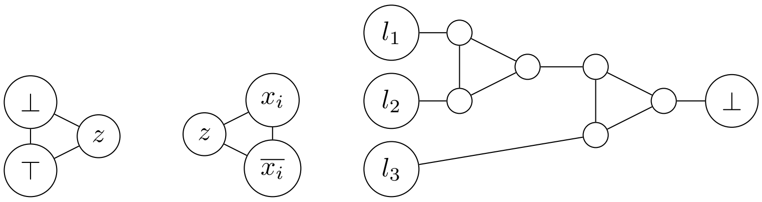


Figure 9 The gadgets H_1, H_2 and H_3 used in the proof of Lemma 8.6.

The proof uses similar gadgets to the standard textbook reduction of 3SAT to 3-coloring, which we show again for reference.

▶ **Lemma 8.6.** *Deciding whether a given graph $G = (V, E)$ admits a proper 3-coloring is* NP*-hard.*

Proof. We reduce from 3-satisfiability. So let $\phi = C_1 \wedge \ldots \wedge C_m$ be a formula in 3-CNF on n variables $x_1, \ldots, x_n$. We construct a graph G as follows. We start with the graph H_1 shown in Figure 9 (left) and call the three vertices $\top$, $\perp$, and z. Then for each $1 \leq i \leq n$ we add a vertex x_i and a vertex $\overline{x_i}$ and add three edges: $\{x_i, \overline{x_i}\}$, $\{x_i, z\}$, $\{\overline{x_i}, z\}$. This is depicted in Figure 9 (middle). For each $1 \leq j \leq m$ we now add 6 vertices and connect them with the existing vertices as shown in Figure 9 (right): The vertices labeled l_1, l_2, l_3 in the figure stand for the vertices corresponding to the three literals (elements in $\{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$) in the clause C_j.

We now analyze potential proper 3-colorings of G. Our colors will conveniently be called $\top$, $\perp$, and z and we assume from now on w.l.o.g. that the three vertices in H_1 are colored according to their names. It follows from H_2 that in every proper 3-coloring the vertices corresponding to literals are colored with $\top$ or $\perp$, but never with z. It is easy to see that H_3 has no proper 3-coloring if l_1, l_2, and l_3 all are colored with $\perp$. Moreover, if at least one of l_1, l_2, and l_3 is colored with $\top$ and the others are colored with $\perp$, then a proper 3-coloring of H_3 exists.

Hence from a proper 3-coloring of G we can easily reconstruct a satisfying assignment of ϕ and vice versa. ◀

In Lemma 8.5 we then proceeded with a planar version of this theorem due to [30] and embedded these resulting graphs as minors of a grid. In essence we used a variant of 3-coloring where the graph is a subset of a grid graph and every edge can either be an equality or inequality edge, i.e. vertices connected by an equality edge have to be colored by the same color and vertices connected by an inequality edge have to be colored with different colors. We already implicitly showed NP-hardness of this variant which we call *relational* 3*-coloring on subgraphs of grids* in the proof of Lemma 8.5.

Note that equality edges are a necessity, since any subgraph of a grid graph is bipartite and thus can be 2-colored.

▶ **Lemma 8.7.** *Unless ETH fails, relational* 3*-coloring on subgraphs G of grids can not be solved in time $2^{o\left(\sqrt{|V(G)|}\right)}$.*

Proof. We reduce from 3-satisfiability. So let $\phi = C_1 \wedge \ldots \wedge C_m$ be a formula in 3-CNF on n variables $x_1, \ldots, x_n$.

We again start with the color choosing gadget H_1' from Figure 10 and assume that each vertex of H_1' is colored with its label to simplify the analysis. Note that vertices with the same labels will be connected by a path of equality edges, so they have the same color in each proper 3-coloring. H_1' forms a border of width ≤ 2 around the rest of the graph.

Connected to the vertices labeled with z are the variable gadgets H'_2. The vertices with labels x_i and $\overline{x_i}$ corresponding to the literals of ϕ appear exactly as often as each of the literals appears in ϕ. Connected to the bottom vertices $\perp$ are the clause gadgets H'_3. Both of these gadgets can be found in Figure 10.

The only thing left is connecting the vertices corresponding to literals in the clause gadgets to those in the variable gadgets via an equality edge. Unfortunately this would make the graph not be a subgraph of a grid, so we need the crossing gadget H'_4 from Figure 11 which is an embedding of the crossing gadget used in [30]. In H'_4 the vertices pairs labeled a, a' and b, b' each have the same color in every proper 3-coloring. Additionally there is a proper 3-coloring for every choice of colors of a and b.

We now need to "sort" the vertices corresponding to literals into the order $(l_{1,1}, l_{1,2}, l_{1,3}, l_{2,1}, l_{2,2}, l_{2,3}, \dots, l_{m,1}, l_{m,2}, l_{m,3})$ where $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$. We do this via an iterative procedure. We add the crossing gadget H'_4 between every two consecutive vertices l_i, l_j which are in the wrong order in each step. In case some vertices could be part of multiple swaps choose the pairs in a way that maximizes the number of possible swaps per iteration. We connect l_i to the vertex a of H'_4 via an equality edge and similarly l_j to b. The vertices a' and b' now form the next step in the ordering process and have essentially swapped adjacent l_i and l_j. All the vertices l_i which do not change their position will be extended via paths made out of equality edges to be on the same layer as the outlets of the crossing gadgets. After at most $O(m)$ of these steps the vertices are sorted in our desired order and can be directly connected to the corresponding vertices of the clause gadgets.

We call this resulting graph G. Note that H'_4 enforces a finer subdivision of the grid than H'_1, H'_2 and H'_3, but we can always split an equality edge into two equality edges connected by a vertex or split vertices into two vertices connected by an equality edge to stretch these gadgets, so G is a subgraph of a grid graph.

It can be easily checked that H'_1, H'_2 and H'_3 behave in exactly the same way as their counterparts in the proof of Lemma 8.6, so the correctness of this reduction can easily be seen with the same reasoning as there together with the properties of H'_4.

G is a subgraph of an $O(m) \times O(m)$ grid, so $|V(G)| = O(m^2)$. If we could decide relational 3-colorability on subgraphs of grids in size $2^{o\left(\sqrt{|V(G)|}\right)}$ we could thus decide 3-satisfiability in time $2^{o(m)}$ which is a contradiction unless ETH fails, see [37] for this lower bound for 3-satisfiability. ◀

▶ **Lemma 8.8.** *Unless ETH fails, 3-coloring doesn't admit an $2^{o\left(\sqrt{|V|}\right)}$ time algorithm for 8-regular grid-like layered graphs $G = (V, E)$.*

Proof. We reduce from relational 3-coloring on subgraphs of grids. Let $G = (V, E)$ be a subgraph of a grid. We proceed in the same way as Lemma 8.5 did except that $\boxplus$ is replaced by G, each equality edge is replaced by the corresponding equality gadget and each inequality edge is replaced by the corresponding inequality gadget.

Note that the obtained graph G_2 now has $O(|V(G)|)$ many vertices. If we can decide whether the 8-regular grid-like layered graph G_2 allows for a proper 3-coloring in time $2^{o\left(\sqrt{|V(G_2)|}\right)}$ we can decide via this reduction whether G allows for a relational 3-coloring in time $\mathrm{poly}(|V(G)|) \cdot 2^{o\left(\sqrt{|V(G)|}\right)}$, contradicting Lemma 8.7 unless ETH fails. ◀

We now have all the necessary intermediate results to prove that even evaluation of highest weight vectors given by semistandard tableaux is NP-hard. We use the same general idea of coloring the cells of the Young tableau s.t. all cells with the same number receive the

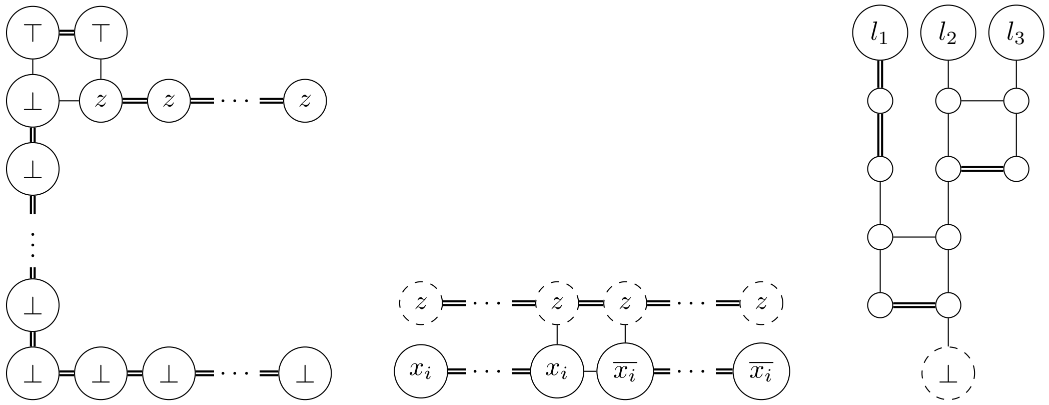


Figure 10 The gadgets H_1', H_2' and H_3' used in the proof of Lemma 8.7. Double lines denote equality edges and single lines denote inequality edges. Vertices corresponding to other gadgets are visualized with dashed outline to show how to connect the gadgets.

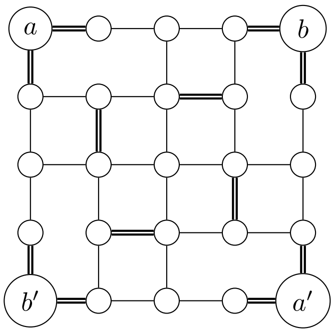


Figure 11 The gadget H_4' used in the proof of Lemma 8.7. Double lines denote equality edges and single lines denote inequality edges.

same color. Additionally each column of the Young tableau has to be repeated often enough that any summands are guaranteed to be positive iff each column is colorful, i.e. does not contain any color multiple times and zero otherwise.

▶ **Theorem 8.9.** *The evaluation of highest weight vectors $f_{\hat{T}}$ of $\mathrm{Sym}^n\mathrm{Sym}^d\mathbb{C}^m$ is* NP*-hard for any constant $d \geq 16$ with $16 \mid d$ and $m \geq 5$, when $f_{\hat{T}}$ is given as a semistandard Young tableau $\hat{T}$. This even holds if evaluation is restricted to points of Waring rank 5 and the algorithm only has to decide whether the evaluation is non-zero.*

Additionally this evaluation can not be computed in time $2^{o(\sqrt{n})}$ unless ETH fails.

Proof. We reduce from checking whether an 8-regular grid-like layered graph allows for a proper 3-coloring which was proven in Lemma 8.5 to be NP-hard.

Let $G = (V, E_{\leftrightarrow} \cup E_{\updownarrow})$ be an 8-regular grid-like layered graph where $E_{\leftrightarrow}$ denotes the edges inside layers and $E_{\updownarrow}$ denotes edges between layers. W.l.o.g. the vertex set V are the numbers $1, \ldots, |V|$ assigned in a layer by layer and left to right fashion, given by the embedding of G. To ease the description of the constructed semistandard tableaux $\hat{T}$ we will describe it in 5 parts $T_1, \ldots, T_5$ over the symbolic entries a_i, b_i, c_i, d_i, e_i. For better readability we will colorcode each of the symbolic entries in the constructions of this theorem. The point of evaluation is now $p = \sum_{i=1}^{5} \ell_i^d$ with $\ell_i = (1, i, i^2, i^3, \ldots, i^m)$ Then any determinants arising in the evaluation are determinants of Vandermonde matrices and thus are well known to be non-zero.

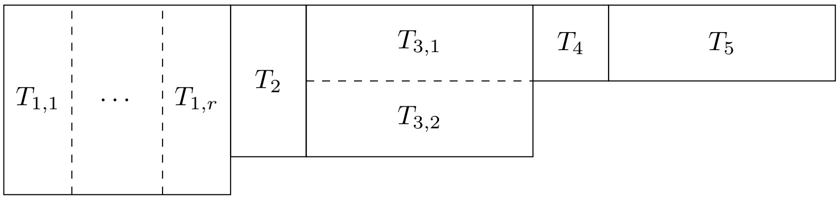


Figure 12 The general structure of the 5 row Young tableau $\hat{T}$ constructed in Theorem 8.9.

p is a point of Waring rank 5, so analogously to Theorem 8.1 the summands of the evaluation will consist of assigning one of the 5 linear forms to each number and will be non-zero iff no column contains the same linear form twice. Since all vectors are real, any occuring determinants in the evaluation will also be real and hence every summand will be either 0 or positive due to every column being repeated an even number of times.

The general structure of $\hat{T}$ can be seen in Figure 12 and we will first describe the main idea of each part. The parts T_3 and T_5 both encode the actual 3-coloring restrictions of the edge sets $E_{\updownarrow}$ and $E_{\leftrightarrow}$ respectively, in the entries e_i in the same way as in Theorem 8.1. To ensure that only 3 colors can be used for the graph coloring, the entries c_i are added into T_3 to use up the remaining colors. The consistency of the c_i, i.e. that exactly two colors are used by all of the c_i, is then ensured in T_1 with the help of the entries a_i and b_i. Everything else, i.e. the d_i and the tableaux T_2 and T_4 are only used in order to make $\hat{T}$ semistandard and with rectangular content.

$\hat{T}$ is then the concatenation $T_1T_2 \dots T_5$ where we assign increasing numbers starting from 1 first to all the a_i, then to all the b_i, c_i, d_i and e_i in order, increasing inside each group of symbolic entries with increasing index. This ensures that each of the T_i individually, but also the concatenation $\hat{T}$ will be semistandard. The latter can be seen by looking at the symbolic entries at the left and right borders of the T_i in the following descriptions.

We first describe the construction of all the T_i: T_1 is built as a concatenation of smaller tableaux $T_{1,1}, \dots T_{1,r}$ for $r = \lceil \frac{|E_{\updownarrow}|-1}{24} \rceil$. The construction of $T_{1,1}$ and of $T_{1,i}$ for $1 < i \leq r$ can be seen in Figure 13. T_2 and T_4 are always the same and are given in Figure 14. T_3 is given as a left aligned vertical concatenation of $T_{3,1}$ and $T_{3,2}$, where $T_{3,1}$ consists of the columns $\begin{array}{|c|}\hline c_1\\\hline c_2\\\hline\end{array}$, $\begin{array}{|c|}\hline c_3\\\hline c_4\\\hline\end{array}$, $\dots$, $\begin{array}{|c|}\hline c_{8r-3}\\\hline c_{8r-2}\\\hline\end{array}$ each repeated 12 times and $\begin{array}{|c|}\hline c_{8r-1}\\\hline c_{8r}\\\hline\end{array}$ repeated 14 times. $T_{3,2}$ is obtained from $\hat{T}_{\updownarrow}$ of Lemma 8.4 by doubling every column. Lastly T_5 is constructed in the exact same way as $T_{3,2}$, but is obtained from $\hat{T}_{\leftrightarrow}$ of Lemma 8.4.

We first prove that T is a semistandard Young tableau of rectangular content. T_1 fulfils the following properties which are easy to prove via induction:

- $a_1, \dots, a_{3r}$ all appear exactly 16 times each in T_1.
- b_1 and b_2 appear exactly twice in T_1.
- $c_1, \dots, c_{8r-2}$ all appear exactly 4 times in T_1.
- c_{8r-1} and c_{8r} appear exactly twice in T_1.
- If we replace the symbolic entries as previously described then T_1 is semistandard.

The only important properties of T_2 and T_4 are that b_1, b_2, d_1, d_2 all appear exactly 14 times in T_2 and d_1 and d_2 each appear twice in T_4, while both are clearly semistandard.

The properties of T_3 are now:

- $c_1, \dots, c_{8r-2}$ all appear 12 times in T_3.
- c_{8r-1} and c_{8r} appear exactly 14 times in T_3.
- If we replace the symbolic entries as previously described, T_3 is semistandard.

- $T_{3,1}$ has at least as many columns as $T_{3,2}$ by our choice of $r = \lceil \frac{|E_\updownarrow|-1}{24} \rceil$. $T_{3,1}$ has

$$(4r-1) \cdot 12 + 14 \geq \left(\frac{|E_\updownarrow| - 1}{6} - 1 \right) \cdot 12 + 14 = 2 \cdot |E_\updownarrow|$$

  columns while $T_{3,2}$ has exactly $2 \cdot |E_\updownarrow|$ columns.

The last property of T_3 is important in order for T_3 and thus $\hat{T}$ to be a of proper shape for a Young tableau, i.e. have non-decreasing row lengths.

Combining all the properties we see that $\hat{T}$ contains every entry exactly 16 times each and is semistandard after replacing the symbolic entries. Additionally each column is repeated an even number of times, so no summands of the evaluation can be negative. In case $d > 16$ we repeat every column of T $\frac{d}{16}$ times in order to get the representation of a highest weight vector of $\mathrm{Sym}^n \mathrm{Sym}^d \mathbb{C}^m$ as a semistandard Young tableau.

Next we look at the effects of the gadgets on the possible non-zero summands of the evaluation.

Any further considerations will now assume w.l.o.g. that a_1, a_2, a_3 get assigned the first three linear forms of p, all other cases are symmetric. These three entries all occur together in the very first column of T_1, so they have to be pairwise different in order to be part of a non-zero summand. $T_{1,1}$ then enforces $c_1, \dots, c_8$ to all be assigned the last two linear forms of p. Since $T_{1,i}$ and $T_{1,i+1}$ share the entries of c_{8i-1} and c_{8i}, inductively all of $a_i, \dots, a_{3r}$ will be assigned the first three linear forms of p in some order and all of $c_1, \dots, c_{8r}$ will be assigned the last two linear forms of p in some order.

The last important property is, that $e_1, \dots, e_{|V|}$ all appear at least once in T_3 since every vertex of a grid-like layered graph is incident to an edge going to another layer. This means that all the linear forms being chosen for any $e_1, \dots, e_{|V|}$ can only be the first three linear forms of p since the remaining two are already used for the c_i of which two appear in every column.

Now assume G admits a proper 3-coloring with the colors $1, 2, 3$. We can now construct a placement of the linear forms onto the entries of $\hat{T}$ as follows:

- The entries a_{3i+j} get assigned the linear form ℓ_j for every $i \in \{0, \dots, r-1\}$ and $j \in \{1, 2, 3\}$.
- The entries b_1 and b_2 get assigned the linear forms ℓ_4 and ℓ_5 respectively.
- The entries c_{2i+j} get assigned the linear form ℓ_{3+j} for every $i \in \{0, \dots, 4r-1\}$ and $j \in \{1, 2\}$.
- The entries d_1 and d_2 get assigned the linear forms ℓ_1 and ℓ_2 respectively.
- The entries e_i get assigned the linear form ℓ_j if vertex i was colored with color j in G.

It is now easy to check that in T_1, T_2 and T_4 no column contains any linear form twice. To see that the same holds for T_3 and T_5 note that the only way any column could contain the same linear form twice would be for two entries e_u and e_v to appear in the same column and be assigned the same linear form. That would mean that u and v got colored the same in G, but by our construction there is also an edge $\{u, v\} \in E_\updownarrow \cup E_\leftrightarrow$, a contradition to G being properly 3-colored. Since no column contains a repeated linear form this summand is strictly positive, making the whole evaluation $f_{\hat{T}}(p)$ non-zero.

Conversely assume that the evaluation of $f_{\hat{T}}(p)$ is non-zero. Thus there must be a non-zero summand, placing linear forms on each entry. As by the previous discussion there are only 3 different linear forms being placed on all of the e_i, directly inducing a 3-coloring of G. This 3-coloring is proper since every column can never contain the same linear form twice and every edge of G is represented by a column.

$$T_{1,1} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 & a_1 \\ \hline a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 & a_2 \\ \hline a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 & a_3 \\ \hline b_1 & b_1 & c_1 & c_1 & c_1 & c_1 & c_3 & c_3 & c_3 & c_3 & c_5 & c_5 & c_5 & c_5 & c_7 & c_7 \\ \hline b_2 & b_2 & c_2 & c_2 & c_2 & c_2 & c_4 & c_4 & c_4 & c_4 & c_6 & c_6 & c_6 & c_6 & c_8 & c_8 \\ \hline \end{array}$$

$$T_{1,i} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} & a_{3i-2} \\ \hline a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} & a_{3i-1} \\ \hline a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} & a_{3i} \\ \hline c_{8i-9} & c_{8i-9} & c_{8i-7} & c_{8i-7} & c_{8i-7} & c_{8i-7} & c_{8i-5} & c_{8i-5} & c_{8i-5} & c_{8i-5} & c_{8i-3} & c_{8i-3} & c_{8i-3} & c_{8i-3} & c_{8i-1} & c_{8i-1} \\ \hline c_{8i-8} & c_{8i-8} & c_{8i-6} & c_{8i-6} & c_{8i-6} & c_{8i-6} & c_{8i-4} & c_{8i-4} & c_{8i-4} & c_{8i-4} & c_{8i-2} & c_{8i-2} & c_{8i-2} & c_{8i-2} & c_{8i} & c_{8i} \\ \hline \end{array}$$

Figure 13 The Young tableaux $T_{1,1}$ and $T_{1,i}$ from the proof of Theorem 8.9.

$$T_2 = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 & b_1 \\ \hline b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 & b_2 \\ \hline d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 & d_1 \\ \hline d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 & d_2 \\ \hline \end{array} \qquad T_4 = \begin{array}{|c|c|}\hline d_1 & d_1 \\ \hline d_2 & d_2 \\ \hline \end{array}$$

Figure 14 The Young tableaux T_2 and T_4 from the proof of Theorem 8.9.

To now show that this evaluation is not possible in time $2^{o(\sqrt{n})}$ unless ETH fails, notice that if G has $|V|$ vertices, then $\hat{T}$ has $n = O(|V|)$ many different entries. So any evaluation in time $2^{o(\sqrt{n})}$ would decide whether G admits a proper 3-coloring in time $2^{o\left(\sqrt{|V|}\right)}$, which is a contradiction to Lemma 8.8[8] unless ETH fails. ◀

▶ Remark 8.10. All these hardness results also hold if the highest weight vectors are given as a Young tableau T with content $(nd) \times 1$ opposed to $\hat{T}$ with content $n \times d$ by replacing the entries containing 1 in $\hat{T}$ by $1, \ldots, d$ and 2 by $d+1, \ldots, 2d$ and so on in a left-to-right, top-to-bottom fashion. This corresponds to undoing the projection of $\otimes^n \mathrm{Sym}^d V$ onto $\mathrm{Sym}^n \mathrm{Sym}^d V$. In the cases when $\hat{T}$ is semistandard T is standard.

References

1 Abdelmalek Abdesselam. Feynman diagrams in algebraic combinatorics. *Séminaire Lotharingien de Combinatoire [electronic only]*, 49:B49c, 45 p., electronic only–B49c, 45 p., electronic only, 2002. URL: http://eudml.org/doc/123420.

2 Abdelmalek Abdesselam, Christian Ikenmeyer, and Gordon Royle. 16,051 formulas for Ottaviani's invariant of cubic threefolds. *Journal of Algebra*, 447:649–663, 2016.

3 Daniel J. Bates and Luke Oeding. Toward a Salmon conjecture. *Experimental Mathematics*, 20(3):358–370, 2011. doi:10.1080/10586458.2011.576539.

4 Christine Bessenrodt and Christiane Behns. On the Durfee size of Kronecker products of characters of the symmetric group and its double covers. *Journal of Algebra*, 280(1):132–144, 2004.

5 Christine Bessenrodt, Chris Bowman, and Rowena Paget. The classification of multiplicity-free plethysms of Schur functions, 2020. arXiv:2001.08763.

6 D. Bini. Relations between exact and approximate bilinear algorithms. applications. *CALCOLO*, 17(1):87–97, January 1980. doi:10.1007/BF02575865.

[8] or Lemma 8.5 for a weaker lower bound of $2^{o\left(\sqrt[4]{n}\right)}$

7 Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inf. Process. Lett.*, 8(5):234–235, 1979. doi:10.1016/0020-0190(79)90113-3.

8 Markus Bläser and Christian Ikenmeyer. Introduction to geometric complexity theory, 2018. lecture notes, summer 2017 at Saarland University, http://people.mpi-inf.mpg.de/~cikenmey/teaching/summer17/introtogct/gct.pdf, version from July 25, 2018.

9 Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5), 2018. doi:10.1145/3209663.

10 Russ Bubley, Martin E. Dyer, Catherine S. Greenhill, and Mark Jerrum. On approximately counting colorings of small degree graphs. *SIAM J. Comput.*, 29(2):387–400, 1999. doi:10.1137/S0097539798338175.

11 Peter Bürgisser. The complexity of factors of multivariate polynomials. In *42nd IEEE Symposium on Foundations of Computer Science (Las Vegas, NV, 2001)*, pages 378–385. IEEE Computer Soc., Los Alamitos, CA, 2001.

12 Peter Bürgisser, Matthias Christandl, and Christian Ikenmeyer. Even partitions in plethysms. *Journal of Algebra*, 328(1):322–329, 2011.

13 Peter Bürgisser and Christian Ikenmeyer. Geometric complexity theory and tensor rank. *Proceedings 43rd Annual ACM Symposium on Theory of Computing 2011*, pages 509–518, 2011.

14 Peter Bürgisser and Christian Ikenmeyer. Explicit lower bounds via geometric complexity theory. *Proceedings 45th Annual ACM Symposium on Theory of Computing 2013*, pages 141–150, 2013.

15 Peter Bürgisser and Christian Ikenmeyer. Fundamental invariants of orbit closures. *Journal of Algebra*, 477(Supplement C):390–434, 2017.

16 Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No occurrence obstructions in geometric complexity theory. *Journal of the American Mathematical Society*, 32:163–193, 2019. A conference version appeared in: Proceedings IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS 2016), 386–395.

17 Peter Bürgisser, J.M. Landsberg, Laurent Manivel, and Jerzy Weyman. An overview of mathematical issues arising in the Geometric complexity theory approach to VP v.s. VNP. *SIAM J. Comput.*, 40(4):1179–1209, 2011.

18 Liming Cai and David Juedes. Subexponential parameterized algorithms collapse the w-hierarchy. In *International Colloquium on Automata, Languages, and Programming*, pages 273–284. Springer, 2001.

19 Enrico Carlini, Maria Virginia Catalisano, and Anthony V Geramita. The solution to the Waring problem for monomials and the sum of coprime monomials. *Journal of algebra*, 370:5–14, 2012.

20 Man-Wai Cheung, Christian Ikenmeyer, and Sevak Mkrtchyan. Symmetrizing tableaux and the 5th case of the Foulkes conjecture. *Journal of Symbolic Computation*, 80:833–843, 2017.

21 Luca Chiantini, Jonathan D. Hauenstein, Christian Ikenmeyer, Joseph M. Landsberg, and Giorgio Ottaviani. Polynomials and the exponent of matrix multiplication. *Bulletin of the London Mathematical Society*, 50(3):369–389, 2018. doi:10.1112/blms.12147.

22 Matthias Christandl, Brent Doran, and Michael Walter. Computing multiplicities of lie group representations. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, page 639–648, USA, 2012. IEEE Computer Society. doi:10.1109/FOCS.2012.43.

23 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 4(8). Springer, 2015.

24 Noah Daleo, Jonathan Hauenstein, and Luke Oeding. Computations and equations for Segre-Grassmann hypersurfaces. *Portugaliae Mathematica*, 73, August 2014. doi:10.4171/PM/1977.

25 Julian Dörfler, Christian Ikenmeyer, and Greta Panova. On geometric complexity theory: Multiplicity obstructions are stronger than occurrence obstructions. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece.*, pages 51:1–51:14, 2019. journal version accepted for publication in SIAM J Appl Alg Geom (SIAGA). doi:10.4230/LIPIcs.ICALP.2019.51.

26 Cameron Farnsworth. Koszul–young flattenings and symmetric border rank of the determinant. *Journal of Algebra*, 447:664–676, 2016. doi:10.1016/j.jalgebra.2015.11.011.

27 Nick Fischer and Christian Ikenmeyer. The computational complexity of plethysm coefficients. *computational complexity*, 29(2):8, November 2020. doi:10.1007/s00037-020-00198-4.

28 Michael Forbes. Some concrete questions on the border complexity of polynomials. Talk presented at the Workshop on Algebraic Complexity Theory, WACT 2016, Tel Aviv, 2016. video available at https://www.cs.tau.ac.il/~shpilka/wact2016/videos/index.php accessed 10/17/2019. URL: https://www.cs.tau.ac.il/~shpilka/wact2016/videos/index.php.

29 Michael Andrew Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis, MIT, 2014. URL: https://dspace.mit.edu/handle/1721.1/89843.

30 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified np-complete graph problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

31 Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. Boundaries of VP and VNP. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ICALP.2016.34.

32 Charles Hermite. Sur la theorie des fonctions homogenes à deux indéterminées. *Cambridge and Dublin Mathematical Journal*, 9:172–217, 1854.

33 Anthony Iarrobino and Vassil Kanev. *Power sums, Gorenstein algebras, and determinantal loci.* Springer Science & Business Media, 1999.

34 Christian Ikenmeyer. *Geometric Complexity Theory, Tensor Rank, and Littlewood-Richardson Coefficients.* PhD thesis, Institute of Mathematics, University of Paderborn, 2012. URL: http://nbn-resolving.de/urn:nbn:de:hbz:466:2-10472.

35 Christian Ikenmeyer. The Saxl conjecture and the dominance order. *Discrete Mathematics*, 338(11):1970–1975, 2015.

36 Christian Ikenmeyer and Umangathan Kandasamy. Implementing geometric complexity theory: On the separation of orbit closures via symmetries. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 713–726, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384257.

37 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

38 Mrinal Kumar. On top fan-in vs formal degree for depth-3 arithmetic circuits, 2018. URL: https://eccc.weizmann.ac.il/report/2018/068/revision/1/download.

39 Shrawan Kumar. A study of the representations supported by the orbit closure of the determinant. *Compositio Mathematica*, 151, September 2011. doi:10.1112/S0010437X14007660.

40 Shrawan Kumar and J.M. Landsberg. Connections between conjectures of Alon-Tarsi, Hadamard-Howe, and integrals over the special unitary group. *Discrete Math.*, 338(7):1232–1238, 2015. doi:10.1016/j.disc.2015.01.027.

41 J. M. Landsberg. Geometric complexity theory: an introduction for geometers. *Annali dell'Università di Ferrara*, 61(1):65–117, 2015. doi:10.1007/s11565-014-0202-7.

42 Joseph Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, 2011.

43 M.A.A. Leeuwen, van, A.M. Cohen, and B. Lisser. *Lie : a package for Lie group computations*. Centrum voor Wiskunde en Informatica, 1992.

44 Laurent Manivel and Mateusz Michałek. Effective constructions in plethysms and Weintraub's conjecture. *Algebras and Representation Theory*, 17(2):433–443, April 2014. `doi:10.1007/s10468-012-9402-y`.

45 K.D. Mulmuley and M. Sohoni. Geometric Complexity Theory. I. An approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526 (electronic), 2001.

46 K.D. Mulmuley and M. Sohoni. Geometric Complexity Theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008.

47 Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd ACM Symposium on Theory of Computing, ACM Press*. Citeseer, 1991.

48 Luke Oeding and Claudiu Raicu. Tangential varieties of Segre-Veronese varieties. *Collectanea Mathematica*, 65, November 2011. `doi:10.1007/s13348-014-0111-1`.

49 Giorgio Ottaviani. Five lectures on projective invariants, lecture notes for trento school, september 2012, 2013. `arXiv:1305.2749`, to appear in Rendiconti del Seminario Matematico, Torino.

50 Claudiu Raicu. 3×3 minors of catalecticants. *Mathematical Research Letters*, 20, July 2013. `doi:10.4310/MRL.2013.v20.n4.a10`.

51 Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Comb. Theory, Ser. B*, 62(2):323–348, 1994. `doi:10.1006/jctb.1994.1073`.

52 Steven Sam and Andrew Snowden. Proof of stembridge's conjecture on stability of Kronecker coefficients. *Journal of Algebraic Combinatorics*, 43:1–10, 2016.

53 Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *Automata, Languages and Programming*, pages 60–71, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

54 Yaroslav Shitov. How hard is the tensor rank?, 2016. `arXiv:1611.01559`.

55 Roberto Tamassia and Ioannis G Tollis. Planar grid embedding in linear time. *IEEE Transactions on circuits and systems*, 36(9):1230–1234, 1989.

On Query-To-Communication Lifting for Adversary Bounds

Anurag Anshu
EECS & Challenge Institute for Quantum Computation,
University of California, Berkeley, CA, USA
Simons Institute for the Theory of Computing, Berkeley, CA, USA

Shalev Ben-David
University of Waterloo, Canada

Srijita Kundu
Centre for Quantum Technologies, National University of Singapore, Singapore

Abstract

We investigate query-to-communication lifting theorems for models related to the quantum adversary bounds. Our results are as follows:

1. We show that the classical adversary bound lifts to a lower bound on randomized communication complexity with a constant-sized gadget. We also show that the classical adversary bound is a strictly stronger lower bound technique than the previously-lifted measure known as critical block sensitivity, making our lifting theorem one of the strongest lifting theorems for randomized communication complexity using a constant-sized gadget.
2. Turning to quantum models, we show a connection between lifting theorems for quantum adversary bounds and secure 2-party quantum computation in a certain "honest-but-curious" model. Under the assumption that such secure 2-party computation is impossible, we show that a simplified version of the positive-weight adversary bound lifts to a quantum communication lower bound using a constant-sized gadget. We also give an unconditional lifting theorem which lower bounds bounded-round quantum communication protocols.
3. Finally, we give some new results in query complexity. We show that the classical adversary and the positive-weight quantum adversary are quadratically related. We also show that the positive-weight quantum adversary is never larger than the square of the approximate degree. Both relations hold even for partial functions.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Quantum complexity theory

Keywords and phrases Quantum computing, query complexity, communication complexity, lifting theorems, adversary method

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.30

Related Version *Full Version*: `https://arxiv.org/abs/2012.03415`

Funding *Anurag Anshu*: Supported by the NSF QLCI program through grant number OMA-2016245. Part of this work was done when A.A. was affiliated to the Department of Combinatorics & Optimization and Institute for Quantum Computing, University of Waterloo, as well as to the Perimeter Institute for Theoretical Physics.
Shalev Ben-David: Supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), DGECR-2019-00027 and RGPIN-2019-04804[1].
Srijita Kundu: Supported by the National Research Foundation, including under NRF RF Award No. NRF-NRFF2013-13, the Prime Minister's Office, Singapore; the Ministry of Education, Singapore, under the Research Centres of Excellence program and by Grant No. MOE2012-T3-1-009; and in part by the NRF2017-NRF-ANR004 VanQuTe Grant. Part of this work was done when S.K. was visiting the Institute of Quantum Computing, University of Waterloo.

[1] Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), DGECR-2019-00027 et RGPIN-2019-04804.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 30; pp. 30:1–30:39
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements We thank Rahul Jain and Dave Touchette for helpful discussions related to the $\mathrm{QICZ}(G) > 0$ conjecture. We thank Robin Kothari for helpful discussions related to the adversary bounds. We thank Anne Broadbent for helpful discussions related to quantum secure 2-party computation. We thank Mika Göös for helpful discussions regarding critical block sensitivity and its lifting theorem. We thank Jevgēnijs Vihrovs and the other authors of [6] for helpful discussions regarding the classical adversary method, and particularly Krišjānis Prūsis for the proof of Lemma 27.

1 Introduction

Communication complexity is an important model of computation with deep connections to many parts of theoretical computer science [29]. In communication complexity, two parties, called Alice and Bob, receive inputs x and y from sets $\mathcal{X}$ and $\mathcal{Y}$ respectively, and wish to compute some joint function $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ on their inputs. Alice and Bob cooperate together, and their goal is to minimize the number of bits they must exchange before determining $F(x,y)$.

Recently, a lot of attention has been devoted to connections between communication complexity and query complexity. In particular, query-to-communication "lifting" theorems are powerful tools which convert lower bounds in query complexity into lower bounds in communication complexity in a black-box manner. Since query lower bounds are typically much easier to prove than communication lower bounds, these tools are highly useful for the study of communication complexity, and often come together with new communication complexity results (such as separations between different communication complexity models). For example, see [21, 22, 23, 25, 16].

Lifting theorems are known for many models of computation, including deterministic [23] and randomized [25] algorithms. Notably, however, a lifting theorem for quantum query complexity is not known; the closest thing available is a lifting theorem for approximate degree (also known as the polynomial method), which lifts to approximate logrank [37]. This allows quantum query lower bounds proved via the polynomial method to be turned into quantum communication lower bounds, but a similar statement is not known even for the positive-weight quantum adversary method [5, 39].

In this work, we investigate lifting theorems for the adversary method and related models. We prove a lifting theorem for a measure called the classical adversary bound. For the quantum adversary method, we show that there is a surprising connection with the cryptographic notion of secure 2-party computation. Specifically, we show that a lifting theorem for a simplified version of the positive-weight adversary method follows from a plausible conjecture regarding the impossibility of secure 2-party computation in a certain "honest but curious" quantum model. We also prove an unconditional lifting theorem which lower bounds bounded-round quantum algorithms.

Finally, we prove some query complexity results that may be of independent interest: first, a quadratic relationship between the positive-weight adversary bound and the classical adversary bound; and second, we show that the positive-weight adversary bound can never be larger than the square of the approximate degree. This means that the (positive) adversary method can never beat the polynomial method by more than a quadratic factor. These results hold even for partial functions.

1.1 Lifting theorems

The statement of a lifting theorem typically has the following form:

$$\mathrm{M}^{cc}(f \circ G) = \Omega(\mathrm{M}(f)).$$

Here $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is a (fixed) communication complexity function, called a "gadget", which typically has low communication cost; $f\colon \{0,1\}^n \to \{0,1\}$ is an arbitrary Boolean function; $\mathrm{M}(\cdot)$ is a measure in query complexity, representing the cost of computing the function f in query complexity; and $\mathrm{M}^{cc}(\cdot)$ is a measure in communication complexity. The notation $f \circ G$ denotes the block-composition of f with G. This is a communication complexity function defined as follows: Alice gets input $(x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$, Bob gets input $(y_1, y_2, \ldots, y_n) \in \mathcal{Y}^n$, and they must output $f(G(x_1, y_1), G(x_2, y_2), \ldots, G(x_n, y_n))$. Hence $f \circ G$ is a function with signature $\mathcal{X}^n \times \mathcal{Y}^n \to \{0,1\}$.

There are two primary types of lifting theorems: those that work with a constant-sized gadget G (independent of f), and those that work with a gadget G whose size logarithmic in the input size n of f. [2] The latter type tend to be much more prevalent; recent lifting theorems for deterministic and randomized communication complexities all use log-sized or larger gadgets [23, 42, 17, 25, 16]. We remark, however, that even with log-sized or larger gadgets, lifting theorems are highly nontrivial to prove: a lifting theorem for BPP, which lifts randomized query lower bounds to randomized communication lower bounds, was only established in the last few years, while an analogous result for BQP remains an open problem.

Lifting theorems which work with a constant-sized gadget are even harder to prove, but often turn out to be much more useful. The reason is that common function families, like disjointness (which we denote $\textsc{Disj}_n$) [3] or inner product (which we denote IP_n) [4], are *universal*. This means for every communication function $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, its communication matrix (that is, its truth table) is a submatrix of the communication matrix of a sufficiently large instance of the $\textsc{Disj}$ function. In other words, every communication function is a sub-function of $\textsc{Disj}_k$ and IP_k for sufficiently large k. If the size of G is constant, then it is necessarily contained in a $\textsc{Disj}$ function of *constant* size (and similarly for IP). Hence lifting with any constant-sized gadget G is enough to guarantee a lifting theorem with a constant-sized disjointness gadget and constant-sized inner product gadget (and similarly for every other universal function family). In short, lifting with a constant-sized gadget implies lifting with almost any gadget of your choice.

In particular, a lifting theorem with a constant-sized gadget immediately implies a lower bound for $\textsc{Disj}_n$ and IP_n themselves. To see this, suppose we had a lifting theorem

$$\mathrm{M}^{cc}(f \circ G) = \Omega(\mathrm{M}(f))$$

for all Boolean functions f and a fixed (constant-sized) communication gadget G. Then G is a sub-function of $\textsc{Disj}_k$ and of IP_k for some constant k. Note that $\textsc{Disj}_n = \mathrm{OR}_{n/k} \circ \textsc{Disj}_k$ and that $\mathrm{IP}_n = \textsc{Parity}_{n/k} \circ \mathrm{IP}_k$. Hence we get $\mathrm{M}^{cc}(\textsc{Disj}_n) = \Omega(\mathrm{M}(\mathrm{OR}_{n/k}))$ and $\mathrm{M}^{cc}(\mathrm{IP}_n) = \Omega(\mathrm{M}(\textsc{Parity}_{n/k}))$. Since k is constant, this can potentially give lower bounds on $\mathrm{M}^{cc}(\textsc{Disj}_n)$ and on $\mathrm{M}^{cc}(\mathrm{IP}_n)$ that are tight up to constant factors, depending on the measures $\mathrm{M}^{cc}(\cdot)$ and $\mathrm{M}(\cdot)$.

[2] Sometimes, lifting theorems use a gadget G which is large – polynomial in n – but which can still be computed using $O(\log n)$ communication.

[3] In the disjointness function, Alice and Bob receive n-bit strings x and y and must output 1 if and only if there exists an index $i \in [n]$ such that $x_i = y_i = 1$.

[4] In the inner product function, Alice and Bob receive n-bit strings x and y, and must compute inner product of those strings over $\mathbb{F}_2$.

There have only been a handful of lifting theorems which work with constant-sized gadgets. One such result follows from Sherstov's work for approximate degree and related measures [37]. The part of that work which is most relevant to us is the lifting of approximate degree to lower bounds on approximate logrank, and hence on the quantum communication complexity of the lifted function. Sherstov's work means that if one can prove a quantum lower bound for a query function f using the *polynomial method* [9], then this lower bound will also apply to the quantum communication complexity of $f \circ G$, where G is a constant sized gadget. Such a lifting theorem is not known to hold for the adversary methods [5, 39], however (not even with a log-sized gadget).

Another lifting theorem with a constant-sized gadget appears in [27, 24]. There, a query measure called *critical block sensitivity* [27] is lifted to a lower bound on randomized communication complexity.

1.2 Adversary methods

The quantum adversary bounds are extremely useful methods for lower bounding quantum query complexity. The original adversary method was introduced by Ambainis [5]. It was later generalized in several ways, which were shown to all be equivalent [39], and are known as the positive-weight adversary bound, denoted $\mathrm{Adv}(f)$. This bound has many convenient properties: it has many equivalent formulations (among them a semidefinite program), it is reasonably easy to use in practice, and it behaves nicely under many operations, such as composition. The positive-weight adversary bound is one of the most commonly used techniques for lower bounding quantum query complexity.

A related measure is called the negative-weight adversary bound, introduced in [26], which we denote by $\mathrm{Adv}^{\pm}(f)$. This is a strengthening of the positive adversary bound, and satisfies $\mathrm{Adv}^{\pm}(f) \geq \mathrm{Adv}(f)$ for all (possibly partial) Boolean functions f. Surprisingly, in [35, 32], it was shown that the negative-weight adversary is actually *equal* to quantum query complexity up to constant factors.

The quantum adversary methods have no known communication complexity analogues. However, that by itself does not rule out a lifting theorem: one might still hope to lift $\mathrm{Adv}(f)$ or $\mathrm{Adv}^{\pm}(f)$ to lower bounds on quantum communication complexity, similar to how critical block sensitivity $\mathrm{cbs}(f)$ was lifted to a lower bound on randomized communication complexity [27, 24]. Unfortunately, no such lifting theorems are currently known, not even for the positive-weight adversary method, and not even with a large gadget size.

Interestingly, it is possible to define a lower bound technique for *randomized* algorithms which is motivated by the (positive) quantum adversary method. This measure was first introduced in [1, 30], and different variants of it have been subsequently studied [6]. Here, we use the largest of these variants, which we denote by $\mathrm{CAdv}(f)$ (in [6], it was denoted by $\mathrm{CMM}(f)$). In [6], it was shown that for total functions f, $\mathrm{CAdv}(f)$ is (up to constant factors) equal to a measure called fractional block sensitivity, which we denote $\mathrm{fbs}(f)$. However, for partial functions, there can be a large separation between the two measures. For more on fractional block sensitivity, see [2, 28].

1.3 Our contributions

Lifting the classical adversary

Our first contribution is a lifting theorem for the classical adversary bound $\mathrm{CAdv}(f)$. We lift it to a lower bound on randomized communication complexity using a constant-sized gadget.

▶ **Theorem 1.** *There is an explicitly given function $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ such that for any (possibly partial) Boolean function or relation f,*

$$\mathrm{RCC}(f \circ G) = \Omega(\mathrm{CAdv}(f)).$$

Here $\mathrm{RCC}(f \circ G)$ denotes the randomized communication complexity of $f \circ G$ with shared randomness. We note that [27, 24] provided a lifting theorem that has a similar form, only with the measure $\mathrm{cbs}(f)$ in place of $\mathrm{CAdv}(f)$. To compare the two theorems, we should compare the two query measures. We have the following theorem.

▶ **Lemma 2.** *For all (possibly partial) Boolean functions or relations f, $\mathrm{CAdv}(f) = \Omega(\mathrm{cbs}(f))$. Moreover, there is a family of total functions f for which $\mathrm{CAdv}(f) = \Omega(\mathrm{cbs}(f)^{3/2})$.*

Lemma 2 says that $\mathrm{CAdv}(f)$ is a strictly stronger lower bound technique than $\mathrm{cbs}(f)$, and hence Theorem 1 is stronger than the lifting theorem of [24]. A proof of Lemma 2 follows from our proof that $\mathrm{CAdv}(f) \geq \mathrm{cfbs}(f)/2$ in Lemma 26, together with the known power 3/2 separation between $\mathrm{fbs}(f)$ and $\mathrm{bs}(f)$ for total functions [20].[5] This makes Theorem 1 one of the strongest known lifting theorems for randomized communication complexity which works with a constant-sized gadget.[6]

We note that the lifting theorem of [24] for the measure $\mathrm{cbs}(f)$ also works when f is a *relation*, which is a more general setting than partial functions; indeed, most of their applications for the lifting theorem were for relations f rather than functions. We extend Theorem 1 to relations as well, and also show that $\mathrm{CAdv}(f) = \Omega(\mathrm{cbs}(f))$ for all relations. In fact, it turns out that for partial functions, $\mathrm{CAdv}(f)$ is equal to a fractional version of $\mathrm{cbs}(f)$, which we denote $\mathrm{cfbs}(f)$; however, for relations, $\mathrm{CAdv}(f)$ is a stronger lower bound technique than $\mathrm{cfbs}(f)$ (which in turn is stronger than $\mathrm{cbs}(f)$). We also note that our techniques for lifting $\mathrm{CAdv}(f)$ are substantially different from those of [27, 24].

Lifting quantum measures

Our first quantum result says that $\mathrm{CAdv}(f)$ lifts to a lower bound on bounded-round quantum communication protocols. This may seem surprising, as $\mathrm{CAdv}(f)$ does not lower bound quantum algorithms in query complexity; however, one can show that $\mathrm{CAdv}(f)$ does lower bound *non-adaptive* quantum query complexity, or even quantum query algorithms with limited adaptivity. This motivates the following result.

▶ **Theorem 3.** *There is an explicitly given function $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ such that for any (possibly partial) Boolean function or relation f,*

$$\mathrm{QCC}^r(f \circ G) = \Omega(\mathrm{CAdv}(f)/r^2).$$

Here $\mathrm{QCC}^r(\cdot)$ denotes the quantum communication complexity for an r-round quantum protocol with shared entanglement.

We note that since any r-round protocol has communication cost at least r, we actually get a lower bound of $\mathrm{CAdv}(f)/r^2 + r$. Minimizing over r yields a lower bound of $\mathrm{CAdv}(f)^{1/3}$ even on unbounded-round protocols. This may not seem very useful, since $\mathrm{CAdv}(f)^{1/3}$ is

[5] For total functions, we have $\mathrm{fbs}(f) = \mathrm{cfbs}(f)$ and $\mathrm{cbs}(f) = \mathrm{bs}(f)$.

[6] Sherstov's lifting theorem for approximate degree [37] also works with a constant-sized gadget, and is incomparable to our result as a lower bound technique for randomized communication complexity.

smaller than $\widetilde{\deg}(f)$, a measure we know how to lift [37]. However, we can generalize this result to relations. For relations, we do not know how to compare $\mathrm{CAdv}(f)^{1/3}$ to $\widetilde{\deg}(f)$, and therefore our lifting theorem gives something new, even in the unbounded-round setting.

▶ **Corollary 4.** *There is an explicitly given function $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ such that for any (possibly partial) Boolean function or relation f,*

$$\mathrm{QCC}(f \circ G) = \Omega(\mathrm{CAdv}(f)^{1/3}),$$

where QCC *denotes the quantum communication complexity with shared entanglement.*

We next turn our attention to lower bounding unbounded-round quantum communication protocols by lifting a quantum adversary method. Instead of aiming for the positive-weight adversary bound, we work with a simplified version, studied in [3], which we denote $\mathrm{Adv}_1(f)$. This measure is a restriction of Adv to a pairs of inputs with a single bit of difference.

We have $\mathrm{Adv}_1(f) \leq \mathrm{Adv}(f)$, and [3] showed that $\mathrm{Adv}_1(f) = O(\widetilde{\deg}(f))$. However, their proof of the latter is tricky, and we do not use it here; we give a direct lifting of $\mathrm{Adv}_1(f)$ (under a certain assumption), and we argue that the techniques we use are likely to generalize to lifting $\mathrm{Adv}(f)$ in the future.

We prove the following theorem, which lifts $\mathrm{Adv}_1(f)$ but has a dependence on a new complexity measure $\mathrm{QICZ}(G)$ that we introduce.

▶ **Theorem 5.** *For any (possibly partial) Boolean function or relation f and any communication function G which contains both AND_2 and OR_2 as subfunctions, we have*

$$\mathrm{QCC}(f \circ G) = \Omega(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)).$$

At first glance, this theorem might look very strong: not only does it lift the simplified adversary bound for a single gadget G, it even does so for *all* G and gives an explicit dependence on G. Unfortunately, there is a catch: the measure $\mathrm{QICZ}(G)$ may be 0 for some communication functions G. In fact, we cannot rule out the possibility that $\mathrm{QICZ}(G) = 0$ for *all* communication functions G, in which case Theorem 5 does not say anything. On the other hand, note that if $\mathrm{QICZ}(G) > 0$ for even a single function G, then Theorem 5 gives a lifting theorem for $\mathrm{Adv}_1(f)$ with a constant-sized gadget, which works even for relations.

We give an interpretation of the measure $\mathrm{QICZ}(G)$ in terms of a cryptographic primitive called secure 2-party computation. In such a primitive, Alice and Bob want to compute a function G on their inputs x and y, but they do not want to reveal their inputs to the other party. Indeed, Alice wants to hide everything about x from Bob and Bob wants to hide everything about y from Alice, with the exception of the final function value $G(x, y)$ (which they are both expected to know at the end of the protocol). We also seek information-theoretic security: there are no limits on the computational power of Alice and Bob. Since we are interested in a quantum version, we will allow Alice and Bob to exchange quantum communication rather than classical communication, potentially with shared entanglement.

Secure 2-party computation is known to be impossible in general, even quantumly [33, 18, 14, 19, 36]. However, in our case, we care about an "honest but curious" version of the primitive, in which Alice and Bob trust each other to execute the protocol faithfully, but they still do not trust each other not to try to learn the others' input. In the quantum setting, it is a bit difficult to define such an honest-but-curious model: after all, if Alice and Bob are honest, they might be forbidden by the protocol from ever executing intermediate measurements, and the protocol might even tell them to "uncompute" everything except for the final answer, to ensure all other information gets deleted. Hence it would seem that honest parties can trivially do secure 2-party computation.

The way we will define quantum secure two-party computation in the honest-but-curious setting will be analogous to the information-based classical definition (see, for example, [13]). Classically, the information leak that Alice and Bob must suffer in an honest execution of the best possible protocol is captured by $\mathrm{IC}(G)$, the information cost of the function G. The measure $\mathrm{IC}(G)$ is the amount Alice learns about Bob's input plus the amount Bob learns about Alice's input, given the best possible protocol and the worst possible distribution over the inputs; we note that this measure includes the value of $G(x,y)$ as part of what Alice and Bob learn about each others' inputs, whereas secure two-party computation does not count learning $G(x,y)$ as part of the cost, but this is only a difference of at most 2 bits of information (one on Alice's side and one on Bob's side); hence, up to an additive factor of 2, $\mathrm{IC}(G)$ captures the information leak necessary in a two-party protocol computing G.

For a quantum version of this, we will use QIC, a measure which is a quantum analogue of IC and which was introduced in [41]. However, we note that if Alice and Bob send the same bit $G(x,y)$ back and forth n times, this will add $\Theta(n)$ to the value of QIC for that protocol, due to subtleties in the definition of QIC (this does not occur classically with IC). Hence, in the quantum setting, QIC does not capture the two-party information leak as cleanly as IC did classically.

Instead, we modify the definition of QIC to a measure we denote $\mathrm{QICZ}(G)$. For this measure, Alice and Bob want a protocol Π such that for any distribution μ that has support only on 0-inputs or only on 1-inputs, $\mathrm{QIC}(\Pi,\mu)$ is small. In other words, if we use $\mathrm{QIC}\,0(\Pi)$ to denote the quantum information cost of Π against 0-distributions and $\mathrm{QIC}\,1(\Pi)$ to denote the quantum information cost of Π against 1-distributions, then we define $\mathrm{QICZ}(\Pi) = \max\{\mathrm{QIC}\,0(\Pi), \mathrm{QIC}\,1(\Pi)\}$.

When $\mathrm{QICZ}(\Pi)$ is near zero, it means that Alice and Bob learn nothing about each others' inputs when conditioned on the output of the function. The two-party secure computation question then becomes: does such a secure protocol Π exists for computing any fixed communication function G?

Intuitively, we believe that the answer should be no, at least for some communication functions G. This would align with the known impossibility of various types of secure 2-party quantum computation, though none of those impossibility results seem to apply to our setting. Interestingly, we have the following lemma, which follows directly form the way we define $\mathrm{QICZ}(G)$.

▶ **Lemma 6.** *Suppose that our version of secure 2-party quantum computation is impossible for a communication function G which contains both AND and OR as sub-functions. Then $\mathrm{QICZ}(G) > 0$, and hence $\mathrm{Adv}_1(\cdot)$ lifts to a quantum communication lower bound with the gadget G.*

We hope that future work can extend this lemma to a lifting theorem for the positive-weight quantum adversary $\mathrm{Adv}(\cdot)$; if so, the problem of lifting the positive quantum adversary bound will reduce to the problem of ruling out secure 2-party quantum computation in the model we outlined above.

New query relations

Finally, our study of the classical adversary bound led to some new relations in query complexity that are likely to be of independent interest.

▶ **Theorem 7.** *For all (possibly partial) Boolean functions f,*

$$\mathrm{Adv}(f) = O(\widetilde{\deg}(f)^2).$$

Here $\widetilde{\deg}(f)$ is the approximate degree of f to bounded error.[7] This relationship is interesting, as it says that the positive-weight adversary method can never beat the polynomial method by more than a quadratic factor. Conceivably, this can even be used as a lower bound technique for the approximate degree of Boolean functions (which is a measure that is often of interest even apart from quantum lower bounds). In fact, we prove a strengthening of Theorem 7.

▶ **Theorem 8.** *For all (possibly partial) Boolean functions f,*

$$\widetilde{\deg}_\epsilon(f) \geq \frac{\sqrt{(1-2\epsilon)\,\mathrm{CAdv}(f)}}{\pi}.$$

This version of the theorem is stronger, since $\mathrm{Adv}(f) \leq \mathrm{CAdv}(f)$. Finally, we prove a quadratic relationship between the classical and quantum (positive-weight) adversary bounds.

▶ **Theorem 9.** *For all (possibly partial) Boolean functions f,*

$$\mathrm{Adv}(f) \leq \mathrm{CAdv}(f) \leq 2\,\mathrm{Adv}(f)^2.$$

We note that all of these new relations hold even for partial functions. This is unusual in query complexity, where most relations hold only for total functions, and where most pairs of measures can be exponentially separated in the partial function setting.

1.4 Our Techniques

We introduce several new techniques that we believe will be useful in future work on adversary methods in communication complexity.

A lifting framework for adversary methods

One clear insight we contribute in this work is that lifting theorems for adversary method can be fruitfully attacked in a "primal" way, and using information cost. To clarify, our approach is to take a protocol Π for the lifted function $f \circ G$, and to convert it into a solution to the primal (i.e. minimization) program for the target adversary bound of f.

The primal program for an adversary method generally demands a non-negative weight $q(z,i)$ for each input string $z \in \{0,1\}^n$ and each index $i \in [n]$, such that a certain feasibility constraint is satisfied for each pair (z,w) with $f(z) \neq f(w)$, and such that $\sum_{i\in[n]} q(z,i)$ is small for each input z. Our approach is to use an information cost measure to define $q(z,i)$, where the information is measured against a distribution μ_z over n-tuples of inputs to G that evaluate to z, and where we only measure the information transmitted by the protocol about the i-th input to G, conditioned on the previous bits.

We show that this way of getting a solution to the (minimization version of) the adversary bound for f using a communication protocol for $f \circ G$ suffices for lifting CAdv to a randomized communication lower bound (with a constant-sized gadget), and that it also suffices for getting some quantum lifting theorems. Our information cost approach is similar to the approach taken in [7] to lower bound the information complexity of the AND function against the uniform distribution over 0-inputs.

[7] This is the minimum degree of an n-variate real polynomial p such that $|p(x)| \in [0,1]$ for all $x \in \{0,1\}^n$ and such that $|p(x) - f(x)| \leq 1/3$ for all x in the domain of f.

Product-to-sum reduction

One of the main tools we use in the proof of the lifting theorem for Adv_1 is what we call a product-to-sum reduction for quantum information cost. We show that if there is a protocol Π which computes some communication function F such that the geometric mean $\sqrt{\mathrm{QIC}(\Pi, \mu_0) \cdot \mathrm{QIC}(\Pi, \mu_1)}$ is small (where μ_0 and μ_1 are distributions over 0- and 1-inputs to F), then there is also a protocol Π' which also computes F and for which the arithmetic mean $\frac{1}{2}(\mathrm{QIC}(\Pi', \mu_0) + \mathrm{QIC}(\Pi', \mu_1))$ is small. In particular, a lower bound for the latter measure implies a lower bound for the former. This is useful because the sum (or maximum) of the two quantum information costs is a natural operation on quantum information measures to which lower bound tools may apply, while the product is not; yet the product of these information measures arises naturally in the study of adversary methods for a lifted query function.

To prove our product-to-sum reduction, we employ a chain of reductions. First, we show that if one of $\mathrm{QIC}(\Pi, \mu_0)$ or $\mathrm{QIC}(\Pi, \mu_1)$ is much smaller than the other, then we can use Π to get a low-information protocol for $\mathrm{OR} \circ F$, the composition of the OR function with F. Next, we use an argument motivated by [11]: we use Belov's algorithm for the combinatorial group testing problem [10] to use the low-information cost protocol for $\mathrm{OR} \circ F$ to get a low-information cost protocol for the task of computing n copies of F. Finally, we use an argument from [41] to get a low-information cost protocol for F itself.

Connection to secure two-party computation

Another insight important for this work is that lifting theorems for quantum adversary methods are related to quantum secure two-party computation, a cryptographic primitive. This connection comes through the measure $\mathrm{QICZ}(G)$: for communication gadgets G for which $\mathrm{QICZ}(G) > 0$, we know that secure two-party computation of G is impossible (in an "honest-but-curious" setting, where we require information-theoretic security); yet for such G, we can then lift $\mathrm{Adv}_1(f)$ to a lower bound on $\mathrm{QCC}(f \circ G)$. We believe this result is likely to extend to lifting theorems for other adversary methods in the future, though the dependence on $\mathrm{QICZ}(G) > 0$ may still remain.

We provide a minimax theorem for $\mathrm{QICZ}(G)$, giving an alternate characterization of the measure. This minimax theorem is used in our lifting theorem, and may also be useful for a future lower bound on $\mathrm{QICZ}(G)$ for some communication function G, which we view as an interesting open problem.

Insights into query complexity

Our results for query complexity follow from the following insights. First, we show that for partial functions, $\mathrm{CAdv}(f)$ is equivalent to the measure $\mathrm{cfbs}(f)$ (a fractional version of critical block sensitivity [27]) by converting the primal versions of the two programs to each other; this is not difficult to do, and the main contribution comes from (1) using the correct definition of $\mathrm{CAdv}(f)$ (out of the several definitions in [6], which are not equivalent to each other for partial functions), and (2) using the correct definition of $\mathrm{cfbs}(f)$ (which is a new definition introduced in this work). We attribute one direction of this conversion to Krišjānis Prūsis (personal communication).

Second, we show that the positive-weight adversary method $\mathrm{Adv}(f)$ is smaller than, but quadratically related to, $\mathrm{CAdv}(f)$. Once again, this result is not difficult, but relies on using the correct definition of $\mathrm{CAdv}(f)$ and on using the primal versions (i.e. minimization versions) of both programs. (Indeed, we use only the primal form of all the adversary methods throughout this paper; one of our insights is that this primal form is more convenient for proving structural properties of the adversary methods, including lifting theorems.)

Finally, we show that $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{cfbs}(f)})$, and hence $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{Adv}(f)})$, and this holds even for partial functions. We do this by essentially reducing it to the task of showing $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{fbs}(f)})$. The latter is already known [28]; however, it was only known for total functions, whereas we need it to hold for partial functions as well. The problem is that the previous proof relied on recursively composing f with itself, an operation which turns the fractional block sensitivity $\mathrm{fbs}(f)$ into the block sensitivity $\mathrm{bs}(f)$; unfortunately, this trick works only for total Boolean functions. Instead, we use a different trick for turning $\mathrm{fbs}(f)$ into $\mathrm{bs}(f)$: we compose f with the promise-OR function, and show that the block sensitivity of $f \circ \textsc{PrOR}$ is proportional to the fractional block sensitivity of f. We then convert an arbitrary polynomial approximating f into a polynomial approximating $f \circ \textsc{PrOR}$ by composing it with a Chebyshev-like polynomial computing $\textsc{PrOR}$; finally, we appeal to the known result that the square root of block sensitivity lower bounds approximate degree to finish the proof.

2 Preliminaries

2.1 Distance & information measures

We define all the distance and information measures for quantum states. The classical versions can be obtained by making the corresponding registers classical.

The ℓ_1 distance between two quantum states ρ and σ is defined as

$$\|\rho - \sigma\|_1 = \mathrm{Tr}\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}.$$

The entropy of a quantum state ρ_A on register A is defined as

$$H(A)_\rho = -\mathrm{Tr}(\rho\log\rho).$$

For a state ρ_{AB} on registers AB, the conditional entropy of A given B is

$$H(A|B)_\rho = H(AB)_\rho - H(B)_\rho.$$

Conditional entropy satisfies the following continuity bound [4]: if ρ and σ on registers AB satisfy $\|\rho-\sigma\|_1 \leq \epsilon$, then

$$|H(A|B)_\rho - H(A|B)_\sigma| \leq 4\epsilon\log|A| + 2h(\epsilon)$$

where $h(.)$ is the binary entropy function. For ρ_{ABC}, we define the mutual information and conditional mutual information as

$$I(A:B)_\rho = H(A)_\rho - H(A|B)_\rho \qquad I(A:B|C) = H(A|C)_\rho - H(A|BC)_\rho.$$

Mutual information satisfies

$$0 \leq I(A:B|C)_\rho \leq \min\{\log|A|, \log|B|\}$$

and the chain rule

$$I(A:BC)_\rho = I(A:B)_\rho + I(A:C|B)_\rho.$$

2.2 Query complexity

In query complexity, the primary object of study are Boolean functions, which are functions $f\colon \{0,1\}^n \to \{0,1\}$ where n is a positive integer. Often, we will actually study partial Boolean functions, which are defined on only a subset of $\{0,1\}^n$. We will use $\mathrm{Dom}(f)$ to denote the domain of f; this is a subset of $\{0,1\}^n$.

For a (possibly partial) Boolean function f, we use $\mathrm{D}(f)$, $\mathrm{R}(f)$, and $\mathrm{Q}(f)$ to denote its deterministic query complexity, randomized query complexity (to bounded error), and quantum query complexity (to bounded error), respectively. For the definition of these measures, see [15], though we won't use these definitions in this work.

2.2.1 Block sensitivity and its variants

We will use the following definitions.

Block notation. For a Boolean string $x \in \{0,1\}^n$ and a set $B \subseteq [n]$, we let x^B denote the string with the bits in B flipped; that is, $x_i^B = x_i$ for all $i \notin B$ and $x_i^B = 1 - x_i$ for all $i \in B$. The set B is called a *block*.

Sensitive block. For a (possibly partial) Boolean function f on n bits and an input $x \in \mathrm{Dom}(f)$, we say that a set $B \subseteq [n]$ is a *sensitive block* for x (with respect to f) if $x^B \in \mathrm{Dom}(f)$ and $f(x^B) \neq f(x)$.

Block sensitivity. The *block sensitivity* of a string $x \in \{0,1\}^n$ with respect to a (possibly partial) Boolean function f satisfying $x \in \mathrm{Dom}(f)$ is the maximum integer k such that there are k blocks $B_1, B_2, \dots, B_k \subseteq [n]$ which are all sensitive for x and which are all disjoint. This is denoted $\mathrm{bs}(x,f)$.

Block sensitivity of a function. The *block sensitivity* of a (possibly partial) Boolean function f is the maximum value of $\mathrm{bs}(x,f)$ over $x \in \mathrm{Dom}(f)$. This is denoted $\mathrm{bs}(f)$. Block sensitivity was originally introduced by Nisan [34], and is discussed in the survey by Buhrman and de Wolf [15].

Fractional block sensitivity. The *fractional block sensitivity* of a string $x \in \{0,1\}^n$ with respect to a (possibly partial) Boolean function f satisfying $x \in \mathrm{Dom}(f)$ is the maximum possible sum of weights $\sum_B w_B$, where the weights $w_B \geq 0$ are assigned to each sensitive block of x and must satisfy $\sum_{B:i\in B} w_B \leq 1$ for all $i \in [n]$. This is denoted by $\mathrm{fbs}(x,f)$. The fractional block sensitivity of a function f, denoted $\mathrm{fbs}(f)$, is the maximum value of $\mathrm{fbs}(x,f)$ over $x \in \mathrm{Dom}(f)$. Fractional block sensitivity was defined by [2], but see also [28].

Critical block sensitivity. For a (possibly partial) Boolean function f, we say that a total Boolean function f' is a *completion* of f if $f'(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$. The *critical block sensitivity* of f, denoted $\mathrm{cbs}(f)$, is defined as

$$\min_{f'} \max_{x \in \mathrm{Dom}(f)} \mathrm{bs}(x, f'),$$

where the minimum is taken over completions f' of f. This measure was defined by [27]. It equals $\mathrm{bs}(f)$ for total functions, but may be larger for partial functions.

Critical fractional block sensitivity. For a (possibly partial) Boolean function f, we define its *critical fractional block sensitivity*, denoted $\mathrm{cfbs}(f)$, as

$$\min_{f'} \max_{x\in\mathrm{Dom}(f)} \mathrm{cfbs}(x, f'),$$

where the minimum is taken over completions f' of f. This measure has not previously appeared in the literature.

2.2.2 Adversary bounds

Positive adversary bound. For a (possibly partial) Boolean function f, we define the positive-weight adversary bound, denoted $\mathrm{Adv}(f)$, as the minimum of the following program. We will have one non-negative weight $q(x,i)$ for each $x \in \mathrm{Dom}(f)$ and each $i \in [n]$. We call such a weight scheme feasible if, for all $x, y \in \mathrm{Dom}(f)$ with $f(x) \neq f(y)$, we have

$$\sum_{i:x_i\neq y_i} \sqrt{q(x,i)q(y,i)} \geq 1.$$

Then $\mathrm{Adv}(f)$ is defined as the minimum of $\max_{x\in\mathrm{Dom}(f)} \sum_{i\in[n]} q(x,i)$ over feasible weight schemes $q(\cdot,\cdot)$. A different version of the positive-weight adversary bound was defined in [5], though the version we've currently defined appears in [30] and [39] (in the latter, our definition is equivalent to $\mathrm{MM}(f)$).

Classical adversary bound. For a (possibly partial) Boolean function f, we define the classical adversary bound, denoted $\mathrm{CAdv}(f)$, as the minimum of the following program. We will have one non-negative weight $q(x,i)$ for each $x \in \mathrm{Dom}(f)$ and each $i \in [n]$, as before. We call such a weight scheme feasible if, for all $x, y \in \mathrm{Dom}(f)$ with $f(x) \neq f(y)$, we have

$$\sum_{i:x_i\neq y_i} \min\{q(x,i), q(y,i)\} \geq 1.$$

Then $\mathrm{CAdv}(f)$ is defined as the minimum of $\max_{x\in\mathrm{Dom}(f)} \sum_{i\in[n]} q(x,i)$ over feasible weight schemes $q(\cdot,\cdot)$. Observe that this definition is the same as that of $\mathrm{Adv}(f)$, except that the feasibility constraint sums up the minimum of $q(x,i)$ and $q(y,i)$ instead of the geometric mean. This feasibility constraint is harder to satisfy, and hence we have $\mathrm{CAdv}(f) \geq \mathrm{Adv}(f)$. A different version of the classical adversary was defined in [1], though the version we've currently defined appears in [30] and [6] (in the latter, our definition is equivalent to $\mathrm{CMM}(f)$).

Singleton adversary bound. [3] introduced a simplified version of the quantum adversary bound, which we denote $\mathrm{Adv}_1(f)$. As in the other adversaries, this will be the minimum over a program that has one non-negative weight $q(x,i)$ for each pair of input $x \in \mathrm{Dom}(f)$ and index $i \in [n]$. The objective value will once again be $\max_{x\in\mathrm{Dom}(f)} \sum_{i\in[n]} q(x,i)$. The only difference is the constraints: instead of placing a constraint for each $x, y \in \mathrm{Dom}(f)$ with $f(x) \neq f(y)$, we only place this constraint for such x, y that have Hamming distance exactly 1. Observe that this is a relaxation of the constraint in the definition of $\mathrm{Adv}(f)$, and hence $\mathrm{Adv}_1(f) \leq \mathrm{Adv}(f)$ for all (possibly partial) Boolean functions f.

2.3 A generalization to relations

So far, we've defined our query measures for partial Boolean functions. However, in many cases we will be interested in studying *relations*, which are a generalization of partial Boolean functions.

In query complexity, a *relation* is a subset of $\{0,1\}^n \times \Sigma$, where Σ is some finite output alphabet. We will equate a relation $f \subseteq \{0,1\}^n \times \Sigma$ with a function that maps $\{0,1\}^n$ to subsets of Σ, so that for a string $x \in \{0,1\}^n$, the notation $f(x)$ denotes $\{\sigma \in \Sigma : (x,\sigma) \in f\}$. An algorithm which computes a relation f to error ϵ must have the guarantee that for inputs $x \in \{0,1\}^n$, the algorithm outputs a symbol in $f(x)$ with probability at least $1-\epsilon$.

Relations are generalizations of partial functions. This is because we can represent a partial function f with domain $\mathrm{Dom}(f) \subseteq \{0,1\}^n$ by a relation f' such that $f'(x) = \{f(x)\}$ for $x \in \mathrm{Dom}(f)$ and $f'(x) = \{0,1\}$ for $x \notin \mathrm{Dom}(f)$. In other words, the relational version f' of the partial function f will accept all input strings (it will be a total function), but it will consider every output symbol to be valid when given an input not in $\mathrm{Dom}(f)$. This essentially makes the inputs not in $\mathrm{Dom}(f)$ become trivial, and hence makes the relation f' intuitively equivalent to the partial function f.

We will generalize several of our query measures to relations.

Critical (fractional) block sensitivity. The original definition of $\mathrm{cbs}(f)$ from [27] actually defined it for relations. We say that a total function $f' \colon \{0,1\}^n \to \Sigma$ is a *completion* of a relation $f \subseteq \{0,1\}^n \times \Sigma$ if $(x, f'(x)) \in f$ for all $x \in \{0,1\}^n$. In other words, f' is a completion if it gives a fixed, valid output choice for each input to f. Next, we say an input $x \in \{0,1\}^n$ is *critical* if it has a unique valid output symbol in f; that is, if $|f(x)| = 1$. We let $\mathrm{crit}(f)$ denote the set of all critical inputs to f. (Note that if f is the relational version of a partial function, then $\mathrm{crit}(f)$ is equal to the domain of the partial function.) We then define

$$\mathrm{cbs}(f) \coloneqq \min_{f'} \max_{x \in \mathrm{crit}(f)} \mathrm{bs}(x, f')$$

$$\mathrm{cfbs}(f) \coloneqq \min_{f'} \max_{x \in \mathrm{crit}(f)} \mathrm{fbs}(x, f'),$$

where the minimizations are over completions f' of f. Observe that if f is the relational version of a partial function, these definitions match the previous ones.

Adversary bounds. The adversary bounds easily generalize to relations: both the positive adversary bound and the classical adversary bound will still be minimizations over weight schemes $q(x,i)$, with a non-negative weight assigned to each pair of input in $\{0,1\}^n$ and $i \in [n]$. The objective value to be minimized is the same as before: $\max_{x \in \{0,1\}^n} \sum_{i \in [n]} q(x,i)$. As for the constraints, we previously had one constraint for each pair of inputs x, y with $f(x) \neq f(y)$. For relations, we will replace this condition with the condition $f(x) \cap f(y) = \varnothing$ (that is, x and y have disjoint allowed-output-symbol sets). Hence the new constraint for $\mathrm{Adv}(f)$ becomes that for all pairs $x, y \in \{0,1\}^n$ with $f(x) \cap f(y) = \varnothing$, we have

$$\sum_{i : x_i \neq y_i} \sqrt{q(x,i)q(y,i)} \geq 1.$$

Similarly, the constraint for $\mathrm{CAdv}(f)$ is that for all pairs $x, y \in \{0,1\}^n$ with $f(x) \cap f(y) = \varnothing$, we have

$$\sum_{i : x_i \neq y_i} \min\{q(x,i), q(y,i)\} \geq 1,$$

and the constraint for $\mathrm{Adv}_1(f)$ is similar.

2.3.1 Degree measures

Degree of a function. For a (possibly partial) Boolean function f, we define its *degree* to be the minimum degree of a real polynomial p which satisfies $p(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$ as well as $p(x) \in [0,1]$ for all $x \in \{0,1\}^n$. We denote this by $\deg(f)$.

Approximate degree. For a (possibly partial) Boolean function f, we define its *approximate degree* to error ϵ to be the minimum degree of a real polynomial p which satisfies $|p(x)-f(x)| \leq \epsilon$ for all $x \in \mathrm{Dom}(f)$ as well as $p(x) \in [0,1]$ for all $x \in \{0,1\}^n$. We denote this by $\widetilde{\deg}_\epsilon(f)$. When $\epsilon = 1/3$, we omit it and write $\widetilde{\deg}(f)$.

These measures are both defined and discussed in the survey by Buhrman and de Wolf [15]. We note that for partial functions, some authors do not include the requirement that the polynomial approximating the function is bounded outside of the promise set. Without this requirement, one gets a smaller measure. In this work we will only use degree and approximate degree to refer to the bounded versions of these measures.

We also note that approximate degree can be *amplified*: if a polynomial p approximates a function f to error ϵ, then we can modify p to get a polynomial q which approximates f to error $\epsilon' < \epsilon$ and which has degree that is at most a constant factor larger than the degree of p (this constant factor will depend on ϵ and ϵ').

2.3.2 Known relationships between measures

See Figure 1 for a summary of relationships between these measures (for partial functions).

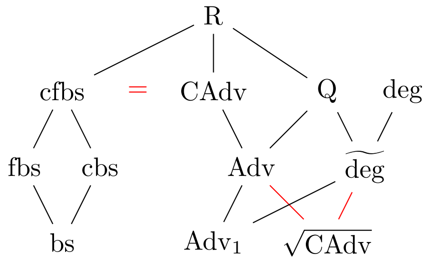


Figure 1 Relations between query complexity measures used in this work, applicable to partial functions. An upwards line from $\mathrm{M}_1(f)$ to $\mathrm{M}_2(f)$ means that $\mathrm{M}_1(f) = O(\mathrm{M}_2(f))$ for all (possibly partial) Boolean functions f. Red indicates new relationships proved in this work. We warn that some of these relationships are false for relations; in particular, CAdv may be strictly larger than cfbs and its square root may be incomparable to $\widetilde{\deg}$ for relations.

It is not hard to see that $\mathrm{bs}(f)$ is the smallest of the block sensitivity measures, and $\mathrm{cfbs}(f)$ is the largest. We know [2, 27] that $\mathrm{fbs}(f)$ and $\mathrm{cbs}(f)$ both lower bound $\mathrm{R}(f)$ for all (possibly partial) Boolean functions f; in Section 6, we show that $\mathrm{cfbs}(f)$ is also a lower bound.

We know [30, 39, 6] that $\mathrm{Q}(f) = \Omega(\mathrm{Adv}(f))$ and $\mathrm{R}(f) = \Omega(\mathrm{CAdv}(f))$. Although this it not ordinarily stated for relations, both lower bounds hold when f is a relation as well. In Section 6, we show that $\mathrm{CAdv}(f) = \Theta(\mathrm{cfbs}(f))$ for all partial functions f, and we also show that $\mathrm{CAdv}(f) = O(\mathrm{Adv}(f)^2)$ which holds for both partial functions and relations.

Approximate degree lower bounds quantum query complexity: $\mathrm{Q}(f) = \Omega(\widetilde{\deg}(f))$. It is known [9] that approximate degree is lower-bounded by $\sqrt{\mathrm{bs}(f)}$. Tal [40] showed that for total functions, $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{fbs}(f)})$. In Section 6, we extend this result to partial functions, and also prove that $\widetilde{\deg}(f) = \Omega(\sqrt{\mathrm{cfbs}(f)})$.

In conclusion, $\mathrm{CAdv}(f)$ turns out to be the same as $\mathrm{cfbs}(f)$ for partial functions, and its square root lower bounds both $\mathrm{Adv}(f)$ and $\widetilde{\deg}(f)$, both of which are lower bounds on $\mathrm{Q}(f)$. Without taking square roots, $\mathrm{CAdv}(f)$ is a lower bound on $\mathrm{R}(f)$ but not on $\mathrm{Q}(f)$.

When we move from partial functions to relations, the measure $\mathrm{CAdv}(f)$ appears to get stronger in comparison to the other measures: it is strictly larger than $\mathrm{cfbs}(f)$, and appears to be incomparable to $\widetilde{\deg}(f)$ (though defining the latter for relations is a bit tricky, and we don't do so in this work).

2.4 Communication complexity

In the communication model, two parties, Alice and Bob, are given inputs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ respectively, and in the most general case the task is to jointly compute a relation $f \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ by communicating with each other. In other words, on input (x, y), Alice and Bob must output a symbol $z \in \mathcal{Z}$ such that $(x, y, z) \in f$. Without loss of generality, we can assume Alice sends the first message, and Bob produces the output of the protocol.

In the classical randomized model, Alice and Bob are allowed to use shared randomness R (and also possibly private randomness R_A and R_B) in order to achieve this. The cost of a communication protocol Π, denoted by $\mathrm{CC}(\Pi)$ is the number of bits exchanged between Alice and Bob. The randomized communication complexity of a relation f with error ϵ, denoted by $\mathrm{R}^{\mathrm{CC}}_{\epsilon}(f)$, is defined as the minimum $\mathrm{CC}(\Pi)$ of a randomized protocol Π that computes f with error at most ϵ on every input.

Classical information complexity. The information complexity of a protocol with inputs X, Y according to μ, shared randomness R and transcript Π is given by

$$\mathrm{IC}(\Pi, \mu) = I(X : \Pi | YR)_{\mu} + I(Y : \Pi | XR)_{\mu}.$$

For any μ we have, $\mathrm{IC}(\Pi, \mu) \leq \mathrm{CC}(\Pi)$.

Quantum communication complexity. In a quantum protocol Π, Alice and Bob initially share an entangled state on registers A_0B_0, and they get inputs x and y from a distribution μ. The global state at the beginning of the protocol is

$$|\Psi^0\rangle = \sum_{x,y} \sqrt{\mu(x,y)}\, |xxyy\rangle_{X\widetilde{X}Y\widetilde{Y}} \otimes |\Theta^0\rangle_{A_0B_0}$$

where the registers $\widetilde{X}$ and $\widetilde{Y}$ purify X and Y and are inaccessible to either party. In the t-th round of the protocol, if t is odd, Alice applies a unitary $U_t : A'_{t-1}C_{t-1} \to A'_tC_t$, on her input, her memory register A'_{t-1} and the message C_{t-1} from Bob in the previous round (where $A'_0 = XA_0$ and C_0 is empty), to generate the new message C_t, which she sends to Bob, and new memory register A'_t. Similarly, if t is even, then Bob applies the unitary $U_t : B'_{t-1}C_{t-1} \to B'_tC_t$ and sends C_t to Alice. It is easy to see that $B'_t = B'_{t-1}$ for odd t, and $A'_t = A'_{t-1}$ for even t. We can assume that the protocol is safe, i.e., for all t, $A'_t = XA_t$ and $B'_t = YB_t$, and U_t uses X or Y only as control registers. The global state at the t-th round is then

$$|\Psi^t\rangle = \sum_{x,y} \sqrt{\mu(x,y)}\, |xxyy\rangle_{X\widetilde{X}Y\widetilde{Y}} \otimes |\Theta^t\rangle_{A_tB_tC_t|xy}\,.$$

[31] (Proposition 9) showed that making a protocol safe does not decrease its QIC (defined below), so we shall often work with protocols of this form.

The quantum communication cost of a protocol Π, denoted by $\mathrm{QCC}(\Pi)$, is the total number of qubits exchanged between Alice and Bob in the protocol, i.e., $\sum_t \log |C_t|$. The quantum communication complexity of f with error ϵ, denoted by $\mathrm{Q}^{\mathrm{CC}}(f)$, is defined as the minimum $\mathrm{QCC}(\Pi)$ of a quantum protocol Π that computes f with error at most ϵ on every input.

Quantum information complexity. Given a quantum protocol Π as described above with classical inputs distributed as μ, its quantum information complexity is defined as

$$\mathrm{QIC}(\Pi, \mu) = \sum_{t \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_t | Y B'_t)_{\Psi^t} + \sum_{t \text{ even}} I(\widetilde{X}\widetilde{Y} : C_t | X A'_t)_{\Psi^t}.$$

The Holevo quantum information complexity is defined as

$$\begin{aligned}\mathrm{HQIC}(\Pi, \mu) &= \sum_{t \text{ odd}} I(X : B'_t C_t | Y)_{\Psi^t} + \sum_{t \text{ even}} I(Y : A'_t C_t | X)_{\Psi^t} \\ &= \sum_{t \text{ odd}} I(X : B_t C_t | Y)_{\Psi^t} + \sum_{t \text{ even}} I(Y : A_t C_t | X)_{\Psi^t} \quad \text{(for safe protocols).}\end{aligned}$$

For brevity, we shall often only use the classical input distribution μ as the subscript, or drop the subscript entirely, for these information quantities.

It was proved in [31], that for an r-round protocol Π^r, HQIC and QIC satisfy the following relation:

$$\mathrm{QIC}(\Pi^r, \mu) \geq \frac{1}{r}\,\mathrm{HQIC}(\Pi^r, \mu) \geq \frac{1}{2r}\,\mathrm{QIC}(\Pi^r, \mu).$$

Moreover, for any μ, $\mathrm{QIC}(\Pi, \mu) \leq \mathrm{QCC}(\Pi)$.

3 Lifting the classical adversary

3.1 The gadget and its properties

The gadget we use is the same one used in [24], called VER in that work. This is the function $\mathrm{VER}\colon \{0,1\}^2 \times \{0,1\}^2 \to \{0,1\}$ defined by $G(x,y) = 1$ if and only if $x + y$ is equivalent to 2 or 3 modulo 4, where $x, y \in \{0,1\}^2$ are interpreted as binary representations of integers in $\{0,1,2,3\}$. This gadget has the property of being *versatile*, which means that it satisfies the following three properties:

1. Flippability: given any input (x,y), Alice and Bob can perform a local operation on their respective inputs (without communicating) to get (x', y') such that $G(x', y') = 1 - G(x,y)$.
2. Random self-reducibility: given any input (x,y), Alice and Bob can use shared randomness (without communicating) to generate (x', y') which is uniformly distributed over $G^{-1}(G(x,y))$. That is, Alice and Bob can convert any 0-input into a random 0-input and any 1-input into a random 1-input, without any communication. More formally, if the domain of G is $\mathcal{X} \times \mathcal{Y}$, we require a probability distribution ν_G over pairs of permutations in $S_{\mathcal{X}} \times S_{\mathcal{Y}}$ such that for each $(x,y) \in \mathcal{X} \times \mathcal{Y}$, sampling (σ_A, σ_B) from ν_G and constructing the pair $(\sigma_A(x), \sigma_B(y))$ gives the uniform distribution over $G^{-1}(G(x,y))$.
3. Non-triviality: the function G contains AND_2 as a sub-function (and by flippability, it also contains OR_2 as a sub-function).

These three properties were established in [24] for the function VER; a gadget which satisfies them is called *versatile*. Our lifting proof will work for any versatile gadget G. We will need the following simple lemma, which allows us to generate n-tuples of inputs to G that evaluate to either a string $s \in \{0,1\}^n$ or its complement $\hat{s} \in \{0,1\}^n$. We use the notation $G^{-1}(s)$ to denote the set of all n-tuples of inputs to G that together evaluate to $s \in \{0,1\}^n$; this abuses notation slightly (we would technically need to write $(G^{\oplus n})^{-1}(s)$, where $G^{\oplus n}$ is the function we get by evaluating n independent inputs to G).

▶ **Lemma 10.** *Let $s \in \{0,1\}^m$ be a given string and G be a versatile gadget. Then there is a protocol with no communication using shared randomness between Alice and Bob, who receive inputs (a,b) in the domain of G such that*

- *If $G(a,b) = 0$, Alice and Bob produce output strings (x,y) that are uniformly distributed in $G^{-1}(s)$*
- *If $G(a,b) = 1$, Alice and Bob produce output strings (x,y) that are uniformly distributed in $G^{-1}(\hat{s}) = G^{-1}(s \oplus 1^m)$.*

Proof. Alice and Bob share independent instances of the permutations ν_G, σ_A and σ_B as randomness. Applying independent instances of ν_G, Alice and Bob can produce (x',y') that are uniformly distributed in $G^{-1}((G(a,b))^m)$: this is done by applying m independent instances of σ_A and σ_B from ν_G to a and b respectively. Now Alice and Bob know where s differs from 0^m. By applying independent instances of the local flipping operation on x' and y' at these locations, they can negate the output of G. It is clear the resultant string (x,y) is uniformly distributed in $G^{-1}(s)$ if $G(a,b) = 0$ and in $G^{-1}(\hat{s})$ if $G(a,b) = 1$. ◀

We additionally have the following lemma, which uses the non-triviality property of a versatile gadget.

▶ **Lemma 11.** *If G is a constant-sized non-trivial gadget (containing AND_2 and OR_2 as subfunctions), and μ_0 and μ_1 are uniform distributions over its 0- and 1-inputs, then any classical protocol Π for computing G with bounded error has $\mathrm{IC}(\Pi,\mu_0), \mathrm{IC}(\Pi,\mu_1) = \Omega(1)$.*

Proof. G contains the AND_2 function, and μ_0 puts uniform $\Omega(1)$ weight on the 0-inputs of the AND_2 subfunction. [8] showed that any protocol computing the AND_2 function must have $\Omega(1)$ information cost with respect to the distribution that puts $1/3$ weight on all 0-inputs of the AND_2 function. Hence any protocol for G must also have $\mathrm{IC}(\Pi,\mu_0) = \Omega(1)$. Similarly, by considering the fact that G contains the OR_2 function, we can show that $\mathrm{IC}(\Pi,\mu_1) = \Omega(1)$. ◀

Although we only need a single versatile gadget, such as VER, we will briefly remark that there is actually an infinite family of versatile gadgets, and that this family is universal (i.e. every communication function is a sub-function of some gadget in the family).

▶ **Lemma 12.** *There is a universal family of versatile gadgets.*

Proof. For ease of notation, let G denote VER. For each $n \in \mathbb{N}^+$, we define G_n to be $\textsc{Parity}_n \circ G$. Note that G_n has the signature $\{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}$. We observe that G_n is versatile for each $n \in \mathbb{N}^+$. This is because, given a single input $((x_1,x_2,\ldots,x_n),(y_1,y_2,\ldots,y_n))$ to G_n with $x_i, y_i \in \{0,1,2,3\}$ for each $i \in [n]$, Alice and Bob can locally generate the uniform distribution over all inputs with the same G_n-value. They can do this by first negating a random subset of the positions i of even size (using the flippability property of VER), and then converting each of the n resulting inputs to G into a random input to G with the same G-value.

Suppose z is the n-bit string with $z_i = G(x_i,y_i)$. Then flipping a random even subset of the bits of z is equivalent to generating a random string w that has the same parity as z. It follows that the above procedure generates a random input to G_n that has the same G_n

value as the original input, meaning that G_n is random self-reducible. By flipping any single gadget G within G_n, we can negate G_n, so it is also flippable. Finally, since G_n contains G as a sub-function, it also contains AND as a sub-function, so G_n is versatile for each $n \in \mathbb{N}^+$.

It remains to show that $\{G_n\}_n$ is universal. We note that since G contains AND as a sub-function, and since $G_n = \textsc{Parity}_n \circ G$, the function G_n contains $\textsc{Parity}_n \circ \text{AND}$ as a sub-function. The latter is the inner product function IP_n on n bits, which is well-known to be universal. Hence G_n is also universal. ◀

3.2 The lifting theorem

▶ **Theorem 13.** *Let G be a constant-sized versatile gadget such as VER, and let $f\colon \{0,1\}^n \to \Sigma$ be a relation. Then* $\text{RCC}(f \circ G) = \Omega(\text{CAdv}(f))$.

Proof. Let Π be a randomized protocol for $f \circ G$ which uses T rounds of communication (with one bit sent each round), and successfully computes $f \circ G$ with probability at least $1 - \epsilon/2$ for each input. Consider inputs XY distributed according to $\mu_z = \mu_{z_1} \otimes \ldots \otimes \mu_{z_n}$, where each μ_{z_i} is the uniform distribution over (x_i, y_i) in $G^{-1}(z_i)$. Suppose Π uses public randomness R which is independent of the inputs XY. We introduce the dependency-breaking random variables D and U [8] in the following way: D is independent of X, Y, R and is uniformly distributed on $\{0,1\}^n$. For each $i \in [n]$, if $D_i = 0$, then $U_i = X_i$, and if $D_i = 1$, then $U_i = Y_i$. Defined this way, given D_iU_i, X_i and Y_i are independent under μ_z. We shall use this algorithm to give a weight scheme $q'(z,i)$:

$$q'(z,i) = I(X_i : \Pi | X_{<i} Y D U R)_{\mu_z} + I(X_i : \Pi | Y_{<i} X D U R)_{\mu_z}$$

where $X_{<i}$ denotes $X_1 \ldots X_{i-1}$, and similarly for $Y_{<i}$. Clearly q is non-negative, and we shall show that

$$\sum_{i : z_i \neq w_i} \min\{q'(z,i), q'(w,i)\} = \Omega(1)$$

for all z, w such that $f(z) \cap f(w) = \varnothing$, where the constant in the $\Omega(1)$ is universal. Using this constant to normalize $q'(z,i)$, we get $q(z,i)$ which is a valid weight scheme. Since for any fixed value of $DU = du$, $I(X : \Pi | YR)_{\mu_{zdu}}$ is an information cost,

$$\sum_{i \in [n]} q'(z,i) = I(X : \Pi | Y D U R)_{\mu_z} + I(Y : \Pi | X D U R)_{\mu_z} \leq \text{CC}(\Pi),$$

we have for any protocol Π,

$$\text{CC}(\Pi) \geq \Omega\left(\sum_{i \in [n]} q(z,i)\right) \geq \Omega\left(\min_{\{q(z,i)\}} \sum_{i \in [n]} q(z,i)\right)$$

where the minimization is over all valid weight schemes. This proves the result.

Let z and w be two inputs to f such that $f(z) \cap f(w) = \varnothing$. Suppose z and w differ on indices in the block $\mathcal{B}$. Let $\mathcal{B}^1$ be the subset of indices in $\mathcal{B}$ where $\min\{q'(z,i), q'(w,i)\}$ is achieved by $q'(z,i)$, and $\mathcal{B}^2$ be the subset where the minimum is achieved by $q'(w,i)$. For an index $i \in \mathcal{B}^1$, let $\mathcal{B}^1_i$ denote $\mathcal{B}^1 \cap [i-1]$, and $\mathcal{B}^2_i$ denote $\mathcal{B}^2 \cap [i-1]$. We also use $\mathcal{B}^{1,c}$ to denote $[n] \setminus \mathcal{B}^1$, and $\mathcal{B}^{1,c}_i$ to denote $[i-1] \setminus \mathcal{B}^1_i$. Then,

$$
\begin{aligned}
&\sum_{i:z_i\neq w_i} \min\{q'(z,i), q'(w,i)\}\\
&= \sum_{i\in\mathcal{B}^1} (I(X_i : \Pi|X_{<i}YDUR)_{\mu_z} + I(Y_i : \Pi|Y_{<i}XDUR)_{\mu_z})\\
&\quad + \sum_{i\in\mathcal{B}^2} (I(X_i : \Pi|X_{<i}YDUR)_{\mu_w} + I(Y_i : \Pi|Y_{<i}XDUR)_{\mu_w})\\
&\overset{(1)}{=} \frac{1}{2}\sum_{i\in\mathcal{B}^1} (I(X_i : \Pi|X_{<i}YD_{-i}U_{-i}R)_{\mu_z} + I(Y_i : \Pi|Y_{<i}XD_{-i}U_{-i}R)_{\mu_z})\\
&\quad + \frac{1}{2}\sum_{i\in\mathcal{B}^2} (I(X_i : \Pi|X_{<i}YD_{-i}U_{-i}R)_{\mu_w} + I(Y_i : \Pi|Y_{<i}XD_{-i}U_{-i}R)_{\mu_w})\\
&\overset{(2)}{\geq} \frac{1}{2}\sum_{i\in\mathcal{B}^1} (I(X_i : \Pi|X_{\mathcal{B}^1_i}Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z} + I(Y_i : \Pi|Y_{\mathcal{B}^1_i}X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z})\\
&\quad + \frac{1}{2}\sum_{i\in\mathcal{B}^2} (I(X_i : \Pi|X_{\mathcal{B}^2_i}Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w} + I(Y_i : \Pi|Y_{\mathcal{B}^2_i}X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w})\\
&= \frac{1}{2}(I(X_{\mathcal{B}^1} : \Pi|Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z} + I(Y_{\mathcal{B}^1} : \Pi|X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_z})\\
&\quad + \frac{1}{2}(I(X_{\mathcal{B}^2} : \Pi|Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w} + I(Y_{\mathcal{B}^2} : \Pi|X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w}). \qquad (1)
\end{aligned}
$$

Above, equality (1) follows by using the definition of D_iU_i. The inequality (2) follows from the fact that given Y_i, X_i is independent of all other X_j, Y_j D_j and U_j under both the z and w distributions, hence $I(Y_i : \Pi|Y_{<i}X(DU)_{-i}R)_{\mu_z} \geq I(Y_i : \Pi|Y_{\mathcal{B}^1_i}X_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}}R)_{\mu_z}$, and equivalent inequalities hold for the other terms.

Consider $v \in \{0,1\}^n$ which agrees with w on the bits in $\mathcal{B}^1$, with z on the bits in $\mathcal{B}^2$, and with both of them outside $\mathcal{B}$. Since $f(z)$ and $f(w)$ are disjoint, at least one of the following must be true:

1. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y) \in f(z)] \leq \frac{1}{2}$
2. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y) \in f(w)] \leq \frac{1}{2}$.

In case 1, we shall give a protocol Π' that computes G correctly with probability at least $1-\epsilon$ in the worst case, such that

$$\mathrm{IC}(\Pi', \mu_0) = O(I(X_{\mathcal{B}^1} : \Pi|Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z} + I(Y_{\mathcal{B}^1} : \Pi|X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R)_{\mu_z}).$$

Similarly, in case 2, we can use Π to give a protocol Π'' for G, such that

$$\mathrm{IC}(\Pi'', \mu_1) = O(I(X_{\mathcal{B}^2} : \Pi|Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w} + I(Y_{\mathcal{B}^2} : \Pi|X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}}R)_{\mu_w}).$$

Due to equation (1) and Lemma 11, this proves the theorem.

In fact we only show how to construct the protocol Π' in case 1; the construction of Π'' is identical. Since z is in the domain of f and Π has worst case correctness for $f \circ G$, we must have $\Pr_{(x,y)\sim\mu_z}[\Pi(x,y) \in f(z)] \geq 1-\epsilon/2$. Therefore, in case 1, Π can distinguish between samples from μ_z and μ_v on average: on getting a sample from μ_z or μ_v, we can run Π to see if it gives an output in $f(z)$ or not, and output z or v accordingly. This average distinguishing probability can be boosted by running Π multiple times.

In Π', Alice and Bob will share $RD_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R_AR_B$ as randomness, where we use R_A and R_B to denote Alice and Bob's part of the shared randomness from Lemma 10, required to generate $z_{\mathcal{B}^1}$ if $G(a,b) = 0$ and $v_{\mathcal{B}^1}$ if $G(a,b) = 1$. On input (a,b) to G, Alice and Bob do the following k times for $k = \frac{2}{\epsilon^2}\ln(1/\epsilon)$:

- Alice sets $x_{\mathcal{B}^1} = R_A(a)$ and Bob sets $y_{\mathcal{B}^1} = R_B(b)$.
- Alice samples $x_{\mathcal{B}^{1,c}}$ and Bob $y_{\mathcal{B}^{1,c}}$ from private randomness, so that $G^{|\mathcal{B}^{1,c}|}(x_{\mathcal{B}^{1,c}}, y_{\mathcal{B}^{1,c}}) = z_{\mathcal{B}^{1,c}}$. They can do this since given $(DU)_{\mathcal{B}^{1,c}}$, $X_{\mathcal{B}^{1,c}}$ and $Y_{\mathcal{B}^{1,c}}$ are independent under μ_z and μ_v.
- They run Π on this (x, y) and generate the corresponding output.

(There are k independent instances of the D, U, R_A, R_B variables for each run above, but we denote all of them the same way for brevity.) The final output of Π' is 1 if the number of runs which have given an output in $f(z)$ is at least $(1-\epsilon)k$, and 0 otherwise.

Clearly if $G(a, b) = 0$, then (x, y) generated this way is uniformly distributed in the support of μ_z, and if $G(a, b) = 1$, then (x, y) is uniform in the support of μ_v. Calling the protocol in the i-th round of Π', Π_i, notice that the transcript of each Π_i is independent of AR_A, where A is the random variable for Alice's input, given the generated $X_{\mathcal{B}^1}$ (and this holds true even conditioned on $BR_B D_{\mathcal{B}^{1,c}} U_{\mathcal{B}^{1,c}} R$). Moreover, both $X_{\mathcal{B}^1}$ and Π_i are independent of BR_B given $Y_{\mathcal{B}^1}$ and of $Y_{\mathcal{B}^1}$ given BR_B (even conditioned on $D_{\mathcal{B}^1} U_{\mathcal{B}^1} R$). Let $\mu_{0,z}$ denote the distribution of $ABR_A R_B (DU)_{\mathcal{B}^{1,c}}$ when $G(a, b) = 0$, which induces the distribution $\mu_{z_{\mathcal{B}^1}}$ on $X_{\mathcal{B}^1} Y_{\mathcal{B}^1}$. Hence,

$$\begin{aligned}
I(A : \Pi_i | BR_A R_B (DU)_{\mathcal{B}^1} R)_{\mu_{0,z}} &\leq I(AR_A : \Pi_i | BR_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(1)}{\leq} I(X_{\mathcal{B}^1} : \Pi_i | BR_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(2)}{\leq} I(X_{\mathcal{B}^1} : \Pi_i | Y_{\mathcal{B}^1} BR_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(3)}{=} I(X_{\mathcal{B}^1} : \Pi_i | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \\
&\overset{(4)}{=} I(X_{\mathcal{B}^1} : \Pi | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z}.
\end{aligned}$$

where inequality (1) follows from the fact that $I(U' : V|W) \leq I(U : V|W)$ if U' is independent of V given UW, (2) follows from the fact that $I(U : V|W) \leq I(U : V|WW')$ if V is independent of W' given W, and (3) follows from $I(U : V|WW') = I(U : V|W')$ if U and V are both independent of W given W'. Finally, (4) follows from the definition of Π_i and the the fact that the variables AB don't appear in the expression, so we can switch from $\mu_{0,z}$ to μ_z. Similarly,

$$I(B : \Pi_i | AR_A R_B (DU)_{\mathcal{B}^{1,c}} R)_{\mu_{0,z}} \leq I(Y_{\mathcal{B}^1} : \Pi | X_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z},$$

which lets us conclude that

$$\mathrm{IC}(\Pi_i, \mu_0) \leq I(X_{\mathcal{B}^1} : \Pi | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z} + I(Y_{\mathcal{B}^1} : \Pi | X_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z}.$$

Thus $\mathrm{IC}(\Pi', \mu_0)$ is at most $k(I(X_{\mathcal{B}^1} : \Pi | Y_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z} + I(Y_{\mathcal{B}^1} : \Pi | X_{\mathcal{B}^1} (DU)_{\mathcal{B}^{1,c}} R)_{\mu_z})$.

Now let us analyze the worst case error made by Π'. Since the output of Π_i on (a, b) is expected output of Π on (x, y) uniformly sampled from either μ_z or μ_v, Π_i produces an output in $f(z)$ on (a, b) such that $G(a, b) = 0$ with probability at least $1 - \epsilon/2$, and on (a, b) such that $G(a, b) = 1$ with probability at most $\frac{1}{2}$. Hence by the Hoeffding bound, the probability of $(1-\epsilon)k$ many 0 outputs in the first case is at least

$$1 - e^{-\epsilon^2 k/2} \geq 1 - \epsilon$$

and in the second case is at most

$$e^{-2(1/2-\epsilon)^2 k} \leq \epsilon.$$

Hence the probability of error on either input is at most ϵ. ◀

4 Quantum bounded-round lifting

The following result, analogous to Lemma 11 except with round dependence, holds in the quantum case.

▶ **Lemma 14.** *Let G be a constant-sized gadget which contains AND_2 and OR_2 as subfunctions, and μ_0 and μ_1 be uniform distributions over its 0- and 1-inputs. Then any r-round quantum protocol Π for computing G with bounded error has $\mathrm{QIC}(\Pi,\mu_0), \mathrm{QIC}(\Pi,\mu_1) = \tilde{\Omega}(1/r)$.*

The lemma has a similar proof to the classical case, and invokes the near-optimal lower bound for the quantum information cost of the AND_2 and OR_2 functions due to [12].

▶ **Theorem 15.** *If G is a constant-sized versatile gadget, then $\mathrm{QCC}^r(f\circ G) = \tilde{\Omega}(\mathrm{CAdv}(f)/r^2)$.*

Proof. For an r-round quantum protocol Π that computes $f \circ G$ to error at most $\epsilon/2$, we define

$$q'(z,i) = \sum_{t \text{ odd}} I(X_i : B_tC_t|X_{<i}YDU)_{\mu_z} + \sum_{t \text{ even}} I(Y_i : A_tC_t|Y_{<i}XDU)_{\mu_z}$$

where the the distribution μ_z and correlation-breaking variables DU are as in the classical case. Clearly,

$$\begin{aligned}\frac{1}{r}\sum_{i=1}^{n} q'(z,i) &= \frac{1}{r}\sum_{t \text{ odd}} I(X : B_tC_t|YDU)_{\mu_z} + \sum_{t \text{ even}} I(Y : A_tC_t|XDU)_{\mu_z}\\ &= \frac{1}{r}\mathrm{HQIC}(\Pi,\mu_z) \leq \mathrm{QIC}(\Pi,\mu_z) \leq \mathrm{QCC}(\Pi).\end{aligned}$$

Clearly $q'(z,i)$ is non-negative, and for all z,w such that $f(z)\cap f(w) = \varnothing$, we shall show that

$$\sum_{i:z_i\neq w_i} \min\{q'(z,i), q'(w,i)\} = \tilde{\Omega}(1/r). \tag{2}$$

Thus, defining $q(z,i)$ as our weight scheme by normalizing $q'(z,i)$ with the r factor, we get the required result.

Showing (2) proceeds very similar to the classical case. For two inputs z,w to f such that $f(z)\cap f(w) = \varnothing$, which differ on the bits in block $\mathcal{B}$, let $\mathcal{B}^1 \subseteq \mathcal{B}$ be the indices where $\min\{q'(z,i), q'(w,i)\}$ is achieved by $q'(z,i)$, and $\mathcal{B}^2 \subseteq \mathcal{B}$ be the indices where it is achieved by $q'(w,i)$. By the same chain of inequalities as in the classical case, we have

$$\begin{aligned}&\sum_{i:z_i\neq w_i} \min\{q'(z,i), q'(w,i)\}\\ &\geq \frac{1}{2}\left(\sum_{t \text{ odd}} I(X_{\mathcal{B}^1} : B_tC_t|Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}})_{\mu_z} + \sum_{t \text{ even}} I(Y_{\mathcal{B}^1} : A_tC_t|X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{2,c}})_{\mu_z}\right)\\ &\quad + \frac{1}{2}\left(\sum_{t \text{ odd}} I(X_{\mathcal{B}^2} : B_tC_t|Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}})_{\mu_w} + \sum_{t \text{ even}} I(Y_{\mathcal{B}^2} : A_tC_t|X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}})_{\mu_w}\right).\end{aligned}$$

Note that if we had used a QIC-based definition, instead of an HQIC-based definition, for $q'(z,i)$, where we conditioned on the B_t, A_t registers, the above chain of inequalities would not have been valid, since X_i is not independent of $X_jY_jD_jU_j$ at $j\neq i$ conditioned on B_t, and the same holds for Y_i.

Define the hybrid input v which agrees with w on the bits in $\mathcal{B}^1$, with z on the bits in $\mathcal{B}^2$ and with both outside $\mathcal{B}$. At least one of the following is true of v:

1. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y) \in f(z)] \leq \frac{1}{2}$
2. $\Pr_{(x,y)\sim\mu_v}[\Pi(x,y) \in f(w)] \leq \frac{1}{2}$.

In case 1, we shall give a protocol Π' that computes G correctly with probability at least $1-\epsilon$ in the worst case, such that

$$\begin{aligned}&\mathrm{HQIC}(\Pi',\mu_0)\\ &= O\left(\sum_{t\text{ odd}} I(X_{\mathcal{B}^1} : B_tC_t|Y_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}})_{\mu_z} + \sum_{t\text{ even}} I(Y_{\mathcal{B}^1} : A_tC_t|X_{\mathcal{B}^1}D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}})_{\mu_z}\right).\end{aligned}$$

Similarly, in case 2, we can use Π to give a protocol Π'' for G, such that

$$\begin{aligned}&\mathrm{HQIC}(\Pi'',\mu_1)\\ &= O\left(\sum_{t\text{ odd}} I(X_{\mathcal{B}^2} : B_tC_t|Y_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}})_{\mu_w} + \sum_{t\text{ even}} I(Y_{\mathcal{B}^2} : A_tC_t|X_{\mathcal{B}^2}D_{\mathcal{B}^{2,c}}U_{\mathcal{B}^{2,c}})_{\mu_w}\right).\end{aligned}$$

The number of rounds in Π' and Π'' will be kr, for $k = \frac{2}{\epsilon^2}\ln(1/\epsilon)$. This proves the theorem due to Lemma 14, and the fact that $\mathrm{HQIC}(\Pi',\mu) = \Omega(\mathrm{QIC}(\Pi',\mu))$ for any μ.

We only describe the protocol Π'. In Π', Alice and Bob will share the initial entangled state of Π, as well as $D_{\mathcal{B}^{1,c}}U_{\mathcal{B}^{1,c}}R_AR_B$ as randomness, where R_A and R_B are Alice and Bob's parts of the shared randomness from Lemma 10. Note that sharing randomness is equivalent to sharing an entangled state whose Schmidt coefficients are equal to the square roots of the corresponding probabilities, and locally measuring this state to get classical variables to use. We denote the inputs of Π' by (x',y') here to avoid confusion with the memory registers. On input (x',y'), Alice and Bob do the following k times in Π':

- Alice sets $x_{\mathcal{B}^1} = R_A(x')$ and Bob sets $y_{\mathcal{B}^1} = R_B(y')$.
- Alice samples $x_{\mathcal{B}^{1,c}}$ and Bob samples $y_{\mathcal{B}^{1,c}}$ from private randomness (this can be done unitarily), so that $G^{|\mathcal{B}^{1,c}|}(x_{\mathcal{B}^{1,c}}, y_{\mathcal{B}^{1,c}}) = z_{\mathcal{B}^{1,c}}$. They can do this since given $(DU)_{\mathcal{B}^{1,c}}$, $X_{\mathcal{B}^{1,c}}$ and $Y_{\mathcal{B}^{1,c}}$ are independent under μ_z and μ_v.
- They run Π on this (x,y) and generate the corresponding output.

The final output of Π' is 1 if the number of runs which have given an output in $f(z)$ is at least $(1-\epsilon)k$, and 0 otherwise.

Let $\mu_{0,z}$ denote the distribution of $X'Y'R_AR_B(DU)_{\mathcal{B}^{1,c}}$ when $G(x',y') = 0$, which induces $\mu_{z_{\mathcal{B}^1}}$ on $X_{\mathcal{B}^1}Y_{\mathcal{B}^1}$. Let $C_{t,i}$ denote the message and $A_{t,i}, B_{t,i}$ the memory registers of the i-th run of Π in Π', which we denote by Π_i. (There are also independent D, U, R_A, R_B variables for each run, but we drop the i dependence here.) For every i, and an odd round t, we have similar to the classical case,

$$I(X' : B_{t,i}C_{t,i}|Y'R_AR_B(DU)_{\mathcal{B}^{1,c}})_{\mu_{0,z}} \leq I(X_{\mathcal{B}^1} : B_tC_t|Y_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z}$$

Similarly, for even t,

$$I(Y' : A_{t,i}C_{t,i}|X'R_AR_B(DU)_{\mathcal{B}^{1,c}})_{\mu_{0,z}} \leq I(Y_{\mathcal{B}^1} : A_tC_t|X_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z}$$

which gives us

$$\mathrm{HQIC}(\Pi_i,\mu_0) \leq \sum_{t\text{ odd}} I(X_{\mathcal{B}^1} : B_tC_t|Y_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z} + \sum_{t\text{ even}} I(Y_{\mathcal{B}^1} : A_tC_t|X_{\mathcal{B}^1}(DU)_{\mathcal{B}^{1,c}})_{\mu_z}.$$

Finally, $\mathrm{HQIC}(\Pi',\mu_0) = k\,\mathrm{HQIC}(\Pi_i,\mu_0)$.

Since z is in the domain of f and Π is correct for $f \circ G$ with probability at least $1-\epsilon/2$, we have $\Pr_{(x,y)\sim\mu_z}[\Pi(x,y) \in f(z)] \geq 1-\epsilon/2$, and the probability when (x,y) is sampled according to μ_v instead is at most $\frac{1}{2}$. Therefore, by the definition of Π' and the Hoeffding bound, Π' is correct for G with probability at least $1-\epsilon$. This completes the proof. ◀

5 Towards a full quantum adversary lifting theorem

In this section, we will prove a conditional lifting theorem for a somewhat weak quantum adversary method, Adv_1. The goal of this section is primarily to introduce some tools that we believe will be helpful in eventually proving a lifting theorem for the positive-weight quantum adversary method (hopefully with a constant-sized gadget such as the VER). Specifically, we prove a product-to-sum reduction for quantum information cost in Section 5.2, which should be helpful for handling the $\sqrt{q(z,i)q(w,i)}$ terms that occur in the positive-weight adversary method; indeed, we use this product-to-sum reduction for our Adv_1 lifting theorem. We also show how lifting theorems for quantum adversary methods are related to 2-party secure communication.

We now introduce the definition of $\mathrm{QICZ}(G)$, our measure of the information leak that must happen in any purported 2-party secure computation of G.

▶ **Definition 16.** *Let $G\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ be a communication function. Let P be the set of all communication protocols which solve G to worst-case error $1/3$. Let Δ_0 be the set of all probability distributions over $G^{-1}(0)$, and let Δ_1 be the set of all probability distributions over $G^{-1}(1)$. We define*

$$\mathrm{QICZ}(G) \coloneqq \inf_{\Pi\in P}\ \sup_{\mu\in\Delta_0\cup\Delta_1} \mathrm{QIC}(\Pi,\mu).$$

We note that since $\mathrm{QIC}(\Pi,\cdot)$ is a continuous function of distributions [12], the inner supremum is actually attained as a maximum. We can now state our lifting theorem, as follows.

▶ **Theorem 17.** *Let $f\colon\{0,1\}^n\to\Sigma$ be a relation (where $n\in\mathbb{N}^+$ and Σ is a finite alphabet) and let $G\colon\mathcal{X}\times\mathcal{Y}\to\{0,1\}$ be a communication function which contains both AND_2 and OR_2 as subfunctions. Then*

$$\mathrm{QCC}(f\circ G)=\tilde{\Omega}(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)).$$

5.1 A minimax for QICZ

Before attacking the proof of Theorem 17, we first prove a minimax theorem for the measure $\mathrm{QICZ}(G)$, giving an alternate characterization of it. To do so, we invoke Sion's minimax theorem [38].

▶ **Fact 18** (Sion's minimax). *Let V_1 and V_2 be real topological vector spaces, and let $X\subseteq V_1$ and $Y\subseteq V_2$ be convex. Let $\alpha\colon X\times Y\to\mathbb{R}$ be semicontinuous and saddle. If either X or Y is compact, then*

$$\inf_{x\in X}\sup_{y\in Y}\alpha(x,y)=\sup_{y\in Y}\inf_{x\in X}\alpha(x,y).$$

To understand the statement of this theorem, we need a few definitions:

1. A real-valued function ϕ is *convex* if $\phi(\lambda x_1+(1-\lambda)x_2)\le\lambda\phi(x_1)+(1-\lambda)\phi(x_2)$ for all $x_1,x_2\in\mathrm{Dom}(\phi)$ and all $\lambda\in(0,1)$.
2. A real-valued function ϕ is *concave* if $\phi(\lambda x_1+(1-\lambda)x_2)\ge\lambda\phi(x_1)+(1-\lambda)\phi(x_2)$ for all $x_1,x_2\in\mathrm{Dom}(\phi)$ and all $\lambda\in(0,1)$.
3. A function $\alpha\colon X\times Y\to\mathbb{R}$ is *saddle* if $\alpha(\cdot,y)$ is convex as a function of x for each fixed $y\in Y$, and if $\alpha(x,\cdot)$ is concave as a function of y for each fixed $x\in X$.

4. A real-valued function ϕ is *upper semicontinuous* at a point x if for any $\epsilon > 0$, there exists a neighborhood U of x such that for all $x' \in U$, we have $\phi(x') < \phi(x) + \epsilon$.
5. A real-valued function ϕ is *lower semicontinuous* at a point x if for any $\epsilon > 0$, there exists a neighborhood U of x such that for all $x' \in U$, we have $\phi(x') > \phi(x) - \epsilon$.
6. A function $\alpha\colon X \times Y \to \mathbb{R}$ is *semicontinuous* if $\alpha(\cdot, y)$ is lower semicontinuous over all of X for each $y \in Y$ and if $\alpha(x, \cdot)$ is upper semicontinuous over all of Y for each $x \in X$.

We now use Sion's minimax theorem to prove a minimax theorem for QICZ.

▶ **Theorem 19.** *Fix a communication function G. Let P be the set of all protocols which solve G to worst-case error $1/3$, let Δ_0 be the set of probability distributions over 0-inputs to G, and let Δ_1 be the set of probability distributions over 1-inputs to G. Then*

$$\frac{1}{2} \max_{\substack{\mu_0 \in \Delta_0 \\ \mu_1 \in \Delta_1}} \inf_{\Pi \in P} \mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1) \leq \mathrm{QICZ}(G) \leq \max_{\substack{\mu_0 \in \Delta_0 \\ \mu_1 \in \Delta_1}} \inf_{\Pi \in P} \mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1).$$

Moreover, the maximum is attained.

Proof. We will aim to use Sion's minimax theorem [38]. To this end, we start with a bit of formalism. The set P of protocols is, of course, an infinite set, and has somewhat complicated structure. In order to apply a minimax theorem, however, we want to switch over to a convex subset of a real topological vector space. To do so, we first consider the free real vector space over P, which we denote by $V(P)$. This is the real vector space consisting of all formal (finite) linear combinations of elements in P; the set P is a basis of this vector space. We further consider the 1-norm on this space, where we define the 1-norm of a formal (finite) linear combination as the sum of absolute values of coefficients in the linear combination. This norm induces a topology over $V(P)$, making it a real topological vector space.

Our set of algorithms will be the subset of $V(P)$ consisting of vectors with norm 1 that have non-negative coefficients in the linear combination; we denote this subset by R. It is not hard to see that the elements of R are simply all the finite-support probability distributions over protocols in P. We observe that R is a convex set. This will be the set over which we take the infimum in Sion's minimax theorem.

Observe that since the input set $\mathrm{Dom}(G)$ of G is finite, the sets Δ_0 and Δ_1 are both convex, compact subsets of the real vector space $\mathbb{R}^{|\mathrm{Dom}(G)|}$, which has a standard topology. It follows that the set $\Delta_0 \times \Delta_1$ is also convex and compact (as a subset of the real topological vector space $\mathbb{R}^{2|\mathrm{Dom}(G)|}$). This will be the set over which we take the supremum in Sion's minimax.

Let $A \in R$. This is a finite-support probability distribution over protocols in P; however, it is always possible to use shared randomness to construct a single protocol $\Pi_A \in P$ whose behavior exactly matches that of A (that is, in Π_A, Alice and Bob will sample a protocol from A using shared randomness, and then run that protocol). Finally, we define $\alpha\colon R \times (\Delta_0 \times \Delta_1) \to [0, \infty)$ by setting

$$\alpha(A, (\mu_0, \mu_1)) \coloneqq \mathrm{QIC}(\Pi_A, \mu_0) + \mathrm{QIC}(\Pi_A, \mu_1).$$

This will be the function on which we apply Sion's minimax.

It remains to show that α is semicontinuous and quasisaddle. It is not hard to see that the sum of two semicontinuous functions (on the same domain) is semicontinuous, and that the sum of two saddle functions is saddle. It will therefore be sufficient to show that QIC is semicontinuous and saddle.

In [41] (Lemma 5), it was shown that $\mathrm{QIC}(\cdot,\mu)$ is linear (and hence convex) for each μ. In [41] (Lemma 6), it was shown that $\mathrm{QIC}(\Pi,\cdot)$ is concave. Hence QIC is saddle, and therefore so is α. In [12] (Lemma 4.8), it was shown that $\mathrm{QIC}(\Pi,\cdot)$ is continuous.

It remains to show the lower semicontinuity of $\mathrm{QIC}(\cdot,\mu)$. More explicitly, for each fixed distribution μ, each $A\in R$ and each $\epsilon>0$, there exists $\delta>0$ such that for all $A'\in R$ with $\|A-A'\|_1<\delta$, we have $\mathrm{QIC}(\Pi_{A'},\mu)>\mathrm{QIC}(\Pi_A,\mu)-\epsilon$.

We can write $A=(1-p)B+pC$ and $A'=(1-p)B+pC'$ where $B,C,C'\in R$, and (C,C') is a pair of distributions with disjoint support. In other words, B is the probability distribution consisting of the (normalized) overlap between A and A', while C and C' are the probability distributions we get from subtracting out the overlap from A and from A' respectively. If $\|A-A'\|_1<\delta$, we must have $p<\delta/2$. Now, by the linearity of $\mathrm{QIC}(\cdot,\mu)$, we have $\mathrm{QIC}(\Pi_A,\mu)=(1-p)\,\mathrm{QIC}(\Pi_B,\mu)+p\,\mathrm{QIC}(\Pi_C,\mu)$ and $\mathrm{QIC}(\Pi_{A'},\mu)=(1-p)\,\mathrm{QIC}(\Pi_B,\mu)+p\,\mathrm{QIC}(\Pi_{C'},\mu)$. We want to choose δ so that $\mathrm{QIC}(\Pi_{A'},\mu)>\mathrm{QIC}(\Pi_A,\mu)-\epsilon$, or equivalently, so that $\mathrm{QIC}(\Pi_{C'},\mu)>\mathrm{QIC}(\Pi_C,\mu)-\epsilon/p$. This rearranges to wanting $\epsilon/p>\mathrm{QIC}(\Pi_C,\mu)-\mathrm{QIC}(\Pi_{C'},\mu)$; hence it is sufficient to choose δ so that $2\epsilon/\delta>\mathrm{QIC}(\Pi_C,\mu)-\mathrm{QIC}(\Pi_{C'},\mu)$. It is clear that such δ can always be chosen, as $\mathrm{QIC}(\Pi_C,\mu)$ must be finite.

We conclude that $\mathrm{QIC}(\cdot,\mu)$ is lower semicontinuous. Sion's minimax theorem (Fact 18) then gives

$$\begin{aligned}&\inf_{A\in R}\sup_{(\mu_0,\mu_1)\in\Delta_0\times\Delta_1}\mathrm{QIC}(\Pi_A,\mu_0)+\mathrm{QIC}(\Pi_A,\mu_1)\\ &=\sup_{(\mu_0,\mu_1)\in\Delta_0\times\Delta_1}\inf_{A\in R}\mathrm{QIC}(\Pi_A,\mu_0)+\mathrm{QIC}(\Pi_A,\mu_1).\end{aligned}$$

Since R contains P as a subset, and since every protocol in R can be converted into an equivalent protocol in P, taking an infimum over $A\in R$ is the same as taking an infimum over $\Pi\in P$. It is then clear that the left hand side is at least $\mathrm{QIC}(G)$ (since the latter has only one $\mathrm{QIC}(\Pi,\mu_0)$ or $\mathrm{QIC}(\Pi,\mu_1)$ term instead of both), but no more than twice $\mathrm{QIC}(G)$ (since the maximum of $\mathrm{QIC}(\Pi,\mu_0)$ and $\mathrm{QIC}(\Pi,\mu_1)$ is at least the average of the two). Hence the desired result follows. The attainment of the maximum comes from the fact that an upper semicontinuous function on a nonempty compact set attains is maximum, combined with the fact that a pointwise infimum of upper semicontinuous functions is upper semicontinuous. ◀

5.2 Product-to-sum reduction for quantum information

In order to prove Theorem 17, we will need a way to bound the product of quantum information cost on the "0-input" side and the quantum information cost on the "1-input" side. We start with the following definition.

▶ **Definition 20.** *Let G be a communication function. We say a distribution μ is* nontrivial *for G if for any protocol Π computing G (to bounded error against worst-case inputs), it holds that $\mathrm{QIC}(\Pi,\mu)>1/\operatorname{poly}(r)$, where r is the number of rounds of Π. (In particular, it should not be possible to achieve $\mathrm{QIC}(\Pi,\mu)=0$ if μ is nontrivial.)*

Using this definition, we state the following theorem, which is the main result of this subsection.

▶ **Theorem 21.** *Let G be a gadget, let μ_0 and μ_1 be nontrivial 0- and 1-distributions for G, and let Π be a protocol solving G (to bounded error against worst-case inputs). Then there is a protocol Π' which also solves G (to bounded error against worst-case inputs) which satisfies*

$$\mathrm{QIC}(\Pi',\mu_0)+\mathrm{QIC}(\Pi',\mu_1)=O\left(\sqrt{\mathrm{QIC}(\Pi,\mu_0)\,\mathrm{QIC}(\Pi,\mu_1)}\cdot\operatorname{polylog}r\right),$$

where r is the number of rounds of Π and where the constant in the big-O is universal. Moreover, the number of rounds of Π' is polynomial in that of Π.

Before we prove this, we will need a few lemmas. In the following, we will use $G^{\oplus n}$ to denote the direct sum of n copies of G; that is, if $G\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$, then $G^{\oplus}\colon (\mathcal{X}^n)\times(\mathcal{Y}^n)\to\{0,1\}^n$ is the function that takes in n separate copies to G and outputs n separate outputs from G.

▶ **Lemma 22.** *Let (G,μ_0,μ_1) be any gadget, 0-distribution, and 1-distribution, let $n\in\mathbb{N}^+$, and let Π be a protocol which solves $G^{\oplus n}$ (to bounded error against worst-case inputs). Then there is a protocol Π' which solves G (to bounded error against worst-case inputs) which satisfies*

$$\frac{\mathrm{QIC}(\Pi',\mu_0)+\mathrm{QIC}(\Pi',\mu_1)}{2}\leq\frac{1}{n}\cdot\max_{z\in\{0,1\}^n}\mathrm{QIC}\left(\Pi,\mu_z\right).$$

Proof. Let for $z\in\{0,1\}^n$, let $\Pi_{i,z}$ be the protocol which: takes an input to G; artificially generates $n-1$ inputs from μ_{z_j} for $j\neq i$ for all the gadgets G except at position i; places the true input at position i; runs Π on the resulting input to $G^{\oplus n}$; traces out all the outputs except for position i; and returns the result. Note that $\Pi_{i,z}$ does not depend on the value of z_i, but depends on the rest of z. If we use z^i to denote the string x with i flipped, we have $\Pi_{i,z}=\Pi_{i,z^i}$ for all x and i.

[41] (Lemma 4) showed that for all $x\in\{0,1\}^n$,

$$\sum_{i=1}^{n}\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})=\mathrm{QIC}\left(\Pi,\mu_z\right).$$

Let $\Pi':=\frac{1}{n}\frac{1}{2^n}\sum_{i=1}^{n}\sum_{x\in\{0,1\}^n}\Pi_{i,z}$. Again by [41] (Lemma 5),

$$\begin{aligned}\frac{\mathrm{QIC}(\Pi',\mu_0)+\mathrm{QIC}(\Pi',\mu_1)}{2}&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\frac{\mathrm{QIC}(\Pi_{i,z},\mu_0)+\mathrm{QIC}(\Pi_{i,z},\mu_1)}{2}\\&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\frac{\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})+\mathrm{QIC}(\Pi_{i,z},\mu_{z^i_i})}{2}\\&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\frac{\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})+\mathrm{QIC}(\Pi_{i,z^i},\mu_{z^i_i})}{2}\\&=\frac{1}{n2^n}\sum_{i=1}^{n}\sum_{z\in\{0,1\}^n}\mathrm{QIC}(\Pi_{i,z},\mu_{z_i})\\&=\frac{1}{n2^n}\sum_{z\in\{0,1\}^n}\mathrm{QIC}\left(\Pi,\mu_z\right)\\&\leq\frac{1}{n}\cdot\max_{z\in\{0,1\}^n}\mathrm{QIC}\left(\Pi,\mu_z\right).\end{aligned}$$ ◀

▶ **Lemma 23.** *Let $G_1,G_2,\ldots,G_n$ be any sequence of communication tasks, and for each $i\in[n]$ let Π_i be a protocol which solves G_i (to bounded error against worst-case inputs). Let F be a (possibly partial) query function on n bits, and let Q be a T-query quantum query algorithm computing F (to bounded error against worst-case inputs). Then there is a protocol Π' computing $F\circ\{G_i\}_i$ (to bounded error against worst-case inputs) such that for any $z\in\mathrm{Dom}(F)$ and any distribution μ_z supported on $(G_1\oplus G_2\oplus\cdots\oplus G_n)^{-1}(z)$, we have*

$$\mathrm{QIC}(\Pi',\mu_z)=\tilde{O}\left(T\log\log n\cdot\max_{i\in[n]}\mathrm{QIC}(\Pi_i,\mu_z^i)\right),$$

where μ_z^i is the marginal of μ_z on gadget number i.

Proof. Let $\hat{\Pi}_i$ be the amplified and purified version of Π_i, reducing its worst-case error on G to $\delta/T^{10}\log n$ and using the uncomputing trick to clean up garbage (δ will be chosen later). Then the information cost of $\hat{\Pi}_i$ against any fixed distribution increases by a factor of at most $O(\log T + \log\log n + \log 1/\delta)$ compared to Π_i. Next, Π' be the protocol where Alice runs the query algorithm for F, and whenever she needs to make a query i, she sends i to Bob and they compute gadget number i using $\hat{\Pi}_i$. Since F succeeds with bounded error on worst-case inputs and since $\hat{\Pi}_i$ has such a low probability of error, the protocol Π' correctly computes $F \circ G$ on worst-case inputs.

Fix $z \in \mathrm{Dom}(F)$ and μ_z supported on $(G^{\oplus n})^{-1}(z)$. We will expand out $\mathrm{QIC}(\Pi', \mu_z)$. In round $t \le T$ of the query algorithm, there are two types of messages between Alice and Bob: one message from Alice to Bob containing a copy E_t of the query register D_t for step $t \le T$, which Alice knows from her simulation of the algorithm Q for F; and all the messages between Alice and Bob implementing $\hat{\Pi}_i$. Denote those messages by $C_{t,j}$. Note that E_t also gets passed back from Bob to Alice at the end of each round for cleanup purposes.

We name the rest of the registers. Let the input registers be X and Y, and let Alice hold register D_t specifying the position to query at round t, a work register $\tilde{A}_t$ related to the implementation of the algorithm Q for f (which stays untouched for all j), and register $A_{t,j}$ related to the implementation of the $\hat{\Pi}_i$ protocols for round t. Bob holds query register E_t (passed from Alice, untouched for all j) as well as work register $B_{t,j}$ for the implementation of the $\hat{\Pi}_i$ protocols. Let R be the purification register. Then using r to denote the index of the last round of the $\hat{\Pi}_i$, we have

$$\begin{aligned}\mathrm{QIC}(\Pi', \mu_z) = {} & \sum_{t=1}^{T} I(\widetilde{X}\widetilde{Y} : E_t | Y B_{t,0})_{\Psi_z^t} + \sum_{t=1}^{T} I(\widetilde{X}\widetilde{Y} : E_t | X A_{t,r} \tilde{A}_t D_t)_{\Psi_z^t} \\ & + \sum_{t=1}^{T} \sum_{j \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | Y B_{t,j} E_t)_{\Psi_z^{t,j}} \\ & + \sum_{t=1}^{T} \sum_{j \text{ even}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | X A_{t,j} D_t \tilde{A}_t)_{\Psi_z^{t,j}}.\end{aligned}$$

For the terms $I(\widetilde{X}\widetilde{Y} : E_t|YB_{t,0})_{\Psi_z^t}$ and $I(\widetilde{X}\widetilde{Y} : E_t|XA_{t,r}\tilde{A}_tD_t)_{\Psi_z^t}$, we note that $B_{t,0}$ and $A_{t,r}$ are the start state on Bob's side and the end state on Alice's side for $\hat{\Pi}$, and can be assumed to be independent of all other registers. Hence we shall ignore the registers $B_{t,0}$ and $A_{t,r}$ in the conditioning systems. Let $|\Phi^t\rangle$ that is obtained by replacing the $\tilde{A}_tD_tE_t$ registers of $|\Psi_z^t\rangle$ with the state of the query algorithm for f after t queries (with the query register D_t duplicated). $|\Phi^t\rangle_{\tilde{A}_tD_tE_t|zxy}$ depends on x and y only through z, which is fixed. Hence $I(\widetilde{X}\widetilde{Y} : E_t|Y)_{\Phi_z^t}$ and $I(\widetilde{X}\widetilde{Y} : E_t|X\tilde{A}_t)_{\Phi_z^t}$ are both 0. Clearly, Φ_z^t is the state the protocol would have been in if $\hat{\Pi}_i$ were run with 0 error. Since the protocol runs of $\hat{\Pi}_i$ make very small error instead, we have instead $\left\| |\Psi^t\rangle_z - |\Phi^t\rangle_z \right\|_1 \le \epsilon$, where $\epsilon = O(\delta/\mathrm{poly}(T)\log n)$. This implies

$$\begin{aligned} I(\widetilde{X}\widetilde{Y} : E_t|Y)_{\Psi_z^t} &= H(E_t|Y)_{\Psi_z^t} - H(E_t|\widetilde{X}\widetilde{Y}Y)_{\Psi_z^t} \\ &\le H(E_t|Y)_{\Phi_z^t} - H(E_t|\widetilde{X}\widetilde{Y}Y)_{\Phi_z^t} + 8\epsilon\log|E_t| + 4h(\epsilon) \\ &= 8\epsilon + 4h(\epsilon). \end{aligned}$$

The total sum of $I(\widetilde{X}\widetilde{Y} : E_t|YB_{t,0})_{\Psi_z^t}$ over all t is therefore at most $\delta/2$, and the same applies to $I(\widetilde{X}\widetilde{Y} : E_t|X\tilde{A}_t)_{\Psi_z^t}$.

We then have

$$\begin{aligned}
\mathrm{QIC}(\Pi', \mu_z) &\le \delta + \sum_{t=1}^{T} \left(\sum_{j \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | Y B_{t,j} E_t)_{\Psi_z^t} \right. \\
&\quad \left. + \sum_{j \text{ even}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | X A_{t,j} D_t \tilde{A}_t)_{\Psi_z^t} \right) \\
&= \delta + \sum_{t=1}^{T} \sum_{i=1}^{n} \Pr[D_t = i] \left(\sum_{j \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | Y B_{t,j})_{\Psi^t_{z, D_t = i}} \right. \\
&\quad \left. + \sum_{j \text{ even}} I(\widetilde{X}\widetilde{Y} : C_{t,j} | X A_{t,j} \tilde{A}_t)_{\Psi^t_{z, D_t = i}} \right) \\
&= \delta + \sum_{t=1}^{T} \sum_{i=1}^{n} \Pr[D_t = i] \, \mathrm{QIC}(\hat{\Pi}_i, \mu_z^i) \\
&\le \delta + T \max_i \mathrm{QIC}(\hat{\Pi}_i, \mu_z^i) \\
&\le \delta + O(T(\log T + \log\log n + \log 1/\delta) \max_i \mathrm{QIC}(\Pi_i, \mu_z^i)).
\end{aligned}$$

Setting $\delta = T \max_i \mathrm{QIC}(\Pi_i, \mu_z^i)$, but ensuring $\epsilon \le 1/3$ (since we can't amplify a negative amount), we get

$$\mathrm{QIC}(\Pi', \mu_z) = O\left(T \max_i \mathrm{QIC}(\Pi_i, \mu_z^i) \log\left(2 + \frac{T^{10} \log n}{\max_i \mathrm{QIC}(\Pi_i, \mu_z^i)} \right) \right). \qquad ◀$$

▶ **Lemma 24.** *Let G be a gadget, let μ_0 and μ_1 be a* 0*-distribution and a* 1*-distribution for G, let $n \in \mathbb{N}^+$, and let Π be a protocol computing $OR_n \circ G$ (to bounded error against worst-case inputs). Then there is a protocol Π' computing $G^{\oplus n}$ (to bounded error against worst-case inputs) such that*

$$\max_{z \in \{0,1\}^n} \mathrm{QIC}\left(\Pi', \mu_z\right) = \tilde{O}\left(\sqrt{n} \cdot \max_{z \in \{0,1\}^n} \mathrm{QIC}\left(\Pi, \mu_z\right) \right).$$

Proof. Consider the following task: the goal is to output a hidden string $z \in \{0,1\}^n$, and the allowed queries are subset-OR queries, meaning that for each subset $S \subseteq [n]$ there is a query which returns $\mathrm{OR}(z_S)$ (which equals 1 if $z_i = 1$ for some $i \in S$, and returns 0 otherwise). We can model this task as a query function F on a promise set $P \subseteq \{0,1\}^{2^n}$. Each string in P is a long encoding $u(z) \in \{0,1\}^{2^n}$ of some string $z \in \{0,1\}^n$, with the long encoding $u(z)$ being a string with $(u(z))_S = \mathrm{OR}(z_S)$ for all S. In other words, u is a function $u \colon \{0,1\}^n \to \{0,1\}^{2^n}$. The function F is defined by $F(u(z)) = z$ for all $z \in \{0,1\}^n$, where $\mathrm{Dom}(F) = \{u(z) : z \in \{0,1\}^n\}$. It is not hard to verify that this function is well-defined.

The function f is sometimes called the combinatorial group testing problem. We have $\mathrm{D}(F) \le n$, since we can query $u(z)_{\{i\}}$ for all $i \in [n]$ to get the bits z_i one by one and then output all of z. (Note that the input size to F is of length $N = 2^n$, so n does not represent the input size here.) Belovs [10] showed that $\mathrm{Q}(F) = O(\sqrt{n})$. This result will play a key role in our analysis here, which is motivated by [11] (where this algorithm of Belovs was similarly used to reduce direct-sum computations to OR computations).

Now, observe that $F \circ u$ is the identity function on n bit strings. The protocol Π' for $G^{\oplus}$ will be a protocol for $F \circ u \circ G$. We use Lemma 23 on the query function F and the communication tasks $u(G^{\oplus n})_1, u(G^{\oplus n})_2, \ldots, u(G^{\oplus n})_{2^n}$. The query algorithm for F makes

$T = O(\sqrt{n})$ queries. Each of the communication tasks is of the following form: take as input n copies to G, and output the OR of a fixed subset S of the copies of G. To solve this task, which we denote F_S, we describe a protocol Π_S. In this protocol, Alice and Bob will use their shared randomness to replace the inputs in positions $i \notin S$ by independent samples from μ_0. They will then run Π to compute the OR of the n copies of G.

The correctness of Π' is clear, so we analyze its information cost. Fix $z \in \{0,1\}^n$, and denote by z_S the string satisfying $(z_S)_i = z_i$ if $i \in S$ and $(z_S)_i = 0$ if $i \in S$. In order to upper bound $\mathrm{QIC}(\Pi', \mu_z)$ using Lemma 23, we let μ'_z be the distribution on strings of length $\{0,1\}^{n2^n}$ that we get by sampling a string from μ_z and making 2^n copies of it. We observe that that the behavior of Π' when acting on μ_z is exactly the composed behavior of the query algorithm for F composed with the protocols Π_S acting on the distribution μ'_z; Lemma 23 therefore gives us

$$\mathrm{QIC}(\Pi', \mu_z) = O\left(\sqrt{n} \cdot \max_S \mathrm{QIC}(\Pi_S, \mu_z) \log\left(2 + \frac{n^5 \log N}{\max_S \mathrm{QIC}(\Pi_S, \mu_z)}\right)\right)$$

(where we used the more precise bound given in the proof of Lemma 23). Recall that Π_S replaces the samples of μ_z that correspond to bits $i \notin S$ with freshly-generated samples from μ_0, and then runs Π; hence $\mathrm{QIC}(\Pi_S, \mu_z) = \mathrm{QIC}(\Pi_S, \mu_{z_S}) \le \mathrm{QIC}(\Pi, \mu_{z_S})$. The maximum over sets S of $\mathrm{QIC}(\Pi, \mu_{z_S})$ is clearly at most the maximum over $w \in \{0,1\}^n$ of $\mathrm{QIC}(\Pi, \mu_w)$. Using $\log N = n$, we can therefore write

$$\mathrm{QIC}(\Pi', \mu_z) = O\left(\sqrt{n} \cdot \max_w \mathrm{QIC}(\Pi, \mu_w) \log\left(2 + \frac{n^6}{\max_w \mathrm{QIC}(\Pi, \mu_w)}\right)\right). \qquad \blacktriangleleft$$

▶ **Lemma 25.** *Let G be a gadget, let μ_0 and μ_1 be a 0-distribution and a 1-distribution for G, and let Π be a protocol computing G (to bounded error against worst-case inputs). Then for any $n \in \mathbb{N}^+$, there is a protocol Π' computing $OR_n \circ G$ (to bounded error against worst-case inputs) such that*

$$\max_{z \in \{0,1\}^n} \mathrm{QIC}\left(\Pi', \mu_z\right) = O(n\, \mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1)).$$

Proof. In order to compute $\mathrm{OR}_n \circ G$, the protocol Π' will simply compute each copy of G one at a time, stopping as soon as a 1 has been found. The idea is that this will ensure the number of computations of 0-inputs to G is at most $O(n)$ while the number of computations of 1-inputs to G is $\tilde{O}(1)$.

To be more formal, we consider a cleaned up version $\hat{\Pi}$ of Π, which will have error $O(1/n)$ and which cleans up all the garbage and resets Alice and Bob's states to their initial states after the computation is complete. The protocol Π' will run $\hat{\Pi}$ on each input to G, in sequence, stopping when an output 1 has been found. To implement this, we will name the registers: suppose the protocol $\hat{\Pi}$ uses registers A and O_A on Alice's side and registers B and O_B on Bob's side, where O_A and O_B store the final output of $\hat{\Pi}$. At the beginning of $\hat{\Pi}$, the registers are expected to be $|0\rangle_A |0\rangle_B |0\rangle_{O_A} |0\rangle_{O_B}$. The guarantee of $\hat{\Pi}$ is that at the end of the algorithm, the registers will be in the state $|0\rangle_A |0\rangle_B |b\rangle_{O_A} |b\rangle_{O_B}$, where b is close to the output of G on that input. We now implement Π' by adding an additional register on each side, denoted S_A and S_B, which stores the strings of outputs of all the runs of $\hat{\Pi}$. These registers are each initialized to 0^n. At the end of run i of $\hat{\Pi}$ (which computes gadget i), Alice and Bob will each swap the register O_A with the i-th bit of S_A; this resets the registers used by $\hat{\Pi}$ to be all zero, and it stores the output of the i-th run of $\hat{\Pi}$ so that Π' has access to it. It also preserves the property that $S_A = S_B$ throughout the algorithm.

The final detail is that in Π', Alice and Bob only run $\hat{\Pi}$ on gadget i if they see that all the previous runs resulted in output 0; that is, they control the implementation of $\hat{\Pi}$ on the registers S_A and S_B being equal to 0^n. This will ensure that once a 1 is found, no further information will be exchanged between Alice and Bob. The final output of Π' will be 0 if S_A and S_B are 0^n, and it will be 1 otherwise.

The correctness of Π' (to worst-case bounded error) is clear, so we analyze its information cost against μ_z for a fixed string $z \in \{0,1\}^n$. The information cost $\mathrm{QIC}(\Pi', \mu_z)$ is a sum of information exchanged over all rounds; let $\mathrm{QIC}_i(\Pi', \mu_z)$ denote the sum of information exchanged only in the rounds corresponding to the computation of the i-th copy of G, so that $\mathrm{QIC}(\Pi', \mu_z) = \sum_{i=1}^n \mathrm{QIC}_i(\Pi', \mu_z)$.

Let T be the number of rounds used by $\hat{\Pi}$. Let $S_{A,i}$ and $S_{B,i}$ be the registers S_A and S_B during the computation of the i-th copy of G. Use X and Y to denote Alice and Bob's inputs respectively, with X_i and Y_i being the inputs to copy i of G and with $\tilde{X}$ and $\tilde{Y}$ denoting their purifications, and let C be the register passed back and forth between Alice and Bob in $\hat{\Pi}$. Then

$$\mathrm{QIC}_i(\Pi', \mu_z) = \sum_{t \le T \text{ odd}} I(\widetilde{X}\widetilde{Y} : C_t | Y B_t S_{B,i}) + \sum_{t \le T \text{ even}} I(\widetilde{X}\widetilde{Y} : C_t | X A_t S_{A,i}).$$

We note that the register $S_{B,i}$ in the odd terms is classical, as is the register $S_{A,i}$. Hence the conditional mutual information conditioned on $S_{B,i}$ is the expectation of the conditional mutual information conditioned on the events $S_{B,i} = w$ for each string $w \in \{0,1\}^n$ (see, for example, [12], end of Section 3.1). In other words,

$$\begin{aligned} & I(\widetilde{X}\widetilde{Y} : C_t | Y B_t S_{B,i}) \\ & = \Pr[S_{B,i} = 0^n] I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i}=0^n} + \Pr[S_{B,i} \neq 0^n] I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i} \neq 0^n}. \end{aligned}$$

Note that by the construction of Π', in the second term we have $I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i}=\neq 0^n} = 0$, since the registers C_t are all 0 as Alice and Bob do not run $\hat{\Pi}$ at all when $S_{B,i} \neq 0^n$. The term $I(\widetilde{X}\widetilde{Y} : C_t | Y B_t)_{S_{B,i}=0^n}$ is just $I(\widetilde{X_i}\widetilde{Y_i} : C_t | Y_i B_t)$, since the run of $\hat{\Pi}$ ignores everything outside of the input to the i-th copy of G. Hence we have

$$\begin{aligned} & \mathrm{QIC}_i(\Pi', \mu_z) \\ & = \Pr[S_{B,i} = 0^n] \sum_{t \le T \text{ odd}} I(\widetilde{X_i}\widetilde{Y_i} : C_t | Y_i B_t) + \Pr[S_{A,i} = 0^n] \sum_{t \le T \text{ even}} I(\widetilde{X_i}\widetilde{Y_i} : C_t | X_i A_t) \\ & = \Pr[S_{A,i} = 0] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i}). \end{aligned}$$

From this, it follows that

$$\mathrm{QIC}(\Pi', \mu_z) = \sum_{i=1}^n \Pr[S_{A,i} = 0^n] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i}).$$

To upper bound this, we note that the total sum of all the terms $\Pr[S_{A,i} = 0^n] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i})$ for i such that $z_i = 0$ is at most $n \, \mathrm{QIC}(\hat{\Pi}, \mu_0)$, where we've upper bounded $\Pr[S_{A,i} = 0^n] \le 1$. For i such that $z_i = 1$, we split into two cases: in the case where i is the first index such that $z_i = 1$, we upper bound $\Pr[S_{A,i} = 0^n] \, \mathrm{QIC}(\hat{\Pi}, \mu_{z_i}) \le \mathrm{QIC}(\hat{\Pi}, \mu_1)$. In contrast, for all i such that $z_i = 1$ and for which there was a previous index $j < i$ with $z_j = 1$, we note that the $1/n$ error guarantee of $\hat{\Pi}$ ensures that $\Pr[S_{A,i} = 0^n] \le 1/n$; hence these terms are individually at most $(1/n) \, \mathrm{QIC}(\hat{\Pi}, \mu_1)$, and the sum of all of them is at most $\mathrm{QIC}(\hat{\Pi}, \mu_1)$. We conclude that

$$\mathrm{QIC}(\Pi', \mu_z) \le n \, \mathrm{QIC}(\hat{\Pi}, \mu_0) + 2 \, \mathrm{QIC}(\hat{\Pi}, \mu_1).$$

Finally, we note that $\hat{\Pi}$ simply repeats Π $O(\log n)$ times and takes a majority votes in order to amplify (and then runs this in reverse to clean up garbage). Hence we have

$$\begin{aligned}
&\mathrm{QIC}(\hat{\Pi}, \mu_0) = O(\log n \cdot \mathrm{QIC}(\Pi, \mu_0)),\\
&\mathrm{QIC}(\hat{\Pi}, \mu_1) = O(\log n \cdot \mathrm{QIC}(\Pi, \mu_1)).
\end{aligned}$$

This gives the upper bound on $\mathrm{QIC}(\Pi', \mu_z)$ of $O(n \log n \cdot \mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1))$.

Finally, we sketch how to shave the log factor from the $\mathrm{QIC}(\Pi, \mu_0)$ term. To do so, we avoid amplifying $\hat{\Pi}$. Instead, we simply run $\hat{\Pi}$ on each input. If the output is 1, we run $\hat{\Pi}$ again on the same copy of G. We do so until the number of 0 outputs outnumbers the number of 1 outputs. If $O(\log n)$ repetitions happened and the number of 1-outputs is still larger than the number of 0 inputs, we finally "believe" that this gadget evaluates to 1 and halt. Otherwise, if the 0s outnumber the 1s before that point, then we assume the gadget evaluated to 0 and move on to the next one.

By analyzing this as the "monkey on a cliff" problem, it is not hard to see that a 1 gadget is correctly labelled as such with constant probability. The total number of runs of $\hat{\Pi}$ on 0-inputs will, on expectation, be at most $O(n)$, while the total number of runs of $\hat{\Pi}$ on 1-inputs will be at most $O(\log n)$ on expectation; since we avoided the $O(\log n)$ loss from amplification, this protocol is more efficient, and we shave a log factor from the $\mathrm{QIC}(\Pi, \mu_0)$ dependence.[8] ◀

We are now ready to prove Theorem 21.

Proof. (of Theorem 21.) Using Lemma 25, we get a protocol Π_2 computing $\mathrm{OR}_n \circ G$ such that for any $z \in \{0,1\}^n$, $\mathrm{QIC}(\Pi_2, \mu_z) = O(n\,\mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1))$. Using Lemma 24, we get a protocol Π_3 computing $G^{\oplus n}$ such that for any $z \in \{0,1\}^n$,

$$\begin{aligned}
&\mathrm{QIC}(\Pi_3, \mu_z)\\
&= O\Big((n^{3/2} \cdot \mathrm{QIC}(\Pi, \mu_0) + \sqrt{n} \log n \cdot \mathrm{QIC}(\Pi, \mu_1))\cdot\\
&\quad \log\left(2 + \frac{n^6}{n\,\mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1)}\right)\Big).
\end{aligned}$$

Finally, using Lemma 22, we get a protocol Π_4 computing G such that

$$\begin{aligned}
&\mathrm{QIC}(\Pi_4, \mu_0) + \mathrm{QIC}(\Pi_4, \mu_1)\\
&= O\left(\left(\sqrt{n} \cdot \mathrm{QIC}(\Pi, \mu_0) + \frac{\log n}{\sqrt{n}} \cdot \mathrm{QIC}(\Pi, \mu_1)\right)\cdot\right.\\
&\quad \left.\log\left(2 + \frac{n^6}{n\,\mathrm{QIC}(\Pi, \mu_0) + \log n \cdot \mathrm{QIC}(\Pi, \mu_1)}\right)\right).
\end{aligned}$$

Moreover, by negating the output of G, such a protocol Π_4 also exists with the μ_0 and μ_1 reversed.

Now, assume without loss of generality that $\mathrm{QIC}(\Pi, \mu_0) \le \mathrm{QIC}(\Pi, \mu_1)$. Let ℓ be the ratio $\mathrm{QIC}(\Pi, \mu_1)/\mathrm{QIC}(\Pi, \mu_0) \ge 1$ (here we use the assumption that $\mathrm{QIC}(\Pi, \mu_0) > 0$ and that $\mathrm{QIC}(\Pi, \mu_1) > 0$). Let $n \in \mathbb{N}^+$ be $\lceil 2\ell \log 2\ell \rceil$. Note that n is at most $3\ell \log 2\ell$, so $n = \Theta(\ell \log 2\ell)$ and $\log n = \Theta(\log 2\ell)$. Using this value of n, we get Π' such that

[8] We thank Thomas Watson and Mika Göös for pointing out this "monkey on a cliff" strategy for computing OR on a noisy oracle.

$$\mathrm{QIC}(\Pi', \mu_0) + \mathrm{QIC}(\Pi', \mu_1)$$
$$= O\left(\sqrt{\mathrm{QIC}(\Pi, \mu_0)\,\mathrm{QIC}(\Pi, \mu_1)}\log^{1/2}\frac{\mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1)}{\sqrt{\mathrm{QIC}(\Pi, \mu_0), \mathrm{QIC}(\Pi, \mu_1)}}\cdot\log(2+\alpha)\right),$$

where

$$\alpha = \frac{(\mathrm{QIC}(\Pi, \mu_0) + \mathrm{QIC}(\Pi, \mu_1))^{11}}{\mathrm{QIC}(\Pi, \mu_0)^6\,\mathrm{QIC}(\Pi, \mu_1)^6}.$$

If μ_0 and μ_1 are nontrivial, so that we have (say) $\mathrm{QIC}(\Pi, \mu_0) > 1/r^{10}$ and $\mathrm{QIC}(\Pi, \mu_1) > 1/r^{10}$, this can be simplified to

$$\mathrm{QIC}(\Pi', \mu_0) + \mathrm{QIC}(\Pi, \mu_1) = O(\sqrt{\mathrm{QIC}(\Pi, \mu_0)\,\mathrm{QIC}(\Pi, \mu_1)}\log^{3/2} r).$$

Finally, since n is at most polynomial in r, it is not hard to check that each of these reductions increases the number of rounds by only a polynomial factor in r, so the final protocol Π' has number of rounds $\mathrm{poly}(r)$. ◀

5.3 Proving the lifting theorem

▶ **Theorem 17.** *Let $f\colon \{0,1\}^n \to \Sigma$ be a relation (where $n \in \mathbb{N}^+$ and Σ is a finite alphabet) and let $G\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a communication function which contains both AND_2 and OR_2 as subfunctions. Then*

$$\mathrm{QCC}(f \circ G) = \tilde{\Omega}(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)).$$

Proof. Let μ_0' and μ_1' be the distributions for G provided by Theorem 19. Let μ_0 be the equal mixture of μ_0' and the uniform distribution over 0-inputs to the AND_2 gadget inside of G, and let μ_1 be the equal mixture of μ_1' and the uniform distribution over the 1-inputs to the OR_2 gadget inside of G. We note that for any protocol Π, $\mathrm{QIC}(\Pi, \mu_0) \geq \mathrm{QIC}(\Pi, \mu_0')/2$ and $\mathrm{QIC}(\Pi, \mu_1) \geq \mathrm{QIC}(\Pi, \mu_1')/2$. By [12], if Π has r rounds, $\mathrm{QIC}(\Pi, \mu_0), \mathrm{QIC}(\Pi, \mu_1) = \Omega(1/r)$. So μ_0 and μ_1 are nontrivial for G.

Let Π be a protocol computing $f \circ G$ to error ϵ, and let r be the number of rounds used by Π, and let T be the communication cost of Π. For $z \in \{0,1\}^n$, we define

$$q'(z,i) := \sum_{t \text{ odd}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{<i}\widetilde{Y}_{<i}B_t')_{\mu_z} + \sum_{t \text{ even}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{>i}\widetilde{Y}_{>i}A_t')_{\mu_z}$$

where C_t is the message in the t-th round of Π and A_t', B_t' are Alice and Bob's memory registers (which don't necessarily have safe copies of their inputs). By the chain rule of mutual information, we have

$$\sum_{i=1}^{n} q'(z,i) = \mathrm{QIC}(\Pi, \mu_z) \leq T$$

for all $z \in \{0,1\}^n$. A feasible weight scheme $q(z,i)$ for $\mathrm{Adv}_1(f)$ will be defined by normalizing $q'(z,i)$.

Let $z, w \in \{0,1\}^n$ be such that $f(z)$ and $f(w)$ are disjoint, and such that their Hamming distance is 1. Let $i \in [n]$ be the bit on which they disagree, so that $z^i = w$ (where z^i denotes the string z with bit i flipped). Suppose without loss of generality that $z_i = 1$ and $w_i = 0$. In order to lower bound $q'(z,i) \cdot q'(w,i)$, we will use the protocol Π for $f \circ G$ to construct a protocol Π' for G.

The protocol Π' is given by [41] (Lemma 4). Alice and Bob start with the shared entangled state of Π, as well the $\widetilde{X}_{-i}X_{-i}\widetilde{Y}_{-i}Y_{-i}$ registers of their inputs and purification according to $\mu_{z_{-i}}$ $(=\mu_{w_{-i}})$ in Π, with Alice holding $A_0\widetilde{X}_{<i}\widetilde{Y}_{<i}X_{-i}$ and Bob holding $B_0\widetilde{X}_{>i}\widetilde{Y}_{>i}Y_{-i}$ (here X_{-i} denotes $X_1 \dots X_{i-1}X_{i+1} \dots X_n$ and the same is true for other variables). They will embed their inputs for Π', which we call X', Y' (with purifications $\widetilde{X}'\widetilde{Y}'$), into the i-th input register for Π (with $\widetilde{X}', \widetilde{Y}'$ being embedded as $\widetilde{X}_i, \widetilde{Y}_i$), and use their shared entanglement for the rest of the input registers, to run Π. After running Π, they will output 1 if Π outputs a symbol in $f(z)$ (outputting 0 otherwise). Note that since Π outputs a symbol in $f(z)$ with probability at least $1-\epsilon$ when given an input from $(G^{\oplus n})^{-1}(z)$ and with probability at most ϵ when given an input from $(G^{\oplus n})^{-1}(w)$ (since $f(w) \cap f(z) = \varnothing$), it follows that Π' succeeds to error ϵ on worst-case inputs to G.

We now analyze the information cost of Π'. Against the distribution μ_0,

$$\begin{aligned}\mathrm{QIC}(\Pi', \mu_0) &= \sum_{t \text{ odd}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{<i}\widetilde{Y}_{<i}B'_t)_{\mu_{w_{-i}} \otimes \mu_{w_i}} \\ &\quad + \sum_{t \text{ even}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{>i}\widetilde{Y}_{>i}A'_t)_{\mu_{w_{-i}} \otimes \mu_{w_i}} \\ &= \sum_{t \text{ odd}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{<i}\widetilde{Y}_{<i}B'_t)_{\mu_w} + \sum_{t \text{ even}} I(\widetilde{X}_i\widetilde{Y}_i : C_t | \widetilde{X}_{>i}\widetilde{Y}_{>i}A'_t)_{\mu_w} \\ &= q'(w,i).\end{aligned}$$

Similarly, $\mathrm{QIC}(\Pi', \mu_1) = q(z,i)$, so we have

$$\sqrt{q'(z,i)q'(w,i)} = \sqrt{\mathrm{QIC}(\Pi', \mu_0)\,\mathrm{QIC}(\Pi', \mu_1)}.$$

By Theorem 21, there is a protocol Π'' such that

$$\sqrt{q'(z,i)q'(w,i)} = \Omega\left(\frac{\mathrm{QIC}(\Pi'', \mu_0) + \mathrm{QIC}(\Pi'', \mu_1)}{\operatorname{polylog} r}\right).$$

By the choice of μ_0 and μ_1, we therefore have

$$\sqrt{q'(z,i)q'(w,i)} = \Omega(\mathrm{QICZ}(G)/\operatorname{polylog} r),$$

and hence by taking $q(z,i) = O(\operatorname{polylog} r/\mathrm{QICZ}(G)) \cdot q'(z,i)$, we get $q(z,i)q(w,i) \geq 1$. If we start with a protocol Π with number of rounds r at most $\mathrm{QCC}_\epsilon(f \circ G)$, we conclude

$$\mathrm{QCC}_\epsilon(f \circ G) = \tilde{\Omega}(\mathrm{Adv}_1(f)\,\mathrm{QICZ}(G)),$$

as desired. ◀

6 New query relations

In this section, we prove our new relationships in query complexity. We start by showing that $\mathrm{cfbs}(f)$ is equivalent to $\mathrm{CAdv}(f)$ for partial functions. To do so, we will first need the well-known dual form for the fractional block sensitivity at a specific input, $\mathrm{fbs}(x, f)$. This dual form can be derived by writing the weight scheme defining $\mathrm{fbs}(x, f)$ as a linear program, and taking the dual; this gives a minimization program in which $\mathrm{fbs}(x, f)$ is the minimum, over weight schemes $q(i) \geq 0$ assigned to each $i \in [n]$ that satisfy $\sum_{i \in B} q(i) \geq 1$ for each sensitive block $B \subseteq [n]$ of x (with respect to f), of the sum $\sum_{i \in [n]} q(i)$. See any of [2, 40, 28] for a formal proof.

▶ **Lemma 26.** $\mathrm{cfbs}(f) \le 2\,\mathrm{CAdv}(f)$.

Proof. Let $q(x,i)$ be a feasible weight scheme for $\mathrm{CAdv}(f)$ with objective value equal to $\mathrm{CAdv}(f)$. We construct a completion f' of f as follows. For each $z \notin \mathrm{Dom}(f)$, let $z' \in \mathrm{Dom}(f)$ be the input in the domain of f which minimizes $\sum_{i:z'_i \ne z_i} q(z',i)$. Set $f'(z) = f(z')$. Now let x be any input in $\mathrm{Dom}(f)$; we wish to upper bound $\mathrm{fbs}(x,f')$.

To this end, we pick weights $q(i) = 2q(x,i)$, and claim that they are a feasible solution to the fractional block sensitivity for f' at x. Let B be any sensitive block for x with respect to f'. Then x^B is some input z which disagrees with x exactly on the bits in B, and which satisfies $f'(z) \ne f(x)$. Let z' be the input in $\mathrm{Dom}(f)$ which minimizes $\sum_{i:z'_i \ne z_i} q(z',i)$, so that $f'(z) = f(z')$. Then $f(z') \ne f(x)$, and in fact z' must be closer to z than to x; hence

$$\begin{aligned}\sum_{i\in B} q(i) &= \sum_{i:x_i\ne z_i} 2q(x,i)\\ &\ge \sum_{i:x_i\ne z_i} q(x,i) + \sum_{i:z'_i\ne z_i} q(z',i)\\ &\ge \sum_{i:x_i\ne z_i} \min\{q(x,i),q(z',i)\} + \sum_{i:z'_i\ne z_i} \min\{q(x,i),q(z',i)\}\\ &\ge \sum_{i:x_i\ne z'_i} \min\{q(x,i),q(z',i)\} \ge 1.\end{aligned}$$

We conclude that $q(i)$ is feasible. Its objective value is $\sum_{i\in[n]} q(i) = \sum_{i\in[n]} 2q(x,i) \le 2\,\mathrm{CAdv}(f)$, and hence $\mathrm{cfbs}(f) \le 2\,\mathrm{CAdv}(f)$, as desired. ◀

▶ **Lemma 27** (Krišjānis Prūsis, personal communication)**.** $\mathrm{CAdv}(f) \le \mathrm{cfbs}(f)$.

Proof. Let f' be a completion of f for which $\mathrm{fbs}(x,f') \le \mathrm{cfbs}(f)$ for all $x \in \mathrm{Dom}(f)$. For each $x \in \mathrm{Dom}(f)$, let $q_x(i)$ be a feasible weight scheme for the minimization problem of $\mathrm{fbs}(x,f')$ which satisfies $\sum_{i\in[n]} q_x(i) \le \mathrm{fbs}(x,f') \le \mathrm{cfbs}(f)$ and for each sensitive block B of f', $\sum_{i\in B} q_x(i) \ge 1$.

We construct a weight scheme for $\mathrm{CAdv}(f)$ by setting $q(x,i) = q_x(i)$ for all $x \in \mathrm{Dom}(f)$. We claim this weight scheme is feasible. To see this, let $x,y \in \mathrm{Dom}(f)$ be such that $f(x) \ne f(y)$. Define the input $z \in \{0,1\}^n$ such that $z_i = x_i$ if $x_i = y_i$, and otherwise, $z_i = x_i$ if $q(x,i) \ge q(y,i)$ and $z_i = y_i$ if $q(y,i) > q(x,i)$. Suppose that $f'(z) \ne f(x)$. Then

$$\begin{aligned}\sum_{i:x_i\ne y_i} \min\{q(x,i),q(y,i)\} &= \sum_{i:x_i\ne z_i} \min\{q(x,i),q(y,i)\} + \sum_{i:y_i\ne z_i} \min\{q(x,i),q(y,i)\}\\ &\ge \sum_{i:x_i\ne z_i} q(x,i) + \sum_{i:y_i\ne z_i} q(y,i) \ge 1.\end{aligned}$$ ◀

▶ **Lemma 28.** *For any (possibly partial) Boolean function f, we have*

$$\widetilde{\deg}_\epsilon(f) \ge \frac{\sqrt{2}}{\pi}\sqrt{(1-2\epsilon)\,\mathrm{fbs}(f)}.$$

Proof. Let $x \in \mathrm{Dom}(f)$ be such that $\mathrm{fbs}(x,f) = \mathrm{fbs}(f)$. By negating the input bits of f if necessary, we may assume that $x = 0^n$ (note that negating input bits does not affect $\mathrm{fbs}(f)$ or $\widetilde{\deg}(f)$). By negating the output of f if necessary, we can further assume that $f(0^n) = 0$. Let p be a polynomial of degree $\widetilde{\deg}_\epsilon(f)$ which approximates f to error ϵ.

Let PrOR_k be the promise problem on k bits whose domain contains all the strings of Hamming weights 0 or 1, and which outputs 0 given 0^k and outputs 1 given an input of Hamming weight 1.

We give an exact polynomial representation of this function. To do so, let T_d be the Chebyshev polynomial of degree d; this is the single-variate real polynomial satisfying $T_d(\cos\theta) = \cos(d\theta)$. This polynomial is bounded in $[-1,1]$ on the interval $[-1,1]$. Moreover, it satisfies $T_d(1) = 1$ and $T_d(\cos(\pi/d)) = -1$. Hence the polynomial $r(t) = (1 - T_d(1 - (1 - \cos(\pi/d))t))/2$ evaluates to 0 at $t = 0$ and to 1 at $t = 1$. Moreover, since this T_d is bounded in $[-1,1]$ on the interval $[-1,1]$, we conclude that $r(t)$ is bounded in $[0,1]$ on the interval $[0, 2/(1-\cos(\pi/d))]$. Since $\cos(z) \ge 1 - z^2/2$, we have $2/(1-\cos(\pi/d)) \ge 4d^2/\pi^2$. Hence $r(t)$ is bounded in $[0,1]$ on the interval $[0, 4d^2/\pi^2]$. If we pick d such that $4d^2/\pi^2 \ge k$, that is, d at least $\lceil \pi\sqrt{k}/2 \rceil$, then we would know that $r(t)$ is bounded on $[0,k]$. In that case, the k-variate polynomial $q(x) = r(x_1 + x_2 + \cdots + x_k)$ would exactly compute $\textsc{PrOR}_k$, and it would have degree at most $\lceil \pi\sqrt{k}/2 \rceil \le \pi\sqrt{k}/2 + 1$.

Next, consider the function $f \circ \textsc{PrOR}_k$. We can approximate this function to error ϵ simply by plugging in n independent copies of the polynomial q into the variables of the polynomial p. This means that the approximate degree of $f \circ \textsc{PrOR}_k$ to error ϵ is at most $\widetilde{\deg}_\epsilon(f) \cdot (\pi\sqrt{k}/2 + 1)$.

On the other hand, we now claim that for appropriate choice of k, we have $\mathrm{bs}(0^{kn}, f \circ \textsc{PrOR}_k) \ge k\,\mathrm{fbs}(0^n, f)$, and hence $\mathrm{bs}(f \circ \textsc{PrOR}_k) \ge k\,\mathrm{fbs}(f)$. To see this, let $\{w_B\}_B$ be an optimal weight scheme for the fractional block sensitivity of 0^n with respect to f, so that $\sum_{B:i\in B} w_B \le 1$ and $\sum_B w_B = \mathrm{fbs}(f)$. Note that since fractional block sensitivity is a linear program, the optimal solution can be taken to be rational; let L be a common denominator of all the weights, so that Lw_B is an integer for each sensitive block B. Now take k to be an integer multiple of L. For each sensitive block B of 0^n with respect to f, we define kw_B different sensitive blocks of 0^{kn} with respect to $f \circ \textsc{PrOR}_k$, such that all of the new blocks are mutually disjoint. To do so, we simply use a different bit in each copy of $\textsc{PrOR}_k$ for each block. Since the sum of weights w_B for blocks that use bit i of the input to f is at most 1, the total number of new blocks we will generate that use copy i of $\textsc{PrOR}_k$ is at most k, and hence we can give each block a different bit of that copy of $\textsc{PrOR}_k$. The total number of disjoint blocks will then be $k\sum_B w_B = k\,\mathrm{fbs}(f)$.

We conclude that $\mathrm{bs}(f \circ \textsc{PrOR}_k) \ge k\,\mathrm{fbs}(f)$ as long as k is a multiple of a certain integer L. Now, by a standard result [9, 15], we know that the approximate degree to error ϵ of a (possibly partial) Boolean function is at least the square root of its block sensitivity; more explicitly, we have

$$\widetilde{\deg}_\epsilon(f \circ \textsc{PrOR}_k) \ge \sqrt{\frac{1-2\epsilon}{2(1-\epsilon)}\,\mathrm{bs}(f \circ \textsc{PrOR}_k)} \ge \sqrt{\frac{1-2\epsilon}{2(1-\epsilon)} k\,\mathrm{fbs}(f)}.$$

Combined with our upper bound on this degree, we have

$$\widetilde{\deg}_\epsilon(f) \cdot (\pi\sqrt{k}/2 + 1) \ge \sqrt{\frac{1-2\epsilon}{2(1-\epsilon)} k\,\mathrm{fbs}(f)},$$

and since k can go to infinity, we must have

$$\widetilde{\deg}_\epsilon(f) \ge \frac{\sqrt{2}}{\pi}\sqrt{\frac{(1-2\epsilon)}{1-\epsilon}\,\mathrm{fbs}(f)},$$

from which the desired result follows. ◀

▶ **Theorem 29.** *For all (possibly partial) Boolean functions f, we have*

$$\widetilde{\deg}_\epsilon(f) \ge \frac{\sqrt{(1-2\epsilon)\,\mathrm{cfbs}(f)}}{\pi}.$$

Proof. Let p be a polynomial which approximates f to error ϵ. Then $p(x) \in [0,1]$ for all $x \in \{0,1\}^n$, so define $f'(x)$ by $f'(x) = 1$ if $p(x) \geq 1/2$ and $f'(x) = 0$ if $p(x) < 1/2$. It is clear that $f'(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$, so f' is a completion of f. Let $x \in \mathrm{Dom}(f)$ be an input so that $\mathrm{fbs}(x, f') \geq \mathrm{cfbs}(f)$. To complete the proof, it will suffice to lower bound the degree of p by $\Omega(\sqrt{\mathrm{fbs}(x, f')})$.

Suppose without loss of generality that $f(x) = 0$ (otherwise, negate f and f' and replace p with $1 - p$). Then we know that $p(x) \in [0, \epsilon]$, and that for any $y \in \{0,1\}^n$ such that $f'(x) \neq f'(y)$, we have $p(y) \in [1/2, 1]$. This means that the polynomial $q(z) = (2p(z) + 1 - 2\epsilon)/(3 - 2\epsilon)$ has the same degree as p, is bounded in $[0,1]$ on $\{0,1\}^n$, and approximates f' to error $1/(3 - 2\epsilon)$ on the input x and on all inputs $y \in \{0,1\}^n$ such that $f'(x) \neq f'(y)$. In other words, consider the partial function f'_x which is the restriction of f' to the promise set $\{x\} \cup \{y \in \{0,1\}^n : f'(y) \neq f'(x)\}$. Then q approximates f'_x to error $1/(3 - 2\epsilon)$, and has the same degree as p. Now, it is not hard to see that $\mathrm{fbs}(f'_x) = \mathrm{fbs}(x, f')$. Hence it suffices to lower bound the degree of q by $\Omega(\sqrt{\mathrm{fbs}(f'_x)})$. Such a lower bound follows from Lemma 28; indeed, we conclude that the degree of p is at least

$$\frac{1}{\pi}\sqrt{\frac{1-2\epsilon}{1-\epsilon}\,\mathrm{cfbs}(f)}. \qquad ◀$$

▶ **Theorem 30.** *For all (possibly partial) Boolean functions f,*

$$\mathrm{CAdv}(f) \leq 2\,\mathrm{Adv}(f)^2.$$

Proof. Let f be a (possibly partial) Boolean function, and $q(x,i)$ be a feasible weight scheme for $\mathrm{Adv}(f)$ that has $\sum_{i \in [n]} q(x,i) \leq \mathrm{Adv}(f)$ for all i. Fix any $x, y \in \mathrm{Dom}(f)$ such that $f(x) \neq f(y)$. Then

$$1 \leq \sum_{i : x_i \neq y_i} \sqrt{q(x,i)q(y,i)} = \sum_{i : x_i \neq y_i} \sqrt{\min\{q(x,i), q(y,i)\} \max\{q(x,i), q(y,i)\}}$$

$$\leq \sqrt{\sum_{i : x_i \neq y_i} \min\{q(x,i), q(y,i)\} \cdot \sum_{i : x_i \neq y_i} \max\{q(x,i), q(y,i)\}}.$$

Note that $\max\{q(x,i), q(y,i)\} \leq q(x,i) + q(y,i)$, and we know the sum over i of $q(x,i)$ and $q(y,i)$ are each at most $\mathrm{Adv}(f)$. Hence we get

$$\sum_{i : x_i \neq y_i} \max\{q(x,i), q(y,i)\} \leq 2\,\mathrm{Adv}(f),$$

and hence

$$\sum_{i : x_i \neq y_i} \min\{q(x,i), q(y,i)\} \geq \frac{1}{2\,\mathrm{Adv}(f)}.$$

This means that if we scale the weights $q(x,i)$ up by a uniform factor of $2\,\mathrm{Adv}(f)$, the resulting weight scheme $q'(x,i)$ will be feasible for $\mathrm{CAdv}(f)$. The objective value of this new weight scheme will then be the maximum over x of

$$\sum_{i \in [n]} q'(x,i) = 2\,\mathrm{Adv}(f) \sum_{i \in [n]} q(x,i) \leq 2\,\mathrm{Adv}(f)^2,$$

so $\mathrm{CAdv}(f) \leq 2\,\mathrm{Adv}(f)^2$, as desired. ◀

References

1 Scott Aaronson. Lower bounds for local search by quantum arguments. *SIAM Journal on Computing*, 35(4):804–824, 2006. Previous version in STOC 2004. doi:10.1137/s0097539704447237.

2 Scott Aaronson. Quantum certificate complexity. *Journal of Computer and System Sciences*, 74(3):313–322, 2008. Previous version in CCC 2003. doi:10.1016/j.jcss.2007.06.020.

3 Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of huang's sensitivity theorem, 2020. Preprint,. arXiv:2010.12629.

4 R Alicki and M Fannes. Continuity of quantum conditional information. *Journal of Physics A: Mathematical and General*, 37(5):L55–L57, 2004. doi:10.1088/0305-4470/37/5/l01.

5 Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Previous version in STOC 2000. doi:10.1006/jcss.2002.1826.

6 Andris Ambainis, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. All classical adversary methods are equivalent for total functions. In *Proceedings in the 35th Symposium on Theoretical Aspects of Computer Science (STACS)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2018. doi:10.4230/LIPICS.STACS.2018.8.

7 Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 209–218, 2002. doi:10.1109/SFCS.2002.1181944.

8 Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. Previous version in FOCS 2002. doi:10.1016/j.jcss.2003.11.006.

9 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Previous version in FOCS 1998. doi:10.1145/502090.502097.

10 Aleksandrs Belovs. Quantum algorithms for learning symmetric juntas via the adversary bound. *Computational Complexity*, 24(2):255–293, 2015. Previous version in CCC 2014. doi:10.1007/s00037-015-0099-2.

11 Shalev Ben-David, Adam Bouland, Ankit Garg, and Robin Kothari. Classical lower bounds from quantum upper bounds. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2018. doi:10.1109/focs.2018.00040.

12 Mark Braverman, Ankit Garg, Young Kun Ko, Jieming Mao, and Dave Touchette. Near-optimal bounds on the bounded-round quantum communication complexity of disjointness. *SIAM Journal on Computing*, 47(6):2277–2314, 2018. Previous version in FOCS 2015. doi:10.1137/16m1061400.

13 Mark Braverman and Omri Weinstein. An interactive information odometer and applications. In *Proceedings of the 47th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM Press, 2015. doi:10.1145/2746539.2746548.

14 Harry Buhrman, Matthias Christandl, and Christian Schaffner. Complete insecurity of quantum protocols for classical two-party computation. *Physical Review Letters*, 109(16), 2012. doi:10.1103/physrevlett.109.160501.

15 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

16 Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting for BPP using inner product. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2019. doi:10.4230/LIPICS.ICALP.2019.35.

17 Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Simulation theorems via pseudo-random properties. *Computational Complexity*, 28(4):617–659, 2019. doi:10.1007/s00037-019-00190-7.

18 Roger Colbeck. Impossibility of secure two-party classical computation. *Physical Review A*, 76(6), 2007. doi:10.1103/physreva.76.062308.

19 Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. Feasibility and completeness of cryptographic tasks in the quantum world. In *Proceedings of the 10th Theory of Cryptography Conference (TCC)*, pages 281–296. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-36594-2_16.

20 Justin Gilmer, Michael Saks, and Sudarshan Srinivasan. Composition limits and separating examples for some boolean function complexity measures. *Combinatorica*, 2016. Previous version in CCC 2013. doi:10.1007/s00493-014-3189-x.

21 Mika Göös. Lower bounds for clique vs. independent set. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1066–1076. IEEE, 2015. doi:10.1109/FOCS.2015.69.

22 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016. Previous version in STOC 2015. doi:10.1137/15M103145X.

23 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM Journal on Computing*, 2018. Previous version in FOCS 2015. doi:10.1137/16M1059369.

24 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM Journal on Computing*, 47(5):1778–1806, 2018. Previous version in STOC 2014. doi:10.1137/16m1082007.

25 Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. *SIAM Journal on Computing*, 49(4):FOCS17–441–FOCS17–461, 2020. Previous version in FOCS 2017. doi:10.1137/17m115339x.

26 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 526–535, 2007. doi:10.1145/1250790.1250867.

27 Trinh Huynh and Jakob Nordstrom. On the virtue of succinct proofs. In *Proceedings of the 44th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM Press, 2012. doi:10.1145/2213977.2214000.

28 Raghav Kulkarni and Avishay Tal. On fractional block sensitivity. *Chicago Journal of Theoretical Computer Science*, 2016. doi:10.4086/cjtcs.2016.008.

29 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996. doi:10.1017/cbo9780511574948.

30 Sophie Laplante and Frédéric Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. *SIAM Journal on Computing*, 2008. Previous version in CCC 2004. doi:10.1137/050639090.

31 Mathieu Laurière and Dave Touchette. The flow of information in interactive quantum protocols: the cost of forgetting. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2017. doi:10.4230/LIPICS.ITCS.2017.47.

32 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 344–353, 2011. doi:10.1109/FOCS.2011.75.

33 Hoi-Kwong Lo. Insecurity of quantum secure computations. *Physical Review A*, 56(2):1154–1162, 1997. doi:10.1103/physreva.56.1154.

34 Noam Nisan. Crew prams and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991. Previous version in STOC 1989. doi:10.1137/0220062.

35 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–569. SIAM, 2011. doi:10.1137/1.9781611973082.44.

36 Louis Salvail, Christian Schaffner, and Miroslava Sotáková. Quantifying the leakage of quantum protocols for classical two-party cryptography. *International Journal of Quantum Information*, 13(04):1450041, 2014. doi:10.1142/s0219749914500415.

37 Alexander A. Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969–2000, 2011. Previous version in STOC 2008. doi:10.1137/080733644.

38 Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958. doi:10.2140/pjm.1958.8.171.

39 Robert Špalek and Mario Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2, 2006. Previous version in ICALP 2005. doi:10.4086/toc.2006.v002a001.

40 Avishay Tal. Properties and applications of boolean function composition. In *Proceedings of the 4th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 441–454, 2013. doi:10.1145/2422436.2422485.

41 Dave Touchette. Quantum information complexity. In *Proceedings of the 47th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM Press, 2015. doi:10.1145/2746539.2746613.

42 Xiaodi Wu, Penghui Yao, and Henry Yuen. Raz-mckenzie simulation with the inner product gadget, 2017. Preprint. arXiv:2017/010.

Hardness of Constant-Round Communication Complexity

Shuichi Hirahara ✉
National Institute of Informatics, Tokyo, Japan

Rahul Ilango ✉
Massachusetts Institute of Technology, Cambridge, MA, USA

Bruno Loff ✉
INESC-Tec and University of Porto, Portugal

Abstract

How difficult is it to compute the communication complexity of a two-argument total Boolean function $f : [N] \times [N] \rightarrow \{0, 1\}$, when it is given as an $N \times N$ binary matrix? In 2009, Kushilevitz and Weinreb showed that this problem is cryptographically hard, but it is still open whether it is NP-hard.

In this work, we show that it is NP-hard to approximate the size (number of leaves) of the smallest *constant-round* protocol for a two-argument total Boolean function $f : [N] \times [N] \rightarrow \{0, 1\}$, when it is given as an $N \times N$ binary matrix. Along the way to proving this, we show a new *deterministic* variant of the round elimination lemma, which may be of independent interest.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Problems, reductions and completeness

Keywords and phrases NP-completeness, Communication Complexity, Round Elimination Lemma, Meta-Complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.31

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2021/030/

Funding *Shuichi Hirahara*: This work was partly carried out during a visit supported by ACT-I, JST.
Rahul Ilango: During this work, this author was funded by an Akamai Presidential Fellowship and by NSF Grants CCF-1741615 and CCF-1909429.
Bruno Loff: This project was financed by the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020. This work was partly carried out during a research visit conducted with support from DIMACS in association with its Special Focus on Lower Bounds.

Acknowledgements The authors would like to thank Ryan Williams for his support, and for several discussions and suggestions, without which this paper would not have existed. The authors would also like to thank Igor Oliveira for helpful conversations about hardness of communication complexity.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 31; pp. 31:1–31:30
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Contents

1 Introduction

Suppose you are given a $N \times N$ Boolean matrix representing a (total) two-player communication problem. How difficult is it to determine the (deterministic) communication complexity of this matrix?

In 2009, Kushilevitz and Weinreb [39] studied this question and showed that, under a cryptographic assumption, no polynomial-time algorithm can compute the communication complexity of a given total two-player function. They left open the question of whether this problem is NP-hard.

Our main result is that the problem of determining the minimum number of leaves in an *d-round* communication protocol for a given (total, two player) function is NP-hard, for all integer constants $d \geq 3$.

1.1 Motivation

Determining the difficulty of computing communication complexity is an interesting, basic question in its own right. However, the aforementioned paper of Kushilevitz and Weinreb – which gave the first non-trivial results on this problem – was also motivated by the broader implications this question could have for communication complexity. This fits into an even broader motif that has become prominent in recent years: using "meta-questions" to investigate various aspects of complexity theory.

For example, Kushilevitz and Weinreb argue that understanding the intractability of computing communication complexity can help "explain the difficulty of analyzing the communication complexity of certain functions." Towards this end, their cryptographic hardness result exhibits a family of functions whose communication complexity we are unlikely to ever gain a complete understanding of (since determining their communication complexity is crytographically hard).

Kushilevitz and Weinreb also used the meta-complexity lens to shed light on one of the oldest questions in communication complexity: the log-rank conjecture of Lovasz and Saks [41]. If the log-rank conjecture is true, then it yields a simple polynomial-time approximation algorithm for computing communication complexity (simply output the logarithm of the rank of the input matrix). A natural question is whether one can get a better approximation algorithm. Kushilevitz and Weinreb introduced a plausible conjecture that would imply that the log-rank conjecture, if true, yields an optimal polynomial-time approximation. On the other hand, a strong enough hardness of approximation result could actually disprove the log-rank conjecture (conditioned on P $\neq$ NP). Thus, understanding the inapproximability of computing communication complexity seems closely related to resolving the log-rank conjecture.

Finally, Kushilevitz and Weinreb's paper also introduced a remarkable new technique, showing for the first time how one could devise a total two-player Boolean function whose communication complexity was strongly tied to the Boolean-formula complexity of another, related function. Prior to their work, connections had only been known between Boolean-formula complexity and the communication complexity of *search* problems [35] and were not known for *decision* problems.

Thus, it is plausible that proving NP-hardness results for computing communication complexity could reveal further insights in communication complexity and lead to the development of useful new techniques. Indeed, our constant-round NP-hardness result led us to prove an interesting new direct-sum/round-elimination result in deterministic communication complexity, which we state in the following section.

1.2 Our Results

In order to state our results formally, we fix some notation. If $f : [a] \times [b] \to \{0, 1\}$ is a two-player Boolean-valued function, then

- $\mathsf{C}^{\mathrm{A}}_{d}(f)$ denotes its d-round deterministic communication complexity, namely, the smallest number of bits communicated in a d-round protocol that computes f, where Alice speaks in the first round,
- $\mathsf{L}^{\mathrm{A}}_{d}(f)$ denotes the minimum number of leaves in a d-round protocol that computes f where Alice speaks first,
- $\mathsf{C}^{\mathrm{B}}_{d}(f)$ and $\mathsf{L}^{\mathrm{B}}_{d}(f)$ denote the analogous notions where Bob speaks first, and
- $\mathsf{C}^{\mathrm{A}}_{d,\varepsilon}(f)$, $\mathsf{C}^{\mathrm{B}}_{d,\varepsilon}(f)$, $\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(f)$ and $\mathsf{L}^{\mathrm{B}}_{d,\varepsilon}(f)$ denote the analogous notions but where the protocol is probabilistic, and is allowed to err with probability $\leq \varepsilon$.

Our first result shows that computing 3-round deterministic communication complexity is NP-hard. We construct a reduction from the chromatic number problem to the problem of computing 3-round deterministic communication complexity. Our reduction attains the following hardness:

▶ **Theorem 1** (Informal version of Theorem 18)**.** *It is* NP*-hard to approximate* $\mathsf{L}^{\mathrm{A}}_{3}(f)$ *to within a factor of* $N^{1/8}$*, and* $\mathsf{C}^{\mathrm{A}}_{3}(f)$ *to within an additive term of* $\frac{1}{8}\log N$*,*[1] *when given a function* $f : [N] \times [N] \to \{0, 1\}$*.*

We then work to prove NP hardness for all constants $d \geq 3$ by induction on d, using Theorem 1 as a base case. Thus, in our inductive step, our goal is to show that computing d-round communication complexity reduces to computing $(d + 1)$-round communication complexity.

[1] Since $\mathsf{C}^{\mathrm{A}}_{3}(f) \leq \log N + 1$, this implies it is hard to approximate $\mathsf{C}^{\mathrm{A}}_{3}(f)$ to within a multiplicative factor of $1 + \frac{1}{8}$.

A natural approach would be to use the round elimination lemma [46, 62]. This lemma says that given a two-player function f, one can create a new function F such that the $(d+1)$-round communication complexity of F is closely related to the d-round communication complexity of f.

There are a few difficulties in using round elimination. For one, going from f to F in round elimination requires a dramatic blow up of the input size of the function. As a result, any reduction based on typical round elimination seems to require a superpolynomial running time.

A more significant issue is that round elimination only works for probabilistic protocols, not deterministic protocols. So, in order to use round elimination, we would actually need a much stronger version of Theorem 1 for our base case: that it is hard to distinguish protocols that have small three-round *deterministic* communication complexity from protocols that require large three-round *randomized* communication complexity. As it turns out, we can "almost" prove such a result (see Section 6):

▶ **Theorem 2.** *There exist positive constants γ and δ such that the following holds. There exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N \times N}$ and a number $k \in \mathbb{N}$, with $k \leq N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of* SAT*, then $\mathsf{L}^{\mathrm{B}}_3(M) \leq O(k)$ and $\mathsf{C}^{\mathrm{B}}_3(M) \leq \log k + O(1)$, and*
2. *if x is a NO instance of* SAT*, then $\mathsf{L}^{\mathrm{B}}_{3,N^{-\delta}}(M) \geq \Omega(N^{\gamma} \cdot k)$ and $\mathsf{C}^{\mathrm{B}}_{3,N^{-\delta}}(M) \geq \log k + \gamma \cdot \log N - O(1)$.*

Unfortunately, the hardness parameters we obtain are not enough to make the round-elimination approach work. If in the above theorem we could have chosen $\gamma \geq c \cdot \delta$ for an arbitrarily large constant c, then we would be able to use round elimination to show that $\mathsf{C}^{\mathrm{A}}_d(f)$ is NP-hard for any constant number of rounds d, under subexponential-time reductions (this is proven in Section 7). If we could make the error parameter constant instead of $N^{-\delta}$, then we would be able to show that $\mathsf{C}^{\mathrm{A}}_d(f)$ is NP-hard under quasipolynomial-time reductions. We leave proving a version of Theorem 2 with these stronger parameters as an open problem.

In light of these difficulties, a natural question is whether there exists an alternative to round elimination that works with deterministic protocols. Ideally, this alternative method would also avoid introducing a subexponential blowup. Towards this end, we prove a new result in deterministic communication complexity that gives us a tight relation between the minimum number of leaves in a d-round protocol for f, and the minimum number of leaves in a $(d+1)$-round protocol for a related function F. This function F is a kind of "direct sum" of f with the "XOR-equality" function. It should be remarked that the direct-sum property is known to *fail* for general deterministic protocols [56], so we cannot, for example, replace XOR-equality with another arbitrary function with the same communication complexity. The formal statement of our result is as follows:

▶ **Lemma 3.** *Let $d \geq 3$. Given an arbitrary two-player total Boolean function $f : [a] \times [b] \to \{0,1\}$, define the function $F : ([k] \times [a]) \times ([k] \times [b] \times \{0,1\} \times \{0,1\}) \to \{0,1\}$ given by*

$$F(x_0, x_1; y_0, y_1, z, i) = \begin{cases} \mathsf{XorEq}_k(x_0; y_0, z) & , \text{ if } i = 0 \\ f(x_1; y_1) & , \text{ if } i = 1, \end{cases}$$

where, in turn, $\mathsf{XorEq}_k : [k] \times ([k] \times \{0,1\}) \to \{0,1\}$ is given by

$$\mathsf{XorEq}_k(x; y, z) = \begin{cases} z & \text{if } x \neq y \\ 1 - z & \text{if } x = y. \end{cases}$$

Then

$$\min\{4k, 2k - 2 + \mathsf{L}^{\mathrm{B}}_d(f)\} \leq \mathsf{L}^{\mathrm{A}}_{d+1}(F) \leq 2k + \mathsf{L}^{\mathrm{B}}_d(f).$$

The last two inequalities can be seen as saying that $\mathsf{L}^{\mathrm{A}}_{d+1}(F) - 2k$ is a good approximation of $\mathsf{L}^{\mathrm{B}}_{d}(f)$, for $k \geq \mathsf{L}^{\mathrm{B}}_{d}(f)$. Thus, it is natural to view Lemma 3 not just as a direct-sum-type result, but also as a kind of round elimination lemma, since it relates the $(d+1)$-round communication complexity of F with the d-round communication complexity of f.

Lemma 3 has a few significant differences from the classical round elimination lemma. First, while the classical lemma only applies to probabilistic protocols, Lemma 3 works in the deterministic case.

Second, Lemma 3 is more efficient in the number of inputs of F relative to f. Using the classical round elimination lemma would require at least a quasipolynomial blowup in going from f to F, but a quadratic blowup suffices for Lemma 3. This allows us to build a polynomial-time NP-hardness reduction instead of the superpolynomial-time reduction that would follow from the randomized round-elimination approach.

Third, our proof of Lemma 3 looks very different than the classical proof of the round elimination lemma, which is almost entirely information-theoretic.[2] Our proof is instead more combinatorial and builds on a fooling set argument. We sketch the proof of Lemma 3 in Section 1.5.

Using Theorem 1 and Lemma 3, we obtain our main theorem:

▶ **Theorem 4.** *For any $d \geq 3$ there exists a constant $\Delta_d > 0$ such that the following holds. If there exists a polynomial-time algorithm which, when given a total two-player Boolean-valued function $f : [N] \times [N] \to \{0,1\}$ represented as a Boolean matrix of dimensions $N \times N$, approximates $\mathsf{L}^{\mathrm{A}}_{d}(f)$ within a factor of $1 + \Delta_d$, then* $\mathsf{P} = \mathsf{NP}$.

1.3 Meta-Complexity

Our work fits into a now well-established theme in computational complexity theory of studying "meta-complexity questions." Historically, this kind of question was first studied by Soviet cyberneticians beginning in the 1950s, who were particularly interested in the problem of circuit minimization: the ("meta-complexity") task of computing the smallest circuit for a prescribed Boolean function [63]. At the time, this was considered to be among the least likely computational task to have better-than-brute-force algorithms. Reportedly, Levin delayed the publication of his work on NP-completeness, because he was hoping to show the NP-hardness of this problem [12].

Since then, meta-complexity questions have become so pervasive, that there are few unsolved problems in computational complexity which are not touched by meta-complexity results. For example:

- The relativization barrier [14], and related algebrization barrier [1], imply that a number of proof techniques will be insufficient to settle most uniform complexity-class separation questions.
- The natural proofs barrier [60, 50] also immediately excludes us from considering many properties which might, at a first glance, plausibly imply hardness of a given Boolean function for circuit classes above TC_0.
- It is known that the complexity measures we are interested in understanding, such as the number of leaves in Boolean formulae, are inherently non-convex [34] and non-submodular [59], and thus cannot, for example, be approximated by convex programming or by certain rank-based measures.

[2] Indeed, using information theoretic techniques, like the chain rule and Pinsker's inequality, seems to require a superpolynomial blowup.

- Lower-bounds that mildly improve classic lower-bounds (e.g. super-linear lower-bounds against NC_1 [11], lower-bounds against one-pass streaming algorithms [44]) or lower-bounds for certain problems (e.g. k-vertex cover [54, 55]) against complexity classes for which we already have lower-bounds, could be "magnified" to solve longstanding open problems.
- Just recently, the existence of one-way functions is now known to be equivalent to the average-case hardness of computing polytime-bounded Kolmogorov complexity [40].

In this paper, we contribute to one of the lines of research inscribed in this theme. Generically, fixing a complexity measure $\mathcal{C}$, we can define a "meta"-problem $\mathsf{MP}_{\mathcal{C}}$ where a task T is the input to the problem $\mathsf{MP}_{\mathcal{C}}$, and the problem $\mathsf{MP}_{\mathcal{C}}$ is to compute $\mathcal{C}(T)$, namely, the $\mathcal{C}$-complexity of T. The "meta"-question is then: what is the computational complexity of $\mathsf{MP}_{\mathcal{C}}$?

1.3.1 Previous work

This meta-question has been studied in many previous works. Most of these works deal with the case where T is the truth table of a Boolean function and the complexity measure $\mathcal{C} = \mathsf{SIZE}$ is the size of the smallest circuit computing T; in this case $\mathsf{MP}_{\mathsf{SIZE}}$ is denoted MCSP, which stands for "Minimum Circuit-Size Problem." The question originally posed by Levin is whether MCSP is NP-complete, and this is the main unresolved question in this area.

We seem far from settling this question, but MCSP is known to be hard for various other classes [54, 22, 54, 5, 6]. It is also known that MCSP is *not* NP-hard under various weak reductions [48, 33, 48, 27, 26, 8, 10]. MCSP has many natural connections to other areas, such as cryptography [57, 61], natural proofs [61], hardness magnification [53, 45], learning [18], and proof complexity [37, 47]). A few variants of MCSP are known to be NP-hard, including some relativized versions of MCSP [9, 26, 31], a conditional version of MCSP [28], and MCSP for multi-output functions [30]. For more information on recent research, see Allender's recent survey [4] and the references therein.

Thus far, relatively few works have focused on proving the NP-hardness of computing other complexity measures. The NP-hardness of computing the size of the smallest DNF for a given function (given as a truth table) was first established by Masek, already in 1978 [43]. A series of subsequent works later improved this result to give near-optimal hardness-of-approximation [19, 64, 21, 7, 36]. More recently, it was shown to be NP-hard to compute the size of the smallest DNF-of-XORs [25]. Other works have established NP-hardness of the task of finding an optimal algorithm, such as finding the smallest decision tree for a given partial function [23], finding the weights of a neural network (with a fixed topology) that computes a given function [32, 16], or finding the smallest straight-line program for computing a given linear form [17].

A special reference should be made to the previous work by one of the authors [29], where it was shown that it is NP-hard to approximate the size of the smallest AC^0-formula of a total function given as a truth-table, with an approximation factor of $1 \pm \Delta_d$, where Δ_d depends on the depth d. Our paper is inspired by this earlier paper, and our work can be seen as proving an analogous result for constant-round protocols for total Boolean functions, to the result proven in [29] for constant-round protocols for Karchmer–Wigderson games. The high-level idea of the proof is similar, as well: we first prove hardness for constant depth, and then show how to reduce the depth-d problem to the depth-$(d+1)$ problem. However, the required techniques are completely different, both for establishing the base case and the inductive step. The depth 2 problem is already hard in the case of minimizing Boolean formulas, but it can be solved in polynomial time in the case of communication complexity. Hence, our first hard case is the depth 3 case. Our proof of the inductive depth-d to depth-$(d+1)$ reduction is also very different, and we further discuss the differences in Section 1.5.

1.4 Outline of the Paper

Our proofs take the approach of starting with simple cases and building up to more complicated cases. We begin in Section 3 by showing the NP-hardness of approximating the 3-round communication complexity of total, *multi-output* functions, i.e., total two-player functions $f : [a] \times [b] \to [\ell]$. This problem has a nice combinatorial interpretation (see Proposition 11), and it turns out to be NP-hard to approximate by a simple reduction from graph coloring. The proof is an interesting combination of an NP-hardness reduction with a communication-complexity lower-bound argument.

We then prove in Section 4 that 3-round communication complexity is also hard to approximate for *Boolean* functions $f : [a] \times [b] \to \{0, 1\}$. The reduction is inspired by the multi-output case, but it requires us to devise a particular kind of gadget. The existence of such a gadget can be proven using the probabilistic method, but this only yields a randomized reduction. We show the probabilistic construction can instead be derandomized using efficient constructions of universal sets [51], and this results in a deterministic reduction. So this result combines an NP-hardness reduction, a communication complexity lower bound, and a derandomization result.

After proving hardness for 3 round protocols, it is natural to use the classical round-elimination technique to prove the result for any number of rounds. However, in order to use round-elimination, we need hardness of approximation for 3 round *randomized* communication complexity. We do show, in Section 6, that it is hard to approximate the 3-round *randomized* communication complexity in a low error regime, but the parameters we obtain are just shy of what would be necessary to show hardness of approximation for d-round communication complexity, for any d. We still conjecture that better parameters can be obtained, and we show in Section 7 that our conjecture would imply that d-round communication complexity is NP-hard to approximate, by a reduction from the $(d-1)$-round case. Improving the parameters in our lower bound is left as an open problem.

In Section 5, we finish the proof of our main theorem. There we show our deterministic round-elimination lemma (Lemma 3), and use it to show that the smallest *number of leaves* in a constant-round communication protocol is NP-hard to approximate.

1.5 Sketch of Lemma 3

In this subsection we sketch the proof of our deterministic round-elimination lemma, Lemma 3. Let us restate it here:

▶ **Lemma 3.** *Let $d \geq 3$. Given an arbitrary two-player total Boolean function $f : [a] \times [b] \to \{0, 1\}$, define the function $F : ([k] \times [a]) \times ([k] \times [b] \times \{0, 1\} \times \{0, 1\}) \to \{0, 1\}$ given by*

$$F(x_0, x_1; y_0, y_1, z, i) = \begin{cases} \mathsf{XorEq}_k(x_0; y_0, z) & , \text{ if } i = 0 \\ f(x_1; y_1) & , \text{ if } i = 1, \end{cases}$$

where, in turn, $\mathsf{XorEq}_k : [k] \times ([k] \times \{0, 1\}) \to \{0, 1\}$ is given by

$$\mathsf{XorEq}_k(x; y, z) = \begin{cases} z & \text{if } x \neq y \\ 1 - z & \text{if } x = y. \end{cases}$$

Then

$$\min\{4k, 2k - 2 + \mathsf{L}^{\mathrm{B}}_d(f)\} \leq \mathsf{L}^{\mathrm{A}}_{d+1}(F) \leq 2k + \mathsf{L}^{\mathrm{B}}_d(f).$$

The upper bound is the easy direction (it follows from XorEq_k having a $2k$ leaf 2-round Alice-first protocol), so here we focus on how to prove the lower bound.

High Level Ideas. At a high level, our lower bound proof works by showing that any protocol for computing F must do one of two things:

- compute XorEq_k "twice," or
- compute XorEq_k "once" and (mostly) separately compute f in d rounds.

These two scenarios correspond to the $4k$ and $2k - 2 + \mathsf{L}^{\mathrm{B}}_{d}(f)$ parts of the lower bound respectively.

It is worth noting that an approach similar to this was used in [29] to prove a lower bound that related the depth-d and depth-$(d+1)$ *formula complexity* of two functions. Indeed, our proof was partly inspired by the proof in [29]. We note, however, that the proof in [29] differs significantly in how it implements this high level approach. We highlight a few of these differences:

- The proof of the lower bound in [29] is based on the probabilistic method (in particular, showing that some random subformulas have nice properties). Our lower bound does not involve the probabilistic method and is instead based on fooling sets.
- [29] relies on a complicated formalization of computing g "twice" that involves computing a large one-sided approximation of g with a non-deterministic formula. On the other hand, our formalization of computing XorEq_k "twice" is to contain twice as many leaves as it would take to compute XorEq_k exactly in a three-round protocol.
- [29] uses a random function g instead of the XorEq_k function.
- Our lemma is tight up to an additive 2 term, while the lower bound in [29] is only known to be tight up to a multiplicative $(1 \pm o(1))$ factor.

Another important aspect of our proof is how we make use of the XorEq_k function. The key property used about the XorEq_k function is that it has a tight fooling set lower bound: i.e., a fooling set lower bound that shows it requires exactly $2k$ leaves to compute. (The equality function has a tight fooling set lower bound for its 1-leaves, but not for its 0-leaves. XorEq_k is a modification of the equality function that "symmetrizes" the function enough that we get a tight fooling set for both the 1 and 0 leaves.) This tight fooling set severely limits the structure of monochromatic combinatorial rectangles in F, which we use in both cases of our proof.

Proof Sketch. Suppose that π is a $(d+1)$-round Alice-first protocol for F. We split into cases depending on how Alice partitions her inputs in the first round of the protocol.

Recall that Alice gets an input $(x_0, x_1) \in [k] \times [a]$. Let $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ be Alice's partition of her inputs $[k] \times [a]$ in the first round of the protocol. We say that $\mathcal{P}$ is *good* if there exists a $x_0^\star \in [k]$ such that $\{x_0\} \times [a] \subseteq P_q$ for some q. (The reason behind choosing this to be our definition of good is that it implies that one can obtain a round d protocol for solving f by considering the subprotocol of π obtained by restricting $x_0 = x_0^\star$).

If $\mathcal{P}$ is not good, then for each $x_0 \in [k]$ there are distinct $x_1', x_1'' \in [a]$ such that (x_0, x_1') and (x_0, x_1'') are contained in different parts in Alice's partition. Consequently, any leaf in π that contains an input where Alice's input is (x_0, x_1') must be distinct from any leaf in π that contains an input where Alice's input is (x_0, x_1''). We combine this "distinct leaves" property with the fooling set for XorEq_k in order to show that the protocol must spend twice as many leaves as is optimal for computing XorEq_k. Intuitively, this is because when $i = 0$, F computes $\mathsf{XorEq}_k(x_0; y_0, z)$, which doesn't depend on the value of x_1. Thus, every element $(x_0; y_0, z)$ of a fooling set for XorEq_k can be used to produce two distinct leaves in π: one leaf for when Alice gets the input (x_0, x_1') and one leaf for when Alice gets the input (x_0, x_1'').

On the other hand, suppose $\mathcal{P}$ is good. Then $\{x_0^\star\} \times [a] \subseteq P_q$ for some q. As a result, the d round Bob-first subprotocol of π, given when Alice is restricted to an input in P_q, computes f when we set $x_0 = x_0^\star$ and $i = 1$. This implies that π must have at least $\mathsf{L}_d^{\mathrm{B}}(f)$ leaves.

In fact, the goodness of $\mathcal{P}$ implies a stronger statement: that π contains at least $\mathsf{L}_d^{\mathrm{B}}(f)$ many leaves which contain an input where $x_0 = x_0^\star$. This is crucial because only two elements of the fooling set for XorEq_k satisfy $x_0 = x_0^\star$. Consequently, one can show that, in order to compute XorEq_k when $i = 0$, π must have $2k - 2$ leaves that do not contain any input where $x_0 = x_0^\star$. Putting the two bounds together, we get a $2k - 2 + \mathsf{L}_d^{\mathrm{B}}(f)$ lower bound on the number of leaves in π.

1.6 Concluding remarks and open problems

In this work we make a significant step towards showing that computing communication complexity is NP-hard, by proving it is hard to compute the smallest size of a constant-round protocol for a given function.

There are a few natural open questions suggested by our paper. The biggest question is whether the unbounded-round case is also NP-hard. But even in the constant-round setting, we would like to prove better hardness of approximation for protocol size, and we would like to prove unconditionally that communication complexity is NP-hard. Proving Conjecture 33 would give us both of these results for subexponential-time reductions. Can we get such hardness using polynomial-time reductions, as well? On the other hand, one can also ask: do there exist non-trivial polynomial-time approximation algorithms for computing constant-round communication complexity? It is worth noting that the log-rank conjecture gives a candidate approximation algorithm for computing communication complexity with no bound on the number of rounds.

A crucial ingredient in our hardness result is a deterministic version of the round elimination lemma, which is proved using entirely different techniques than the original version. Does this deterministic version have other applications? Can the new ideas in our proof be used to prove other interesting statements?

2 Preliminaries

For a positive integer n, we let $[n]$ denote the set $\{1, \ldots, n\}$.

Binomial Coefficients and Projections. The binomial coefficient $\binom{n}{k}$ equals the number of distinct subsets of $[n]$ with exactly k elements. Similarly, $\binom{n}{\leq k}$ denotes the number of distinct subsets of $[n]$ that have at most k elements. Finally, $\binom{[n]}{k}$ denotes the set of all subsets of $[n]$ with exactly k elements.

If $x \in \{0,1\}^n$ and $S = \{i_1, \ldots, i_k\} \subseteq \binom{[n]}{k}$, then $x_S \in \{0,1\}^k$ denotes the *projection of x to S*, given by $x_S = x_{i_1} \ldots x_{i_k}$.

Entropy, Mutual Information, Pinsker's inequality. We describe the notation we will use for various information-theoretic quantities, and some basic facts about them. We will not define the notions here, or prove the basic facts. See [58] for a reference that uses these notions in the context of communication complexity. Given a random variable $\mathbf{x} \in X$, we denote its entropy by $\mathsf{H}(\mathbf{x})$. Given random variables $\mathbf{x}, \mathbf{y}, \mathbf{z}$, we will denote the mutual information between $\mathbf{x}$ and $\mathbf{y}$, given $\mathbf{z}$, by $\mathsf{I}(\mathbf{x} : \mathbf{y} \mid \mathbf{z})$. It always holds that $\mathsf{I}(\mathbf{x} : \mathbf{y}) \leq \mathsf{H}(\mathbf{y})$. If we have random variables $\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}$, we then have the *chain rule*:

$$\mathsf{I}(\mathbf{x}_1, \ldots, \mathbf{x}_n : \mathbf{y}) = \sum_{i=1}^{n} \mathsf{I}(\mathbf{x}_i : \mathbf{y} \mid \mathbf{x}_{<i}),$$

where $\mathbf{x}_{<i} = \mathbf{x}_1, \dots, \mathbf{x}_{i-1}$. If two random-variables $\mathbf{x}$ and $\mathbf{y}$ have $\mathsf{I}(\mathbf{x} : \mathbf{y}) \leq 2\varepsilon^2$, then Pinsker's inequality implies (by concavity) that if we compute the average, over the choice y for $\mathbf{y}$, statistical distance between the distribution of $\mathbf{x}$, and the distribution of $\mathbf{x}$ conditioned on $\mathbf{y} = y$, then this average is less than ε.

Communication complexity. We assume basic familiarity with communication complexity [38]. We write the definitions here for clarity.

▶ **Definition 5** (Protocol). *Let $\mathcal{A}, \mathcal{B}, \mathcal{Z}$ be finite sets. A* deterministic protocol *π over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$ is a rooted tree:*

- *Each node v is associated with a rectangle $\pi^{-1}(v) = A \times B$, with $A \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$.*
- *Each non-leaf node v, associated with a rectangle $\pi^{-1}(v) = A \times B$, is labeled by either (a) a partition $A = \dot\bigcup_{c \in \mathcal{P}_v} A_c$ of A, in which case we say it is* Alice's node *or (b) a partition $B = \dot\bigcup_{c \in \mathcal{P}_v} B_c$ of B, in which case we say it is* Bob's node.
- *Each leaf node is labeled by an element of the output domain $\mathcal{Z}$*
- *The rectangle associated with the root is the input domain $\mathcal{A} \times \mathcal{B}$.*
- *If a non-leaf node v of Alice is associated with rectangle $\pi^{-1}(v) = A \times B$ and (a) labeled by a partition $A = \dot\bigcup_{c \in \mathcal{P}_v} A_c$ of A, then for each $c \in \mathcal{P}_v$ there will be one child v_c of v, which will be associated with the rectangle $\pi^{-1}(v_c) = A_c \times B$; similarly for Bob's nodes.*

We let the leaf complexity of π, *written $\mathsf{L}(\pi)$, be the number of leaf nodes of π. We let the* round complexity of π, *written $\mathsf{R}(\pi)$, be the height of π, i.e., the maximum number of edges in any root-to-leaf path of π.*

A root-to-leaf path $v_1 \to \cdots \to v_{k+1}$ is said to have communication length $\sum_{i=1}^{k} \lceil \log |P_{v_i}| \rceil$ *(where $|P_{v_i}|$, as defined above, is the number of parts in the partition of A or B associated with node v_i). We then let the* communication complexity of π, *written $\mathsf{C}(\pi)$, be the maximum communication length of any root-to-leaf path of π.*

Given $(a, b) \in \mathcal{A} \times \mathcal{B}$, we let $\pi(a, b)$ denote the (unique) leaf v of π having $(a, b) \in \pi^{-1}(v)$. For $z \in \mathcal{Z}$, we may write $\pi(a, b) = z$ to mean that the leaf $\pi(a, b)$ is labeled by z.

A randomized protocol *over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$ is a distribution over deterministic protocols over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$. We will use a boldface Greek letter, such as $\boldsymbol{\pi}$, to denote a protocol sampled from this distribution. We then let $\mathsf{L}(\boldsymbol{\pi})$ be the maximum $\mathsf{L}(\pi)$ over all π in the support of $\boldsymbol{\pi}$, and likewise for $\mathsf{R}(\boldsymbol{\pi})$ and $\mathsf{C}(\boldsymbol{\pi})$.*

▶ **Definition 6.** *A function $f : \mathcal{A} \times \mathcal{B} \to \mathcal{Z}$ is said to be* computed *by a deterministic protocol π over $\mathcal{A} \times \mathcal{B} \times \mathcal{Z}$ if we have $f(a, b) = \pi(a, b)$ for every $(a, b) \in \mathcal{A} \times \mathcal{B}$. Furthermore, if $\varepsilon \in [0, 1]$, then f is said to be* computed with error ε *by a randomized protocol $\boldsymbol{\pi}$ if, for every $(a, b) \in \mathcal{A} \times \mathcal{B}$, $\Pr[\boldsymbol{\pi}(a, b) = f(a, b)] > 1 - \varepsilon$ (the probability is over the choice of $\boldsymbol{\pi}$).*

We may then define:

- *$\mathsf{L}^{\mathrm{A}}_{d}(f)$ is the minimum leaf complexity $\mathsf{L}(\pi)$ among all deterministic protocols π that compute f, and have round complexity $\mathsf{R}(\pi) \leq d$, and such that the root node of π is Alice's. $\mathsf{L}^{\mathrm{B}}_{d}(f)$ is defined likewise, but for protocols where the root node is Bob's.*
- *$\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(f)$ is the minimum $\mathsf{L}(\boldsymbol{\pi})$ among all randomized protocols $\boldsymbol{\pi}$ that compute f with error ε, and have $\mathsf{R}(\boldsymbol{\pi}) \leq d$, and such that the root node of $\boldsymbol{\pi}$ is (always) Alice's. $\mathsf{L}^{\mathrm{B}}_{d,\varepsilon}(f)$ is defined likewise for randomized protocols where the root node is (always) Bob's.*
- *$\mathsf{C}^{\mathrm{A}}_{d}$, $\mathsf{C}^{\mathrm{B}}_{d}$, $\mathsf{C}^{\mathrm{A}}_{d,\varepsilon}$, $\mathsf{C}^{\mathrm{B}}_{d,\varepsilon}$ are defined analogously where the communication complexity $\mathsf{C}(\pi)$ replaces the leaf complexity $\mathsf{L}(\pi)$.*

Since the communication transcript of a given run of protocol determines the leaf, it must follow that:

▶ **Proposition 7.** *For any protocol π, $\log \mathsf{L}(\pi) \leq \mathsf{C}(\pi)$.*

Chromatic number. All our NP-hardness reductions are from the chromatic number problem:

▶ **Definition 8** (Chromatic number)**.** *A* coloring *of an undirected graph G, is a partition of the vertices such that no edge has both endpoints in the same part. The* chromatic number *of a graph G, denoted $\chi(G)$, is the smallest number of parts in a coloring of G.*

The NP-hardness of approximating the chromatic number has been established by a series of results [42, 24, 20], culminating in a paper by Zuckerman [65], where the following was proven:

▶ **Theorem 9** (Hardness For Chromatic Number)**.** *For every $\epsilon > 0$, there is a deterministic polynomial time algorithm that on an input $x \in \{0,1\}^*$ outputs a graph G on n vertices such that*

- *if x is a YES instance of* SAT*, then $\chi(G) \leq n^{\epsilon}$, and*
- *if x is a NO instance of* SAT*, then $\chi(G) \geq n^{1-\epsilon}$.*

3 Warmup: deterministic 3-round protocols, large output alphabet

In this section, we show that it is NP-hard to approximate the deterministic 3-round communication complexity of a given matrix over a large alphabet.

We start by observing that deterministic 3-round communication complexity may be approximated by a very simple combinatorial quantity.

▶ **Definition 10.** *Let $\mathcal{A}, \mathcal{B}, \mathcal{Z}$ be finite sets, and let M be an $\mathcal{A} \times \mathcal{B}$ matrix over $\mathcal{Z}$. Let $\mathcal{P} = \{P_i\}_{i\in[k]}$ be a partition of (the columns) $\mathcal{B}$ for some $k \in \mathbb{N}$. (That is, $\emptyset \neq P_i \subseteq \mathcal{B}$, $\bigcup_{i\in[k]} P_i = \mathcal{B}$ and $P_i \cap P_j = \emptyset$ for every $i \neq j$.) For a subset $P \subseteq \mathcal{B}$ of columns, we denote by $\mathsf{Cost}_M(\mathcal{P})$ the number of distinct rows of M restricted to columns in P, i.e.,*

$$\mathsf{Cost}_M(\mathcal{P}) = |\{x_P \in \mathcal{Z}^P \mid x \in \mathcal{Z}^{\mathcal{B}} \text{ is a row of } M\}|.$$

We further define $\mathsf{Cost}_M(\mathcal{P})$ to be $\sum_{i=1}^{k} \mathsf{Cost}_M(P_i)$.

▶ **Proposition 11.** *Let $f \colon \mathcal{A} \times \mathcal{B} \to \mathcal{Z}$ be a function and M be the $\mathcal{A} \times \mathcal{B}$ communication matrix (with entries in $\mathcal{Z}$) that corresponds to f. Let $q \in \mathbb{N}$ be the maximum number of distinct values in a single row of M. We then have*

$$L \leq \mathsf{L}_3^{\mathrm{B}}(f) \leq L \cdot q, \qquad \text{where } L = \min_{\mathcal{P}} \mathsf{Cost}_M(\mathcal{P}),$$

and where the minimum is taken over all partitions $\mathcal{P}$ of $\mathcal{B}$. Furthermore we have that

$$\log L \leq \mathsf{C}_3^{\mathrm{B}}(f) \leq \log L + \log q + O(1).$$

Proof. It may be easily seen that $\mathsf{L}_2^{\mathrm{A}}(f)$ is lower-bounded by the number of distinct rows in the communication matrix of f. Because if Alice's partition of the rows includes a part with two different rows, then Bob's ensuing partition of the columns cannot avoid having a non-monochromatic column. It then follows that, if we have a 3-round protocol π where Bob speaks first, and $\mathcal{P}$ is Bob's partition of the columns in the first round, then $\mathsf{Cost}_M(\mathcal{P})$ is a lower-bound on smallest number of leaves that π needs to use to compute f, and thus $\mathsf{L}_3^{\mathrm{B}}(f) \geq L$. The lower-bound on $\mathsf{C}_3^{\mathrm{B}}(f)$ now follows from Proposition 7.

Conversely, it is also easy to see that $\mathsf{L}_2^{\mathrm{A}}(f)$ is upper-bounded by q times the number of distinct rows in the communication matrix of f. The protocol that achieves this bound has Alice tell which kind of row she has, and now in each rectangle the rows are all equal, hence Bob can just tell Alice the color of his column in this row, of which there are q possibilities. Thus $\mathsf{L}_3^{\mathrm{B}}(f) \leq L \cdot q$, and the same protocol shows that $\mathsf{C}_3^{\mathrm{B}}(f) \leq \log L + \log q + O(1)$. ◀

In general, the approximate factor q of Proposition 11 can be as large as $|\mathcal{Z}|$. However, in the following construction, we will construct a matrix where each row has at most $q = 3$ distinct values; in this case, Proposition 11 guarantees that $\min_{\mathcal{P}} \mathsf{Cost}_M(\mathcal{P})$ provides a 3-factor approximation of $\mathsf{L}_3(M)$.

▶ **Theorem 12.** *Given an undirected graph $G = ([n], E)$ with n vertices and $|E| = m > 0$ edges, one may construct in deterministic polynomial time a function $f_G\colon [a] \times [n] \to \{0, 1, \ldots, \ell\}$, with $\ell = m^2 n$, $k = \sqrt{n\ell} = mn$ and $a = \ell + m \cdot k^2$, such that*

$$\chi(G) \cdot \ell \leq \mathsf{L}_3^{\mathrm{B}}(f_G) \leq \chi(G) \cdot 6\ell.$$

Furthermore, we also have

$$\log \chi(G) + \log \ell \leq \mathsf{C}_3^{\mathrm{B}}(f_G) \leq \log \chi(G) + \log \ell + O(1).$$

Proof. We let A_ℓ denote the $\ell \times 1$ column vector,

$$A_\ell = \begin{bmatrix} 1 \\ \vdots \\ \ell \end{bmatrix}.$$

We let B_k and C_k denote the $k^2 \times 1$ column vectors

$$B_k = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ k \\ \vdots \\ k \end{bmatrix}, \qquad C_k = \begin{bmatrix} 1 \\ \vdots \\ k \\ \vdots \\ 1 \\ \vdots \\ k \end{bmatrix},$$

where each value $i \in [k]$ appears k times. Given an edge $\{v, w\} \in [n] \times [n]$, with $v < w$, we define the $k^2 \times n$ matrix M so that

$$M_{\{v,w\}} = \begin{bmatrix} 0 & \ldots & 0 & B_k & 0 & \ldots & 0 & C_k & 0 & \ldots & 0 \end{bmatrix},$$

where B_k appears in the v-th column and C_k in the w-th column. Finally, let $E = \{e_1, \ldots, e_m\}$ denote the edges of G; then we define the $a \times n$ communication matrix f_G so that

$$f_G = \begin{bmatrix} A_\ell & \cdots & A_\ell \\ & M_{e_1} & \\ & \vdots & \\ & M_{e_m} & \end{bmatrix} \in \{0, 1, \ldots, \ell\}^{a \times n},$$

where $a = \ell + m \cdot k^2$. Observe that each row of f_G has at most $q = 3$ distinct values; thus, Proposition 11 provides a 3-factor approximation. We now show the stated inequality, namely, that

$$\chi(G) \cdot \ell \leq \mathsf{L}_3^{\mathrm{B}}(f_G) \leq \chi(G) \cdot 2q\ell.$$

The upper-bound is easy to see. Given a coloring of G into $\chi(G)$ colors, we may take the 3-round protocol π where Bob first tells Alice which color his vertex has. This partitions the columns by a partition $\mathcal{P} = \{P_c\}_{c \in [\chi(G)]}$ formed of the various color classes of our coloring. Fix any color c and consider $\mathsf{Cost}_{f_G}(P_c)$. Since P_c is an independent set of G, the C_k and B_k columns of each M_{e_i} sub-matrix will always be placed in different parts; therefore, $\mathsf{Cost}_{f_G}(P_c) \leq \ell + mk$, where the first term counts the number of rows in A_ℓ and the second term counts the number of distinct rows in M_{e_i} restricted to columns of P_c for each $i \in [m]$. Using Proposition 11, we obtain

$$\mathsf{L}_3^{\mathrm{B}}(f_G)/q \leq \mathsf{Cost}_{f_G}(\mathcal{P}) = \sum_c \mathsf{Cost}_{f_G}(P_c) \leq \chi(G) \cdot (\ell + mk) = \chi(G) \cdot 2\ell,$$

and Proposition 11 also gives us

$$\mathsf{C}_3^{\mathrm{B}}(f_G) \leq \log \chi(G) + \log \ell + O(1).$$

For the other direction, let π be any 3-round protocol for f_G. The first round of π partitions the columns of f_G, which is to say, it partitions the vertices of G by some partition $\mathcal{P} = \{P_i\}_{i \in I}$. We claim that $\chi(G) \cdot \ell \leq \mathsf{Cost}_{f_G}(\mathcal{P})$ by analyzing the following two cases.

1. If the partition $\mathcal{P}$ does not form a coloring of G, then there must exist an edge $\{v, w\}$ such that the v-th column and the w-th column of G are placed in the same part P_i. This means that the C_k and B_k columns of the $M_{\{v,w\}}$ sub-matrix are both placed in P_i. In this case, we have $\mathsf{Cost}_{f_G}(P_i) \geq k^2 \geq n \cdot \ell \geq \chi(G) \cdot \ell$, and thus the lower bound holds.
2. Otherwise, suppose that the partition $\mathcal{P}$ does form a coloring of G. Then the number of parts is $\geq \chi(G)$, and so just the contribution from the first ℓ rows gives us $\mathsf{Cost}_{f_G}(\mathcal{P}) \geq \chi(G) \cdot \ell$.

The lower-bounds on $\mathsf{L}_3^{\mathrm{B}}(f_G)$ and $\mathsf{C}_3^{\mathrm{B}}(f_G)$ then follow from Proposition 11. ◀

The following corollary shows that it is NP-hard to approximate $\mathsf{L}_3^{\mathrm{B}}(f)$, for a given total two player function $f : [N] \times [N] \to [N]$, with an approximation ratio better than (roughly) $N^{\frac{1}{5}}$.

▶ **Corollary 13.** *For every $L \subseteq \{0,1\}^*$ in NP and every constant $\varepsilon > 0$, there exists a polynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in [N]^{N \times N}$ such that if $x \in L$ then $\mathsf{L}_3^{\mathrm{B}}(M) \leq N$, and if $x \notin L$ then $\mathsf{L}_3^{\mathrm{B}}(M) > N^{\frac{6}{5} - \varepsilon}$.*

Proof. We compose the hardness of approximation result for chromatic number in Theorem 9 with the reduction of Theorem 12, padding the communication matrix with all-0 columns to make it square (since Bob speaks first, this adds at most one leaf), so the communication matrix of f is an $N \times N$ matrix with $N = a = \Theta(n^5)$. The parameters then come from the fact that $n = \Theta(N^{\frac{1}{5}})$. ◀

Using the bounds on $\mathsf{C}_3^{\mathrm{B}}$ instead of the bounds on $\mathsf{L}_3^{\mathrm{B}}$ from Theorem 12, we conclude that it is NP-hard to approximate $\mathsf{C}_3^{\mathrm{B}}(f)$, for a given total two player function $f : [N] \times [N] \to [N]$, with an approximation ratio better than (roughly) $\frac{6}{5}$. More precisely:

▶ **Corollary 14.** *For every $L \subseteq \{0,1\}^*$ in NP and every $\varepsilon > 0$, there exists a polynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in [N]^{N \times N}$ such that if $x \in L$ then $\mathsf{C}_3^{\mathrm{B}}(M) \leq \log N$, and if $x \notin L$ then $\mathsf{C}_3^{\mathrm{B}}(M) > (\frac{6}{5} - \varepsilon) \log N$.*

4 Hardness for deterministic 3-round protocols

Building on the proof ideas presented in Section 3, in this section, we prove NP-hardness of approximating the communication complexity of deterministic 3-round protocols. A key building block is to use the notion of universal set.

▶ **Definition 15** (Universal set). *Let $r, c, k \in \mathbb{N}$, and let $M \in \{0,1\}^{r\times c}$ be a matrix whose columns are $M^{(1)}, \dots, M^{(c)}$. We say that M is (c,k)-*universal *if, for every set $\{y_1, \dots, y_k\} \subseteq [c]$ of k columns of M, the matrix*

$$\begin{bmatrix} M^{(y_1)} & \dots & M^{(y_k)} \end{bmatrix}$$

*has 2^k distinct rows. The set of all the rows of M is called a (c,k)-*universal set.

We use the explicit construction of a universal set due to [49].

▶ **Lemma 16** (Naor, Schulman and Srinivasan [51]). *There exists a deterministic algorithm that, given c and $k \in \mathbb{N}$, outputs a (c,k)-universal matrix $M \in \{0,1\}^{r\times c}$ such that $r = 2^{k+O(\log k)^2} \cdot \log c$ in time a polynomial in c and 2^k.*

Now we state the main result of this section.

▶ **Theorem 17.** *Let $\epsilon > 0$ be an arbitrary constant. Given an undirected graph $G = ([n], E)$ with n vertices and $|E| = m > 0$ edges, one may construct in deterministic polynomial time a function $f_G : [a] \times [b] \to \{0,1\}$ and a number $\ell \in \mathbb{N}$, with $a, b, \ell = O(n^8)$, such that*

$$\chi(G) \cdot \ell \le \mathsf{L}_3^{\mathrm{B}}(f_G) \le \chi(G) \cdot \ell^{1+\epsilon}$$

and

$$\log \chi(G) + \log \ell \le \mathsf{C}_3^{\mathrm{B}}(f_G) \le \log \chi(G) + (1+\epsilon) \cdot \log \ell.$$

The idea of the proof is to build upon the reduction in the proof of Theorem 12, by replacing each column with entries from $[\ell]$ with a block of columns that have entries from $\{0,1\}$. The difficulty in making this work is that a partition of the columns might not respect our blocks and could place columns from the same block into different parts. We solve this by thinking as follows. Either the partition is large, meaning it has many parts, so the protocol also has many leaves, which proves our lower bound, or otherwise for any block C_v of columns the part P_i which has most columns from C_v has many columns from C_v; we may then act as if "C_v was placed in P_i".

Proof. Let $t, k, c \in \mathbb{N}$ be parameters chosen later. Let $A \in \{0,1\}^{r\times c}$ and $M \in \{0,1\}^{s\times c}$ be the (c,t)-universal and (c,k)-universal matrices, respectively, that are constructed by the polynomial-time deterministic algorithm of Lemma 16; then we have $r = 2^{(1+o(1))\cdot t} \cdot \log c$ and $s = 2^{(1+o(1))\cdot k} \cdot \log c$.

Let $x_1, \dots, x_s$ be the rows of M. We then let B and C denote the $s^2 \times c$ matrices

$$B = \begin{bmatrix} x_1 \\ \vdots \\ x_1 \\ \vdots \\ x_s \\ \vdots \\ x_s \end{bmatrix}, \qquad C = \begin{bmatrix} x_1 \\ \vdots \\ x_s \\ \vdots \\ x_1 \\ \vdots \\ x_s \end{bmatrix},$$

where a row x_i appears s times for each $i \in [s]$. Given an edge $\{v, w\} \in [n] \times [n]$, with $v < w$, we define the $s^2 \times c \cdot n$ matrix:

$$M_{\{v,w\}} = \begin{bmatrix} 0 & \dots & 0 & B & 0 & \dots & 0 & C & 0 & \dots & 0 \end{bmatrix},$$

where B appears in the v-th block of c columns and C in the w-th block of c columns. Finally, let $E = \{e_1, \dots, e_m\}$ denote the edges of G; then we define the $(r + m \cdot s^2) \times c \cdot n$ communication matrix f_G so that

$$f_G = \begin{bmatrix} A \quad \dots \quad A \\ M_{e_1} \\ \vdots \\ M_{e_m} \end{bmatrix} \in \{0,1\}^{(r+m \cdot s^2) \times c \cdot n}.$$

Let $\ell := 2^t$. Let $\mathcal{P}$ be a partition of the columns of f_G that minimizes $\mathsf{Cost}_{f_G}(\mathcal{P})$. We now show that, for a suitable choice of t, k and c,

$$\chi(G) \cdot \ell \leq \mathsf{Cost}_{f_G}(\mathcal{P}) \leq \chi(G) \cdot \ell^{1+o(1)}.$$

This will complete the proof, because Proposition 11 shows that, for a binary matrix f_G, $\mathsf{Cost}_{f_G}(\mathcal{P})$ is a 2-factor approximation of $\mathsf{L}_3^{\mathrm{B}}(f_G)$ and $\log \mathsf{Cost}_{f_G}(\mathcal{P})$ is an approximation of $\mathsf{C}_3^{\mathrm{B}}(f_G)$ up to an additive $O(1)$ term. We will choose the parameters t, k and c so that the following conditions are satisfied.

1. Condition 1. $r + ms \leq \ell^{1+o(1)}$.
2. Condition 2. $c \geq n\ell \cdot \max\{t, k\}$.
3. Condition 3. $2^{2k} \geq n\ell$.

Given that Condition 1 is satisfied, the complexity upper-bounds are easy to see. Let $\mathcal{P}$ be a coloring of G that partitions the vertex set into $\chi(G)$ parts. Since no class contains an edge, the C and B sub-matrices of each M_{e_i} sub-matrix will always be placed in different parts, and we thus have

$$\mathsf{Cost}(\mathcal{P}) \leq \chi(G) \cdot (r + ms) \leq \chi(G) \cdot \ell^{1+o(1)}.$$

For the other direction, we will lower-bound $\mathsf{Cost}(\mathcal{P})$ for every partition $\mathcal{P}$. To begin, if the number of parts is $|I| \geq n\ell$, then we must conclude that $\mathsf{Cost}(\mathcal{P}) \geq n\ell \geq \chi(G) \cdot \ell$, as desired.

Otherwise, $|I| \leq n\ell$. Then for every block of columns $v \in [n]$ there must exist a part $P_{i(v)}$ which contains at least $c/|I| \geq c/n\ell$ columns from the v-th block. Observe that $c/n\ell \geq \max\{t, k\}$ by Condition 2.

Now, either the mapping $v \mapsto i(v)$ is a valid coloring of G or not. First, suppose that the mapping $v \mapsto i(v)$ is not a valid coloring of G, meaning that there exists an edge $\{v, w\} \in E$ such that $i = i(v) = i(w)$. Then the v-th column block and the w-th column block each have at least $\max\{t, k\}$ columns in the same part P_i. This will mean that the C and B sub-matrices of the $M_{\{v,w\}}$ sub-matrix each have at least k columns in P_i. But then, since M is (c, k)-universal, it follows from Condition 3 that $\mathsf{Cost}(\mathcal{P}) \geq \mathsf{Cost}(P_i) \geq 2^{2k} \geq n\ell$.

Next, suppose that $i\colon [n] \to \mathcal{P}$ does form a coloring of G. Then there exist at least $\chi(G)$ parts P_i each receiving at least t columns from some column block, and so, since A is (c, t)-universal, the contribution from the A columns give us $\mathsf{Cost}(\mathcal{P}) \geq \chi(G) \cdot 2^t = \chi(G) \cdot \ell$.

This will give us the theorem, and all we are left to do is ensure that the various conditions can be met: We define t so that $\ell = 2^t$ satisfies that $\ell \le (nm^2)^{1+\epsilon/2} < 2\ell$. To meet Condition 3, let k be the smallest integer satisfying that $n\ell \le 2^{2k}$. Let $c := n\ell \cdot \max\{t, k\}$ so that Condition 2 is satisfied. Finally, observe that Condition 1 is satisfied because

$$r + ms \le \ell^{1+o(1)} \cdot \log c + m \cdot 2^{(1+o(1))\cdot k} \cdot \log c \le \ell^{1+o(1)} + m(n\ell)^{\frac{1}{2}+o(1)} \cdot \log c \le \ell^{1+\epsilon},$$

where the last inequality holds for all large n, m. ◀

We can now prove that $\mathsf{L}_3^{\mathrm{A}}$ and $\mathsf{C}_3^{\mathrm{A}}$ are hard to approximate, also for Boolean functions.

▶ **Theorem 18.** *For every constant $\varepsilon > 0$, there exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N\times N}$ and a number $k \in \mathbb{N}$, with $k \le N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of* SAT*, then $\mathsf{L}_3^{\mathrm{B}}(M) \le k$ and $\mathsf{C}_3^{\mathrm{B}}(M) \le \log k$, and*
2. *if x is a NO instance of* SAT*, then $\mathsf{L}_3^{\mathrm{B}}(M) \ge N^{\frac{1}{8}-\varepsilon} \cdot k$ and $\mathsf{C}_3^{\mathrm{B}}(M) \ge \log k + (\frac{1}{8} - \varepsilon)\log N$.*

Proof. We combine the two reductions of Theorems 9 and 17, which we invoke with the same parameter ε. Fix any input x and let G be an n-vertex graph that is produced by the reduction of Theorem 9 on input x. Let $M \in \{0,1\}^{a\times b}$ be the communication matrix of f_G, and $\ell \in \mathbb{N}$, be given by the reduction of Theorem 17 on input G, and set $N = \max(a, b) = O(n^8)$, $k = n^\varepsilon \cdot \ell^{1+\varepsilon}$. We may assume that $a = b$ without loss of generality, since otherwise we may pad M with all-0 rows or all-0 columns, and this changes the leaf complexity by at most a factor of 2, and the communication complexity by at most 1 bit, so this change makes no difference to the result.

We verify that the inequalities are satisfied for M: If $x \in L$, then we have $\mathsf{L}_3^{\mathrm{B}}(M) \le \chi(G) \cdot \ell^{1+\epsilon} \le n^\varepsilon \cdot \ell^{1+\epsilon} = k$. If $x \notin L$, then we have $\mathsf{L}_3^{\mathrm{B}}(M) \ge \chi(G) \cdot \ell \ge n^{1-\epsilon} \cdot \ell \ge N^{\frac{1}{8}-O(\epsilon)} \cdot k$. Similar inequalities hold for $\mathsf{C}_3^{\mathrm{B}}(M)$. Since ε can be arbitrarily small, we may ignore the constant factors. ◀

5 From 3-rounds to multiple rounds using deterministic round elimination

In this section we show how we can use an oracle for computing $\mathsf{L}_{d+1}^{\mathrm{B}}$ in order to compute $\mathsf{L}_d^{\mathrm{A}}$. The gadget in our reduction involves a special function, XorEq_k, which is a small modification of the standard Equality function.

▶ **Definition 19.** *The function $\mathsf{XorEq}_k : [k] \times ([k] \times \{0,1\}) \to \{0,1\}$ is given by*

$$\mathsf{XorEq}_k(x; y, z) = \begin{cases} z & \text{if } x \ne y. \\ 1 - z & \text{if } x = y. \end{cases}$$

The key property of XorEq_k is that it has a fooling set lower bound that is tight. In particular, $\{(x; y, z) : x = y\}$ is a fooling set of cardinality $2k$, and there is a 2-round protocol for solving XorEq_k with $2k$ leaves (where Alice just sends her full input to Bob, and he replies with the output).

▶ **Lemma 20** (Fooling set lower-bound for XorEq_k). *Let π be a protocol for solving the function $f : ([k] \times [a]) \times ([k] \times \{0,1\}) \to \{0,1\}$ given by $f(x_0, x_1; y, z) = \mathsf{XorEq}_k(x_0; y, z)$.*[3] *Then*

$$\pi(x_0, x_1; y, z) \neq \pi(x'_0, x'_1; y', z')$$

if either

- $y = x_0 \neq x'_0$, *or*
- $y = x_0 = x'_0$ *and* $z \neq z'$.

Proof. First, suppose for contradiction that $\pi(x_0, x_1; y, z) = \pi(x'_0, x'_1; y', z')$ and $y = x_0 \neq x'_0$. Since leaves are combinatorial rectangles, we can infer that $\pi(x'_0, x'_1; y, z) = \pi(x_0, x_1; y, z)$. But since $y = x_0 \neq x'_0$, we know that

$$f(x_0, x_1; y, z) = 1 - z \neq z = f(x'_0, x'_1; y, z)$$

so this contradicts that $\pi(x_0, x_1; y, z)$ is a monochromatic leaf.

Similarly, if $y = x_0 = x'_0$ and $z \neq z'$, then we have $f(x_0, x_1; y, z) \neq f(x'_0, x'_1; y', z')$, so $\pi(x_0, x_1; y, z) \neq \pi(x'_0, x'_1; y', z')$ by monochromaticness. ◀

The main technical portion of our reduction is the following *deterministic* variant of the round-elimination lemma.

▶ **Lemma 21** (Restatement of Lemma 3). *Let $d \geq 3$. Let $f : [a] \times [b] \to \{0,1\}$. Let $F : ([k] \times [a]) \times ([k] \times [b] \times \{0,1\} \times \{0,1\}) \to \{0,1\}$ be given by*

$$F(x_0, x_1; y_0, y_1, z, i) = \begin{cases} \mathsf{XorEq}_k(x_0; y_0, z) & , \text{ if } i = 0 \\ f(x_1; y_1) & , \text{ if } i = 1. \end{cases}$$

Then we have

$$\min\{4k, 2k - 2 + \mathsf{L}^{\mathrm{B}}_d(f)\} \leq \mathsf{L}^{\mathrm{A}}_{d+1}(F) \leq 2k + \mathsf{L}^{\mathrm{B}}_d(f).$$

Proof. The upper bound comes from the protocol where Alice skips the first round of communication, Bob sends i to Alice and begins running the best d-round Bob-first protocol for XorEq_k or f, based on whether $i = 0$ or $i = 1$. In particular, we have that

$$\mathsf{L}^{\mathrm{A}}_{d+1}(F) \leq \mathsf{L}^{\mathrm{B}}_d(\mathsf{XorEq}_k) + \mathsf{L}^{\mathrm{B}}_d(f) \leq 2k + \mathsf{L}^{\mathrm{B}}_d(f),$$

where the last upper bound uses that $d \geq 3$.

We now argue the lower bound. Suppose π is a $(d+1)$-round Alice-first protocol for F. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_{\mathsf{L}(\pi)}\}$ denote the set of leaves of π.

Our arguments split into two cases depending on whether there is a *good* input $x_0 \in [k]$. We say an input $x_0 \in [k]$ is good if all of Alice's inputs that begin with x_0 are placed in a single partition. More formally, let $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ be the partition of Alice's inputs corresponding to the first round of π. We say $x_0 \in [k]$ is *good* if there exists a $q \in [|\mathcal{P}|]$ such that $\{x_0\} \times [a] \subseteq P_q$.

[3] In our definition of f, the input x_1 does not affect the output of the function. The fact that this lemma holds even when there is an extraneous input like x_1 will be used later.

Case 1: There is a good input. Suppose that there exists a good input $x_0^\star \in [k]$ such that $\{x_0^\star\} \times [a] \subseteq P_q$ for some q. Let $\mathcal{L}_{x_0^\star}$ be the set of leaves of π that contain an input where $x_0 = x_0^\star$, that is,

$$\mathcal{L}_{x_0^\star} = \{\ell \in \mathcal{L} : \exists (x_1, y_0, y_1, z, i) \text{ with } \pi(x_0^*, x_1; y_0, y_1, z, i) = \ell\}$$

and let $\overline{\mathcal{L}_{x_0^\star}}$ denote the complementary set of leaves, that is $\overline{\mathcal{L}_{x_0^\star}} = \mathcal{L} \setminus \mathcal{L}_{x_0^\star}$. We will show that $|\mathcal{L}_{x_0^\star}| \geq \mathsf{L}_d^{\mathrm{B}}(f)$ and that $|\overline{\mathcal{L}_{x_0^\star}}| \geq 2k - 2$. As a result, we get that

$$\mathsf{L}(\pi) \geq |\mathcal{L}_{x_0^\star}| + \overline{\mathcal{L}_{x_0^\star}} \geq \mathsf{L}_d^{\mathrm{B}}(f) + 2k - 2,$$

as desired.

First, we show that $|\mathcal{L}_{x_0^\star}| \geq \mathsf{L}_d^{\mathrm{B}}(f)$. Let π' be the d-round Bob-first subprotocol of π given when Alice says that her input is in P_q at the first round. Since $\{x_0^\star\} \times [a] \subseteq P_q$, it follows that π' computes $f(x_1, y_1)$ on input $(x_0^\star, x_1; y_0, y_1, 0, 1)$ for all $x_1 \in [a]$ and $y_1 \in [b]$. This yields a d-round Bob-first protocol for computing f, and therefore, the number of leaves in π' containing the input $x_0^\star$ must be at least $\mathsf{L}_d^{\mathrm{B}}(f)$. Hence, $|\mathcal{L}_{x_0^\star}| \geq \mathsf{L}_d^{\mathrm{B}}(f)$.

Next, we argue that $|\overline{\mathcal{L}_{x_0^\star}}| \geq 2k - 2$. Consider the set of leaves given by

$$\{\pi(x_0, x_1; y_0, y_1, z, i) : i = 0, x_0 = y_0 \in [k] \setminus \{x_0^\star\}, y_1 = 1, z \in \{0, 1\}\}.$$

If we consider the restriction of π to the inputs $y_1 = 1$ and $i = 0$, we can apply Lemma 20 to conclude that all $2k - 2$ leaves in this set are in $\overline{\mathcal{L}_{x_0^\star}}$ and are pairwise distinct.

Case 2: No good input. Now we consider the case where there is no good input. For each $x_0 \in [k]$, we define a set $A_{x_0} \subseteq [a]$ of cardinality 2 as follows. Since $x_0 \in [k]$ is not good, there exists a pair $(x_1, x_1') \in [a]^2$ such that $\pi(x_0, x_1; y_0, y_1, z, i) \neq \pi(x_0, x_1'; y_0, y_1, z, i)$ for all y_0, y_1, z and i. Let $A_{x_0} = \{x_1, x_1'\}$. This completes our definition of A_{x_0}.

We claim that the $4k$ inputs in the following set are all in pairwise distinct leaves:

$$W = \{(x_0, x_1, y_0, y_1, z, i) : x_0 = y_0 \in [k], i = 0, x_1 \in A_{x_0}, y_1 = 1, z \in \{0, 1\}\}.$$

To see this, suppose that $w \neq w'$ for some $w, w' \in W$. Let $w = (x_0, x_1; y_0, y_1, z, i)$ and $w' = (x_0', x_1'; y_0', y_1', z', i')$. We prove $\pi(w) \neq \pi(w')$ by considering the following three cases.

1. If $x_0 = x_0'$ and $x_1 \neq x_1'$, then we know that $\{x_1, x_1'\} = A_{x_0}$. By the construction of A_{x_0}, we can conclude that $\pi(w) \neq \pi(w')$.
2. If $x_0 = x_0'$ and $x_1 = x_1'$, then we must have $x_0 = y_0 = x_0'$ and $z \neq z'$ since $w \neq w'$; using Lemma 20, we conclude that $\pi(w) \neq \pi(w')$.
3. Lastly, suppose that $x_0 \neq x_0'$. If we consider the restriction of π to the inputs $y_1 = 1$ and $i = 0$, we can apply Lemma 20 to conclude that $\pi(w) \neq \pi(w')$. ◀

Using this lower bound, we show one can approximately compute round-d complexity given an oracle that approximately computes round-$(d+1)$ complexity. We consider the following notion of approximation.

▶ **Definition 22.** *For every constant $\epsilon > 0$, we say that an oracle $\mathcal{O}$ is a $(1+\epsilon)$-approximation of a function $\mathcal{L}(\cdot)$ if there exists a constant c such that, for all g in the domain of $\mathcal{L}(\cdot)$,*

$$\mathcal{L}(g) \leq \mathcal{O}(g) \leq (1 + \epsilon) \cdot \mathcal{L}(g) + c.$$

▶ **Corollary 23.** *Let $0 < \epsilon < \frac{1}{8}$. Given an oracle computing a $(1 + \epsilon)$-approximation of $\mathsf{L}_{d+1}^{\mathrm{A}}(\cdot)$ and a function $f : [a] \times [b] \to \{0, 1\}$, one can deterministically compute a $(1 + 4\epsilon)$-approximation of $\mathsf{L}_d^{\mathrm{B}}(f)$ in time $(ab)^{O(1)}$*

Proof. First, we give the reduction algorithm and then we prove its correctness. Suppose $\mathcal{O}$ is an oracle that computes an approximation of $\mathsf{L}^{\mathrm{A}}_{d+1}(\cdot)$ satisifying, for all functions g,

$$\mathsf{L}^{\mathrm{A}}_{d+1}(g) \le \mathcal{O}(g) \le (1+\epsilon)\mathsf{L}^{\mathrm{A}}_{d+1}(g) + O(1).$$

For a positive integer k, let $F_k : ([k] \times [a]) \times ([k] \times [b] \times \{0,1\} \times \{0,1\}) \to \{0,1\}$ be given by

$$F_k(x_0, x_1; y_0, y_1, z, i) = \begin{cases} \mathsf{XorEq}_k(x_0; y_0, z) & \text{, if } i = 0 \\ f(x_1; y_1) & \text{, if } i = 1. \end{cases}$$

The reduction computes

$$v := \max\{\mathcal{O}(F_k) - 2(1+\epsilon)k : k \in [ab]\},$$

and outputs $v' := (v+2)/(1-2\epsilon)$. It is easy to see that this reduction runs in time $(ab)^{O(1)}$.

To prove the correctness of the reduction, we claim that

$$\mathsf{L}^{\mathrm{B}}_d(f) \le v' \le (1+4\epsilon) \cdot \mathsf{L}^{\mathrm{B}}_d(f) + O(1)$$

for all functions f.

From Lemma 21, we know that for all k

$$\mathcal{O}(F_k) - 2(1+\epsilon)k \le (1+\epsilon)\mathsf{L}^{\mathrm{A}}_{d+1}(F_k) - 2(1+\epsilon)k + O(1) \le (1+\epsilon)\mathsf{L}^{\mathrm{B}}_d(f) + O(1)$$

so we have that

$$v' = \frac{v+2}{1-2\epsilon} \le \frac{1+\epsilon}{1-2\epsilon} \cdot \mathsf{L}^{\mathrm{B}}_d(f) + O(1) \le (1+4\epsilon) \cdot \mathsf{L}^{\mathrm{B}}_d(f) + O(1),$$

where the last inequality holds because $\epsilon < 1/8$.

On the other hand, if $k = \mathsf{L}^{\mathrm{B}}_d(f)$, we have from Lemma 21 that

$$\begin{aligned} \mathcal{O}(F_k) - 2(1+\epsilon)k &\ge \mathsf{L}^{\mathrm{A}}_{d+1}(F_k) - 2(1+\epsilon)k \\ &\ge \min\{4k, 2k - 2 + \mathsf{L}^{\mathrm{B}}_d(f)\} - 2(1+\epsilon)k \\ &= \min\{4\mathsf{L}^{\mathrm{B}}_d(f), 3\mathsf{L}^{\mathrm{B}}_d(f) - 2\} - 2(1+\epsilon)\mathsf{L}^{\mathrm{B}}_d(f) \\ &= 3\mathsf{L}^{\mathrm{B}}_d(f) - 2 - 2(1+\epsilon)\mathsf{L}^{\mathrm{B}}_d(f) \\ &\ge (1-2\epsilon)\mathsf{L}^{\mathrm{B}}_d(F_k) - 2. \end{aligned}$$

Since $k = \mathsf{L}^{\mathrm{B}}_d(f) \le ab$, we conclude that $v' = (v+2)/(1-2\epsilon) \ge \mathsf{L}^{\mathrm{B}}_d(F_k)$. ◀

Combining Corollary 23 with the hardness result for the $d = 3$ case in Theorem 18, we get that computing L_d is NP-hard (under a polynomial-time truth-table reduction).

▶ **Corollary 24.** *For any integer $d \ge 3$, there exists an $\epsilon > 0$ such that given access to an oracle that computes a $(1+\epsilon)$-approximation of $\mathsf{L}^{\mathrm{A}}_d$, one can compute any language in NP in polynomial time.*

6 Hardness for randomized 3-round protocols

In this section we prove that it is NP-hard to distinguish whether a function having short deterministic 3-round communication protocols, from a function needs long randomized 3-round protocols with a small error.

▶ **Definition 25.** *The (normalized) Hamming distance of two strings $m_1, m_2 \in \{0,1\}^c$, denoted $\Delta(m_1, m_2)$ is the fraction of bit-positions where m_1 and m_2 differ.*

▶ **Definition 26.** *Let $r, c \in \mathbb{N}$, and let $M \in \{0,1\}^{r \times c}$ be a matrix whose columns are $M^{(1)}, \ldots, M^{(c)}$. Then M is called a (t, k, ε)-gadget if for every set $S = \{s_1, \ldots, s_t\} \subseteq [c]$ of t (distinct) columns of M, the matrix*

$$M_S = \begin{bmatrix} M^{(s_1)} & \ldots & M^{(s_t)} \end{bmatrix}$$

has at least k rows which are pairwise ε-far in Hamming distance. Meaning, there are k rows $m_1, \ldots, m_k \in \{0,1\}^t$ of M_S such that $\Delta(m_i, m_j) \geq \varepsilon$ for all $i, j \in [k]$, $i \neq j$.

▶ **Lemma 27.** *Let $10 \log r \leq c \ll 2^{\frac{r}{10}}$; then a uniformly random matrix $M \in \{0,1\}^{r \times c}$ is a $(10 \log r, \frac{2}{3} r, 1/10)$-gadget.*

Proof. The proof is a standard use of the probabilistic method [13], but let us check the parameters. We choose each entry of M uniformly at random, and we wish to prove that M is a $(t, k, 1/10)$-gadget with high probability. Fix any set S of $t = 10 \log r$ columns – there are $\binom{c}{t}$ many such sets. Imagine we choose r rows sequentially, uniformly at random. Whenever we pick a new row, we call "good" row if it is $\frac{1}{10}$-far, in Hamming distance, of any of the previously picked rows. If this does not hold, we call the row "bad". We wish to upper-bound the probability of seeing fewer than $k = \frac{2}{3} r$ good rows. Notice that the number of t-bit strings in the $\frac{1}{10}$-Hamming-ball around the rows we have seen so far, is less than

$$p = r \binom{t}{\leq t/10} 2^{\frac{t}{10}} \leq r \cdot 2^{(H_2(\frac{1}{10}) + \frac{1}{10}) \cdot t} \leq r \cdot 2^{0.57 \cdot t} \leq r^7,$$

where $H_2(p)$ is the binary entropy function. So the probability of seeing another bad row is less than $p \cdot 2^{-t} \leq r^{-3}$. Hence, the probability of seeing more than $\frac{r}{3}$ bad rows, is less than $(r^{-3})^{\frac{r}{3}} = r^{-r}$. It then follows by a union bound that M will fail to be a $(t, k, \frac{1}{10})$-gadget, with probability no greater than

$$\binom{c}{t} r^{-r}$$

which is close to 0 provided that $c \ll 2^{\frac{r}{10}}$. ◀

We now make the observation that it is possible for a constant-depth to decide, in an approximate sense, whether a given matrix is a good enough gadget. Since there exists an explicit pseudorandom generator for AC_0 with polylog seed length [52], a good gadget can be found in deterministic quasi-polynomial time.

▶ **Corollary 28.** *Let $10 \log r \leq c \ll 2^{\frac{r}{11}}$. One can obtain a matrix $M \in \{0,1\}^{r \times c}$ which is a $(10 \log r, \frac{2}{3} r, 1/10)$-gadget, via a deterministic algorithm running in time $2^{\mathsf{polylog}(r \cdot c)}$.*

Proof. We will show that there exists a constant-depth circuit C of size quasipolynomial in r and c, having $r \times c$ inputs, with the property that every matrix $M \in \{0,1\}^{r \times c}$ with $C(M) = 1$ is a $(10 \log r, \frac{2}{3} r, 1/10)$-gadget, and such that $\Pr[C(M) = 1] = 1 - o(1)$. Corollary 28 follows because there exists an explicit pseudorandom generator $G = \{G_{rc} \colon \{0,1\}^{(\log rc)^{O(1)}} \to \{0,1\}^{r \times c}\}_{r,c \in \mathbb{N}}$ for AC_0 [52].

The circuit checks that every set S of $t = 10 \log r$ columns has at least $\frac{2}{3} r$ good rows, in the same sense as described in the proof of Lemma 27. This is strong enough to ensure that the input is a $(t, k, \frac{1}{10})$-gadget, and it suffices to present a quasipolynomial-size circuit to check this property, since the number of such sets S is itself quasipolynomial.

We cannot check this property exactly, but we can check this property approximately. Using approximate counting [2, 3], a polynomial-size constant-depth circuit D may, given two strings $x, y \in \{0,1\}^t$, give us $D(x,y) = 1$ if $\Delta(x,y) \geq \frac{1}{10} + \varepsilon$, and $D(x,y) = 0$ if $\Delta(x,y) \leq \frac{1}{10}$, where $\varepsilon > 0$ can be chosen to be any arbitrarily small constant. A row x_i will be called *good* if $D(x_i, x_j) = 1$ for all previous rows x_j with $j < i$. Again using approximate counting, and letting M_S denote the sub-matrix of M restricted to the columns in S, we may construct a circuit T_S with $T_S(M) = 1$ if at least $(\frac{2}{3} + \varepsilon)r$ of the rows of M_S are good, and with $T_S(M) = 0$ if fewer than $\frac{2}{3}r$ of the rows of M_S are good. The extra ε will still allow for the previous probability bounds to hold, and if $T_S(M) = 1$ for all S, then M is a $(t, k, \frac{1}{10})$-gadget. ◀

We now show that a simple lower bound on two round communication complexity.

▶ **Lemma 29.** *If $M \in \{0,1\}^{r \times c}$ is a matrix containing r rows $x_1, \ldots, x_r \in \{0,1\}^c$, all pairs of which are ε-far in Hamming distance, then for $\delta = \frac{\varepsilon}{8}$*

$$\mathsf{L}^{\mathrm{A}}_{2,\delta}(M) \geq \frac{1}{2}r,$$

and the hard distribution witnessing this is uniform over the rows and columns of M.

Proof. Let us take an arbitrary matrix $M \in \{0,1\}^{r \times c}$, and think about its $\mathsf{L}^{\mathrm{B}}_1$-complexity. We first observe that the $\mathsf{L}^{\mathrm{B}}_1$-protocol for M which has the smallest possible error is the single-bit protocol where Bob sends the majority of his column to Alice, meaning, he sends 1 if and only if half or more of the entries in his column are 1. Hence, the smallest error which a deterministic $\mathsf{L}^{\mathrm{B}}_1$-protocol can make when computing M under the uniform distribution is precisely the error of this smallest-error protocol, which is

$$\mathsf{Err}(M) = \min_{z \in \{0,1\}^c} \frac{1}{r} \sum_{i=1}^{r} \Delta(x_i, z).$$

Indeed, the z giving the minimum is the column-wise majority of M.

Now, suppose $\Delta(x_i, x_j) \geq \varepsilon$ for all $i, j \in [r]$, $i \neq j$. If we have an $\mathsf{L}^{\mathrm{A}}_2$-protocol π for M which partitions the rows of F into fewer than $\frac{r}{2}$ parts, then there must exist $\frac{r}{2}$ rows of M which get placed in a part that contains at least one other row of M. If we have a part which has p rows placed together, then by the triangle inequality this means that, for any $z \in \{0,1\}^c$, we must have $\Delta(x_i, z) \geq \frac{\varepsilon}{2}$ for at least $p - 1$ values of i (if z is $\frac{\varepsilon}{2}$-close to one of the x_i, it must be $\frac{\varepsilon}{2}$-far from all other x_i in the same part, since they are pairwise distant). Hence if M' is any part of M in the partition, having more than one row, we have

$$\mathsf{Err}(M') \geq \frac{p-1}{p} \frac{\varepsilon}{2} \geq \frac{\varepsilon}{4}$$

But since 1/2 of the rows get placed together with other rows, the total error incurred by π on M is at least $\frac{\varepsilon}{8}$. ◀

We now show the following hardness result.

▶ **Theorem 30.** *Let $0 < \delta < 1$ be given. Given an undirected graph $G = ([n], E)$ with n vertices and $|E| = m > 0$ edges, one may construct in deterministic quasipolynomial time a function $f_G : [a] \times [b] \to \{0,1\}$ and a number $\ell \in \mathbb{N}$, with $a, b, \ell = O(n^{27})$, such that*

- $\mathsf{L}^{\mathrm{B}}_{3,n^{-\delta}}(f_G) = \Omega(n^{\frac{\delta}{16}-1} \cdot \chi(G) \cdot \ell)$,
- $\mathsf{C}^{\mathrm{B}}_{3,n^{-\delta}}(f_G) \geq \frac{\delta}{16} \log n + \log \chi(G) - \log n + \log \ell - O(1)$,
- $\mathsf{L}^{\mathrm{B}}_3(f_G) = O(\chi(G) \cdot \ell)$, *and*
- $\mathsf{C}^{\mathrm{B}}_3(f_G) = \log \chi(G) + \log \ell + O(1)$.

Proof. The construction is the same as in Theorem 17, but where use (t, k, ε)-gadgets instead of universal sets, and Lemma 29 instead of Proposition 11. The function f_G is defined exactly as in the proof of Theorem 17, but we use the gadgets from Corollary 28, namely, we set $\ell = m^4n^4$ and $c = n^2\ell^2$, and we choose $A \in \{0,1\}^{r\times c}$ to be an $(O(\log r), \frac{2}{3}r, 1/10)$-gadget, and we choose $B, C \in \{0,1\}^{s\times c}$ to be an $(O(\log s), \frac{2}{3}s, 1/10)$, where $r = \ell = m^4n^4$ and $s = m^3n^3$. This choice obeys the two conditions:

1. Condition 1. $r + ms = O(\ell)$.
2. Condition 2. $\sqrt{c/2} \geq n \cdot \ell$.

The upper-bound is given by the same protocol as in the proof of Theorem 17, where Condition 1 gives us improved parameters. We are left to prove the lower-bound. This is proven via Yao's principle. The hard distribution μ for f_G is chosen as follows:

- With probability 1/2 we will let the input (x, y) be a uniformly chosen row x among the first r rows, and a uniformly chosen column. I.e. a uniform entry of the $[A \dots A]$ sub-matrix of f_G.
- And with probability 1/2 we choose an edge $\{v, w\}$ uniformly at random from the edges of G, and then choose a uniform entry among the B and C sub-matrices of the $M_{\{u,v\}}$ sub-matrix of f_G.

Now suppose we are given a deterministic $\mathsf{L}_3^{\mathsf{B}}$-protocol π with L leaves, which computes f_G with error $\leq n^{-\delta}$ under the distribution μ. We will then show that one of two things must happen: either (1) π has $L \gg n\ell$ leaves, or (2) π gives us a coloring of G with $\leq \frac{3}{\ell}n^{1-\frac{1}{16}\delta}L$ colors. In both cases it must follow that $L = \Omega(n^{\frac{\delta}{16}-1} \cdot \chi(G) \cdot \ell)$.

Indeed, we will show that either (1) π has $L \gg n\ell$ leaves, or (2') π gives us a coloring of a graph G', which has $\leq \frac{1}{\ell}L$ colors, where G' is obtained from G by removing $\leq n^{-\frac{1}{8}\delta}|E|$ edges. We will then make use of the following:

▷ Claim 31. If G' is obtained from G by removing $\leq n^{-\delta} \cdot |E|$ edges, then any coloring of G' with C colors will induce a coloring of G with $3n^{1-\frac{\delta}{2}}C$ colors.

Proof. Let $N = n^{-\delta/2} \cdot n$. Split the vertices of G into two sets: V_1 contains those vertices of G where we have removed $\geq N$ edges, and V_2 contains the remaining vertices. We have that $|V_1| \leq 2N$, otherwise too many edges would have been removed.

So let $\alpha' : [n] \to [C]$ be a C-coloring of G'. We then construct a coloring $\alpha : [n] \to [2N + C \cdot (N + 1)]$, as follows. We first color each vertex of V_1 by its own color. Then we greedily color each vertex $v \in V_2$ by a color $\alpha(v) = (\alpha'(v), \beta(v))$, such that the second coordinate $\beta(v) \in [N+1]$ does not appear as the second coordinate of any neighbours $w \in V_2$ of v which we have already colored. There will always exist such a β because the number of new neighbors of v, when going from G' to G, is $\leq N$. This is a coloring of G with $\leq 3C \cdot N$ colors, as promised. ◁

Now, look at the marginal distribution of μ over the columns of f_G. Then each block of columns of f_G corresponding to a vertex v gets probability mass exactly:

$$\frac{1}{2n} + \frac{1}{2}\frac{\deg(v)}{2|E|}. \qquad (*)$$

Let us now remove *high-error columns*, as follows. We first remove from f_G all column blocks v where the error probability of π, conditioned on Bob's input being a column of v, is greater than $n^{-\delta/2}$. Since the error probability of π is $\leq n^{-\delta}$, then by Markov's inequality, the total probability mass removed in this way is less than $n^{-\delta/2}$. By $(*)$, removing all such vertices v from G will remove fewer than $n^{-\delta/2} \cdot 4|E| \leq 4n^{-\delta/2}n$ edges in total.

Now we do similarly inside each block. For each surviving column block v we know that the error probability inside it (i.e. conditioned on getting a column inside the block) is $\leq n^{-\delta/2}$. Let us then remove every column y where the error probability of π, conditioned on Bob's input being y, is greater than $2n^{-\delta/2}$. Notice that, within each block, every column gets the same probability. Hence, again by Markov's inequality, by removing these high-error y we have removed fewer than $\frac{1}{2}c$ columns from each block. We are left with a sub-matrix of f_G where, in each column, π has error probability less than $n^{-\delta/4}$, and where each surviving column block v has $\geq \frac{1}{2}c$ columns.

We now remove some further columns, which we will call *leftover columns*. To begin, we remove enough columns from each surviving block so that there are exactly $\frac{1}{2}c$ columns in each block. We do this so we don't have to think about having a different number of surviving columns among different blocks.

Now let $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ be the partition of the surviving columns of f_G which is induced by the first round of π. If $|\mathcal{P}| \geq \sqrt{c/2} \gg n\ell$ (by Condition 2), then we have established (1). Otherwise, we must show (2'). If $|\mathcal{P}| \leq \sqrt{c/2}$, then for each column block v there exists a part $P_{i(v)}$ of $\mathcal{P}$ having at least $\sqrt{c/2}$ columns from the v-th block. Let us then remove from $P_{i(v)}$ more columns from the v-th block, so that, for every surviving v, $P_{i(v)}$ always contains exactly $\sqrt{c/2}$ columns from the v-th block.

We then consider the partition $\mathcal{P}'$ containing exactly the parts $P_{i(v)}$ for surviving v, but without any of the columns we have removed thus far, namely, without the high-error columns and without the leftover columns. Let f'_G equal to f_G, but restricted to the surviving columns. Since the error probability of π was $\leq 2n^{-\delta/2}$ on every surviving column, then π will still have error probability $\leq 2n^{-\delta/2}$ on f'_G.

We now remove high-error rows. We first remove each row-block $M_{\{v,w\}}$ such that the error of π on $M_{\{v,w\}}$ is greater than $2n^{-\delta/4}$. Again by Markov's inequality, in doing so we remove $\leq n^{-\delta/4}|E|$ more edges.

Now let E' be the set of surviving edges $\{v, w\}$ such that $i(v) = i(w)$ (i.e. E' contains the low-error edges which violate the coloring constraint). Fix any edge $\{v, w\} \in E'$, and let L be the number of leaves in the 2-round sub-protocol inside part $P_{i(v)} = P_{i(w)}$. If $L \geq \frac{1}{2}s^2 = \Omega(n\ell)$, we then have proven that π has $\Omega(n\ell)$ leaves, and we are done; otherwise, suppose $L < \frac{1}{2}s^2$ leaves. Now notice that, by the gadget property, the surviving columns of the B and C sub-matrices of any such block each have $\frac{2}{3}s$ rows which are pairwise $\frac{1}{10}$-distant in Hamming distance; hence the sub-matrix $[BC]$ within $M_{\{u,v\}}$ containing the surviving columns, must have at least $(\frac{2}{3}s)^2$ rows which are $\frac{1}{20}$-distant in Hamming distance. It then follows from Lemma 29 that the probability of error within $M_{\{u,v\}}$ is $\geq \frac{1/20}{8} = \frac{1}{160}$. And this would happen for every edge $\{u, v\} \in E'$.

It then follows that E' is small. Indeed, if we had $|E'| > n^{-\delta/8} \cdot |E|$, then the total error of π on the surviving sub-matrix would be $\geq \Omega(n^{-\delta/8})$, and since this sub-matrix has $\Omega(1)$ of the total mass of the original matrix, this would contradict π's claimed overall error bound. So we are forced to conclude that $|E'| \leq n^{-\delta/8} \cdot |E|$.

We may then remove all the sub-matrices $M_{\{v,w\}}$ corresponding to edges $\{v, w\} \in E'$. It then follows that the partition $\mathcal{P}'$ is a coloring of the resulting sub-graph G'. Hence $|\mathcal{P}'| \geq \chi(G')$, which by Claim 31 means $|\mathcal{P}'| \geq n^{\frac{\delta}{16}-1}\chi(G)$. Now, within each part P_i of $\mathcal{P}'$, the corresponding A sub-matrix still needs to be solved by an $\mathsf{L}_2^{\mathsf{A}}$-protocol with error $\leq 2n^{-\delta/4}$, which can only be done with ℓ leaves, again by Lemma 29. Hence the total number of leaves is $\Omega(n^{\frac{\delta}{16}-1}\chi(G)\ell)$. ◀

We can then improve upon Theorem 18, and prove that it is NP-hard, under quasipolynomial-time reductions, to distinguish whether a given communication matrix has small deterministic communication complexity, versus large low-error randomized communication complexity. In the next section, we will show that a small improvement of the parameters of the following corollary[4] would be enough to show strong hardness-of-approximation for any number of rounds.

▶ **Corollary 32.** *There exist positive constants γ and δ such that the following holds. There exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N \times N}$ and a number $k \in \mathbb{N}$, with $k \leq N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of* SAT*, then $\mathsf{L}_3^{\mathrm{B}}(M) \leq O(k)$ and $\mathsf{C}_3^{\mathrm{B}}(M) \leq \log k + O(1)$, and*
2. *if x is a NO instance of* SAT*, then $\mathsf{L}_{3,N^{-\delta}}^{\mathrm{B}}(M) \geq \Omega(N^{\gamma} \cdot k)$ and $\mathsf{C}_{3,N^{-\delta}}^{\mathrm{B}}(M) \geq \log k + \gamma \cdot \log N - O(1)$.*

Proof. We will choose $\delta_3 = \delta > 0$ to be a sufficiently small constant. We combine the two reductions of Theorems 9 and 30, which we invoke with parameter $\varepsilon = \delta/64$. Fix any input x and let G be an n-vertex graph that is produced by the reduction of Theorem 9 on input x. We have $n = |x|^{O(1)}$ since the reduction of Theorem 9 is polytime. Let $M \in \{0,1\}^{a \times b}$ be the communication matrix of f_G, and $\ell \in \mathbb{N}$, be given by the reduction of Theorem 17 on input G, and set $N = \max(a,b) = O(n^{27})$, $k = n^{\varepsilon} \cdot \ell$. We may assume that $a = b$ without loss of generality, since otherwise we may pad M with all-0 rows or all-0 columns, and this changes the leaf complexity by at most a factor of 2, and the communication complexity by at most 1 bit, while leaving the error parameter intact.

We verify that the inequalities are satisfied for M: If $x \in L$, then we have $\mathsf{L}_3^{\mathrm{B}}(M) = O(\chi(G) \cdot \ell) = O(k)$. If $x \notin L$, then we have

$$\begin{aligned}
\mathsf{L}_{3,n^{-\delta}}^{\mathrm{B}}(M) &= \Omega(n^{\frac{\delta}{16}-1}\chi(G) \cdot \ell) \\
&= \Omega(n^{\frac{\delta}{16}-\varepsilon} \cdot \ell) \\
&= \Omega(n^{\frac{\delta}{16}-2\varepsilon} \cdot k) \\
&= \Omega(n^{\frac{\delta}{32}} \cdot k) \\
&= \Omega(N^{\frac{\delta}{32 \times 27}} \cdot k)
\end{aligned}$$

Similar inequalities hold for $\mathsf{C}_3^{\mathrm{B}}(M)$. So we set $\gamma = \frac{\delta}{32 \times 27}$. ◀

7 From 3-rounds to multiple rounds using round elimination

We would now like to prove that constant-round communication complexity is NP-hard, for any number of rounds. However, the result we proved in Corollary 32 is not enough. We conjecture that the parameters in that result can be improved, as follows

▶ **Conjecture 33.** *For any constant $C \geq 1$, there exist positive constants $\gamma, \delta \in (0,1]$ with $\gamma \geq C \cdot \delta$ such that the following holds. There exists a deterministic quasipolynomial-time algorithm that, on input $x \in \{0,1\}^*$, outputs a communication matrix $M \in \{0,1\}^{N \times N}$ and a number $k \in \mathbb{N}$, with $k \leq N = |x|^{O(1)}$, such that*

1. *if x is a YES instance of* SAT*, then $\mathsf{L}_3^{\mathrm{B}}(M) \leq O(k)$ and $\mathsf{C}_3^{\mathrm{B}}(M) \leq \log k + O(1)$, and*
2. *if x is a NO instance of* SAT*, then $\mathsf{L}_{3,N^{-\delta}}^{\mathrm{B}}(M) \geq \Omega(N^{\gamma} \cdot k)$ and $\mathsf{C}_{3,N^{-\delta}}^{\mathrm{B}}(M) \geq \log k + \gamma \cdot \log N - O(1)$.*

[4] As we will see, it would be enough if γ could be made arbitrarily larger than δ.

Let us devise notation that will help us better understand the difference. We may now define the following problems:

▶ **Definition 34.** *In the problem* $\mathsf{MPL}^{\mathrm{A}}(d, \varepsilon, \phi, N)$*, defined for each natural number* $d \geq 3$*, all* $\varepsilon \in [0, 1]$*,* $\phi \geq 1$*, and* $N \in \mathbb{N}$*, we are given as input an* $N \times N$ *Boolean matrix* M*, and a natural number* $1 \leq k \leq N$*, with the promise that*

- *either* $\mathsf{L}^{\mathrm{A}}_{d}(M) \leq k$*,*
- *or* $\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \geq \phi \cdot k$

and we wish to decide which is the case. We define $\mathsf{MPL}^{\mathrm{B}}$ *in the same way.*

In the problem $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi, K, N)$*, defined for each natural number* $d \geq 3$*, all* $\varepsilon, \phi \in [0, 1]$*, and all* $K, N \in \mathbb{N}$*, we are given as input an* $N \times N$ *Boolean matrix* M*, and a natural number* $1 \leq k \leq K$*, with the promise that*

- *either* $\mathsf{L}^{\mathrm{A}}_{d}(M) \leq k$*,*
- *or* $\mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \geq k + \phi \cdot K$

and we wish to decide which is the case. We define $\mathsf{MPC}^{\mathrm{B}}$ *analogously.*

Then Corollary 32 and Conjecture 33 tell us that these approximation problems are NP-hard, for different parameters. In this notation we may restate Corollary 32 and Conjecture 33 as follows:

- (Corollary 32) There exist positive constants γ and δ such that $\mathsf{MPL}^{\mathrm{A}}(3, N^{-\delta}, N^{\gamma}, N)$ and $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ are NP-hard under deterministic quasipolynomial-time many-one reductions.
- (Conjecture 33) For any constant $C \geq 1$, there exist positive constants $\gamma, \delta \in (0, 1]$ with $\gamma \geq C \cdot \delta$ such that, $\mathsf{MPL}^{\mathrm{A}}(3, N^{-\delta}, N^{\gamma}, N)$ and $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ are NP-hard under deterministic quasipolynomial-time many-one reductions.

In the rest of this section, we will use Conjecture 33 and the round elimination lemma to prove that quasipolynomial-time algorithms for computing constant-round communication complexity would place all of NP in subexponential time.

We begin by recalling the round-elimination lemma, which was originally proven by Miltersen, Nisan, Safra, and Wigderson [46] and later improved and simplified by Sen and Venkatesh [62], using an information-theoretic argument. Sen and Venkatesh's proof is already information-theoretic in flavor, but it can be made significantly shorter by using a nowadays-standard combination of the chain rule and Pinsker's inequality (see [15, 58]). We include this shortened proof here, on the one hand so that we can confirm that it works for leaf complexity, and not just communication complexity, and on the other hand so we can extract the exact parameters.

▶ **Theorem 35** (Round Elimination Lemma [46, 62])**.** *Let* $3 \leq d \in \mathbb{N}$ *and let* $\alpha > 0$*. Given a Boolean function* $f : [a] \times [b] \to \{0, 1\}$ *and a parameter* $\beta > 0$*, we may construct a Boolean function* $F : [a]^m \times ([m] \times [b]) \to \{0, 1\}$*, with* $m = \frac{1}{4\beta^2}(\lceil \log \min(a, b) \rceil + 1)$*, such that*

$$\mathsf{L}^{\mathrm{A}}_{d+1}(F) \leq m \cdot \mathsf{L}^{\mathrm{B}}_{d}(f), \qquad \mathsf{L}^{\mathrm{A}}_{d+1,\alpha}(F) \leq m \cdot \mathsf{L}^{\mathrm{B}}_{d,\alpha}(f), \qquad \mathsf{L}^{\mathrm{B}}_{d,\alpha}(f) \leq \mathsf{L}^{\mathrm{A}}_{d+1,\alpha-\beta}(F),$$

and also

$$\mathsf{C}^{\mathrm{A}}_{d+1}(F) \leq \mathsf{C}^{\mathrm{B}}_{d}(f) + \lceil \log m \rceil \qquad \mathsf{C}^{\mathrm{A}}_{d+1,\alpha}(F) \leq \mathsf{C}^{\mathrm{B}}_{d,\alpha}(f) + \lceil \log m \rceil \qquad \mathsf{C}^{\mathrm{B}}_{d,\alpha}(f) \leq \mathsf{C}^{\mathrm{A}}_{d+1,\alpha-\beta}(F).$$

Proof. We define $F(x_1, \ldots, x_m; i, y) = f(x_i; y)$. Meaning, Alice is given m Alice-side inputs $x_1, \ldots, x_m \in [a]$ for f, and Bob is given an index $i \in [m]$ and a Bob-side input $y \in [b]$ for f, and they wish to compute $f(x_i; y)$.

The upper-bounds on $\mathsf{L}^{\mathrm{A}}_{d+1}$ and $\mathsf{C}^{\mathrm{A}}_{d+1}$ are simple to see. Indeed, a $d+1$ round protocol where Alice begins to speak may have Alice send nothing in her first round, after which Bob sends i to Alice and then the players may compute $f(x; y_i)$ by executing a d-round protocol for f. This works whether or not the protocol for f is randomized.

Now to prove the lower-bounds on $\mathsf{L}^{\mathrm{A}}_{d+1}$ and $\mathsf{C}^{\mathrm{A}}_{d+1}$. To prove the lower-bound on the leaf complexity, we may assume that $\mathsf{L}^{\mathrm{A}}_{d+1,\alpha-\beta}(F) \leq 2\min(a,b)$, and to prove the lower-bound on the communication complexity, we may assume that $\mathsf{C}^{\mathrm{A}}_{d+1,\alpha-\beta}(F) \leq \lceil \log \min(a,b) \rceil + 1$, since otherwise the respective inequalities are trivial. Let μ be the hard distribution for f. Construct a distribution μ' for F by choosing $i \in [m]$ uniformly at random, sampling (x_i, y) according to μ, and then sampling each x_j with $j \neq i$ from the x-marginal of μ. Now suppose we are given a deterministic Alice-first $(d+1)$-round protocol π' for F with $\mathsf{L}(\pi') \leq 2\min(a,b)$ leaves and communication complexity $\mathsf{C}(\pi') \leq \lceil \log \min(a,b) \rceil + 1$, and which makes $\alpha - \beta$ error (or less) when the input is sampled according to μ', and let us construct a deterministic Bob-first d-round protocol π for f having $\mathsf{L}(\pi) \leq \mathsf{L}(\pi')$ leaves and communication complexity $\mathsf{C}(\pi) \leq \mathsf{C}(\pi')$, which makes α error when the input is sampled according to μ.

Let $(\mathbf{x}_1, \ldots, \mathbf{x}_m; \mathbf{i}, \mathbf{y})$ denote random variables sampled according to the distribution μ', and let $\mathbf{t} = \mathbf{t}(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be the message sent in the first round of π'. Let T denote either $\log \mathsf{L}(\pi')$ or $\mathsf{C}(\pi')$, so that $T \leq \lceil \log \min(a,b) \rceil + 1$. We then have, by the chain rule,

$$T \geq \mathsf{H}(\mathbf{t}) \geq \mathsf{I}(\mathbf{x}_1, \ldots, \mathbf{x}_m : \mathbf{t}) = \sum_{i=1}^{m} \mathsf{I}(\mathbf{x}_i : \mathbf{t} \mid \mathbf{x}_{<i}) = m \cdot \mathsf{I}(\mathbf{x}_{\mathbf{i}} : \mathbf{t} \mid \mathbf{i}, \mathbf{x}_{<\mathbf{i}})$$

Hence there exists a setting $\mathbf{i} = i$ such that

$$\mathsf{I}(\mathbf{x}_{\mathbf{i}} : \mathbf{t} \mid \mathbf{i} = i, \mathbf{x}_{<\mathbf{i}}) \leq \frac{1}{m} T \leq 2\beta^2.$$

Let us then fix some setting $\mathbf{x}_{<i} = x_{<i}$ which attains the above bound, so that

$$\mathsf{I}(\mathbf{x}_{\mathbf{i}} : \mathbf{t} \mid \mathbf{i} = i, \mathbf{x}_{<\mathbf{i}} = x_{<i}) \leq 2\beta^2$$

By Pinsker's inequality, this implies that the average, over choices t for $\mathbf{t}$, statistical distance between $\mathbf{x}_i$ and $\mathbf{x}_i$ conditioned on $\mathbf{t} = t$, is at most β. On the other hand, the average error (of π on μ') over choices of t for $\mathbf{t}$ is at most $\alpha - \beta$. By linearity of expectation, the average sum of the error plus statistical distance is at most α. Let us then fix a choice t for $\mathbf{t}$ where this sum is at most α.

We may then consider the Bob-first d-round protocol $\tilde{\pi}$ that executes the last d-rounds of π, for the case when the first message of π is t, where $\mathbf{x}_{<i}$ has been fixed to $x_{<i}$, and each coordinate of $\mathbf{x}_{>i}$ is chosen at random from the x-marginal of μ. Such a protocol will incur a total error $\leq \alpha$, and has fewer leaves ans smaller communication complexity than π'. ◀

The round-elimination lemma can be used to reduce the computation of complexity on d rounds to the computation of complexity on $d+1$ rounds, as follows.

▶ **Corollary 36.** *We may reduce* $\mathsf{MPL}^{\mathrm{A}}(d, \varepsilon, \phi, N)$ *to* $\mathsf{MPL}^{\mathrm{A}}(d+1, \frac{\varepsilon}{2}, \frac{\phi}{m}, N^m)$*, and we may reduce* $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi, K, N)$ *to* $\mathsf{MPC}^{\mathrm{A}}(d+1, \frac{\varepsilon}{2}, \phi - 2\frac{\log m}{K}, K + \log m, N^m)$*, where* $m = 32\varepsilon^{-2} \log N$*, by a many-one reduction computable in time* $N^{O(m)}$*.*

Proof. The reduction is given an $N \times N$ communication matrix M, which corresponds to a Boolean function $f : [N] \times [N] \to \{0,1\}$, and a number $k \leq N$. Let F given by Theorem 35, with parameters $m = 32\varepsilon^{-2} \log N$, $\alpha = \varepsilon$ and $\beta = \frac{\varepsilon}{2}$. Then the output is a matrix $M' \in \{0,1\}^{N' \times N'}$ where $N' \leq N^m$, obtained from the communication matrix of F padded with extra 0-columns to make it into a square matrix.

Then if $\mathsf{L}^{\mathrm{A}}_{d}(M) \le k$, we will also have $\mathsf{L}^{\mathrm{A}}_{d+1}(M') \le mk$. On the other hand, if $\mathsf{L}^{\mathrm{A}}_{d,\alpha}(M) = \mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \ge \phi \cdot k$, then $\mathsf{L}^{\mathrm{A}}_{d+1,\frac{\varepsilon}{2}}(M') = \mathsf{L}^{\mathrm{A}}_{d+1,\alpha-\beta} \ge \mathsf{L}^{\mathrm{A}}_{d,\varepsilon}(M) \ge \phi \cdot k = \frac{\phi}{m} \cdot mk$.

Furthermore if $\mathsf{C}^{\mathrm{A}}_{d}(M) \le k$, we will also have $\mathsf{C}^{\mathrm{A}}_{d+1}(M') \le k + \log m$. On the other hand, if $\mathsf{C}^{\mathrm{A}}_{d,\alpha}(M) = \mathsf{C}^{\mathrm{A}}_{d,\varepsilon}(M) \ge k + \phi \cdot K$, then $\mathsf{C}^{\mathrm{A}}_{d+1,\frac{\varepsilon}{2}}(M') = \mathsf{C}^{\mathrm{A}}_{d+1,\alpha-\beta} \ge \mathsf{C}^{\mathrm{A}}_{d,\varepsilon}(M) \ge k + \phi \cdot K$, and

$$\begin{aligned} k + \phi \cdot K &= k + \log m + \left(\phi - \frac{(1+\phi)\log m}{K + \log m}\right) \cdot (K + \log m) \\ &\ge k + \log m + \left(\phi - 2\frac{\log m}{K}\right) \cdot (K + \log m). \end{aligned}$$ ◀

We can now show that if communication complexity C^{A} (and not just leaf complexity L^{A}) can be computed in quasipolynomial time and Conjecture 33 holds, then all of NP can be solved in subexponential time, i.e., time $2^{n^{\varepsilon}}$, for any choice $\varepsilon > 0$. A similar result can be proven for leaf complexity, with better hardness-of-approximation than what is obtained in Section 5, but we will omit the proof here, because it is very similar, and we already have the results of Section 5. If the error parameter in Conjecture 33 can be made into a constant, instead of $N^{-\delta}$, then the same proof will give us quasipolynomial NP-hardness instead of subexponential. We also omit this proof.

▶ **Theorem 37.** *If Conjecture 33 holds and* C^{A} *can be computed in quasipolynomial time, then all of* NP *can be computed in subexponential time.*

Proof. Conjecture 33 says that a SAT instance of size n reduces to $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ with $N = n^{O(1)}$, for any choice of γ, δ, where δ can be chosen to be arbitrarily small, and γ can be chosen to be as many times higher than δ as needed. We may now apply Corollary 36 repeatedly. We are only aiming for rough parameters, and so for simplicity, in the first application, we replace $\log N$ factors with N^{δ}, and in all applications, we replace the 32 factor with N^{δ}, as well. We can do this because $N^{\delta} \gg \log N$, and $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi, K, N)$ reduces to $\mathsf{MPC}^{\mathrm{A}}(d, \varepsilon, \phi', K, N')$ whenever $\phi' \le \phi$ and $N' \ge N$.

We then find that $\mathsf{MPC}^{\mathrm{A}}(3, N^{-\delta}, \gamma, \log N, N)$ reduces to

$$\mathsf{MPC}^{\mathrm{A}}(4, \frac{1}{2}N^{-\delta}, \gamma - 8\delta, (1+4\delta)\log N, 2^{N^{5\delta}}),$$

which reduces to

$$\mathsf{MPC}^{\mathrm{A}}(5, 2^{-2} \cdot N^{-\delta}, \gamma - 16\delta, 2^{N^{13\delta}}),$$

etc, which reduces to

$$\mathsf{MPC}^{\mathrm{A}}(d, C_1 \cdot N^{-\delta}, \gamma - C_2\delta, (1 + C_3\delta)\log N, 2^{N^{C_4\delta}})$$

for some positive constants C_1, C_2, C_3, C_4 that depend only on df. This problem in turn reduces to computing $\mathsf{C}^{\mathrm{A}}(f)$ exactly, on instances $f : [N'] \times [N'] \to \{0,1\}$, for $N' = 2^{N^{C_4\delta}}$. Now suppose we could compute $\mathsf{C}^{\mathrm{A}}(f)$ exactly in time quasipolynomial in $N' = 2^{N^{C_4\delta}}$, i.e., in time $2^{N^{O(C_4\delta)}}$. Then by choosing δ to be sufficiently small, given that $N = n^{O(1)}$, we could then solve SAT in time $2^{n^{\varepsilon}}$, for any $\varepsilon > 0$ of our choosing. ◀

References

1 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computation Theory*, 1(1):2, 2009.

2 Miklós Ajtai. Σ^1_1-formulae on finite structures. *Annals of pure and applied logic*, 24(1):1–48, 1983.

3 Miklós Ajtai and Michael Ben-Or. A Theorem on Probabilistic Constant Depth Computations. In *Proceedings of the Symposium on Theory of Computing (STOC)*, pages 471–474, 1984.

4 Eric Allender. The new complexity landscape around circuit minimization. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, *Language and Automata Theory and Applications* (LATA), volume 12038 of *Lecture Notes in Computer Science*, pages 3–16. Springer, 2020.

5 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 25–32, 2014.

6 Eric Allender, Joshua A. Grochow, Dieter van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum circuit size, graph isomorphism, and related problems. *SIAM Journal on Computing*, 47(4):1339–1372, 2018.

7 Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM Journal on Computing*, 38(1):63–84, 2008.

8 Eric Allender and Shuichi Hirahara. New insights on the (non-)hardness of circuit minimization and related problems. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 54:1–54:14, 2017.

9 Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. *Computational Complexity*, 26(2):469–496, 2017.

10 Eric Allender, Rahul Ilango, and Neekon Vafa. The non-hardness of approximating circuit size. In *International Computer Science Symposium in Russia* (CSR), pages 13–24, 2019.

11 Eric Allender and Michal Kouckỳ. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57(3):1–36, 2010.

12 Eric Allender, Michal Kouckỳ, Detlef Ronneburger, and Sambuddha Roy. The pervasive reach of resource-bounded Kolmogorov complexity in computational complexity theory. *Journal of Computer and System Sciences*, 77(1):14–40, 2011.

13 Noga Alon and Joel H Spencer. *The probabilistic method.* John Wiley & Sons, 2016.

14 Theodore Baker, John Gill, and Robert Solovay. Relativizations of the p=?np question. *SIAM Journal on computing*, 4(4):431–442, 1975.

15 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *SIAM Journal on Computing*, 42(3):1327–1363, 2013.

16 Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

17 Joan Boyar, Philip Matthews, and René Peralta. On the shortest linear straight-line program for computing linear forms. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 168–179, 2008.

18 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Conference on Computational Complexity* (CCC), pages 10:1–10:24, 2016.

19 Sebastian Lukas Arne Czort. The complexity of minimizing disjunctive normal form formulas. Master's thesis, University of Aarhus, 1999.

20 Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998.

21 Vitaly Feldman. Hardness of approximate two-level logic minimization and PAC learning with membership queries. In *Symposium on Theory of Computing (STOC)*, pages 363–372, 2006.

22 Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. $ac^0[p]$ lower bounds against mcsp via the coin problem. In *Colloquium on Automata, Languages, and Programming* (ICALP), volume 132, page 66, 2019.

23 Thomas R. Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2):114–122, 1996.

24 Johan Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Symposium on Foundations of Computer Science* (FOCS), pages 627–636, 1996.

25 Shuichi Hirahara, Igor Carboni Oliveira, and Rahul Santhanam. NP-hardness of minimum circuit size problem for OR-AND-MOD circuits. In *Computational Complexity Conference* (CCC), pages 5:1–5:31, 2018.

26 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *Conference on Computational Complexity* (CCC), pages 18:1–18:20, 2016.

27 John M. Hitchcock and Aduri Pavan. On the NP-completeness of the minimum circuit size problem. In *Foundation of Software Technology and Theoretical Computer Science* (FSTTCS), pages 236–245, 2015.

28 Rahul Ilango. Approaching MCSP from above and below: Hardness for a conditional variant and $ac^0[p]$. In Thomas Vidick, editor, *Innovations in Theoretical Computer Science Conference* (ITCS), volume 151 of *LIPIcs*, pages 34:1–34:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

29 Rahul Ilango. Constant depth formula and partial function versions of MCSP are hard. In *Proceedings of the Symposium on Foundations of Computer Science* (FOCS), pages 424–433. IEEE, 2020.

30 Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. Np-hardness of circuit minimization for multi-output functions. In Shubhangi Saraf, editor, *Computational Complexity Conference* (CCC), volume 169 of *LIPIcs*, pages 22:1–22:36. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

31 Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich. The Power of Natural Properties as Oracles. In *Computational Complexity Conference* (CCC), volume 102, pages 7:1–7:20, 2018.

32 J. Stephen Judd. Learning in networks is hard. In *International Conference on Neural Networks* (ICNN), volume 2, pages 685–692, 1987.

33 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Symposium on Theory of Computing* (STOC), pages 73–79, 2000.

34 Mauricio Karchmer, Eyal Kushilevitz, and Noam Nisan. Fractional covers and communication complexity. *SIAM Journal on Discrete Mathematics*, 8(1):76–92, 1995.

35 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 539–550. Association for Computing Machinery, 1988.

36 Subhash Khot and Rishi Saket. Hardness of minimizing and learning DNF expressions. In *Symposium on Foundations of Computer Science* (FOCS), pages 231–240, 2008.

37 Jan Krajícek. *Forcing with Random Variables and Proof Complexity*. Cambridge University Press, 2011.

38 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

39 Eyal Kushilevitz and Enav Weinreb. On the complexity of communication complexity. In *Symposium on Theory of Computing* (STOC), pages 465–474, 2009.

40 Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. *arXiv preprint*, 2020. `arXiv:2009.11514`.

41 L. Lovasz and M. Saks. Lattices, mobius functions and communications complexity. In *Symposium on Foundations of Computer Science* (FOCS), SFCS '88, page 81–90, USA, 1988. IEEE Computer Society.

42 Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

43 William J. Masek. Some NP-complete set covering problems. Unpublished Manuscript, 1979.

44 Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Symposium on Theory of Computing* (STOC), STOC 2019, page 1215–1225, New York, NY, USA, 2019. Association for Computing Machinery.

45 Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Symposium on Theory of Computing* (STOC), 2019.

46 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

47 Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2), 2020.

48 Cody D. Murray and Richard Ryan Williams. On the (non) NP-hardness of computing circuit complexity. In *Conference on Computational Complexity* (CCC), pages 365–380, 2015.

49 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM journal on computing*, 22(4):838–856, 1993.

50 Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004.

51 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and Near-Optimal Derandomization. In *Symposium on Foundations of Computer Science* (FOCS), pages 182–191, 1995.

52 Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991.

53 Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. In *Conference on Computational Complexity* (CCC), 2019.

54 Igor Carboni Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Computational Complexity Conference* (CCC), pages 18:1–18:49, 2017.

55 Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *Symposium on Foundations of Computer Science* (FOCS), pages 65–76, 2018.

56 Denis Pankratov. Direct sum questions in classical communication complexity. *Master's thesis, University of Chicago*, 2012.

57 Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

58 Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

59 Alexander A Razborov. On submodular complexity measures. *Boolean Function Complexity,(M. Paterson, Ed.)*, pages 76–83, 1992.

60 Alexander A Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

61 Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

62 Pranab Sen and Srinivasan Venkatesh. Lower bounds for predecessor searching in the cell probe model. *Journal of Computer and System Sciences*, 74(3):364–385, 2008.

63 Boris A Trakhtenbrot. A survey of Russian approaches to perebor (brute-force search) algorithms. *Annals of the History of Computing*, 6(4):384–400, 1984.

64 Christopher Umans, Tiziano Villa, and Alberto L. Sangiovanni-Vincentelli. Complexity of two-level logic minimization. *IEEE Transactions on CAD of Integrated Circuits and Systems*, 25(7):1230–1246, 2006.

65 David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128, 2007.

Polynomial Time Algorithms in Invariant Theory for Torus Actions

Peter Bürgisser ✉
Institut für Mathematik, Technische Universität Berlin, Germany

M. Levent Doğan ✉
Institut für Mathematik, Technische Universität Berlin, Germany

Visu Makam ✉
Institute for Advanced Study, Princeton, NJ, CA
School of Mathematics and Statistics, University of Melbourne, Australia

Michael Walter ✉
Korteweg-de Vries Institute for Mathematics,
Institute for Theoretical Physics, Institute for Logic, Language and Computation,
University of Amsterdam, The Netherlands

Avi Wigderson ✉
Institute for Advanced Study, Princeton, NJ, USA

Abstract

An action of a group on a vector space partitions the latter into a set of orbits. We consider three natural and useful algorithmic "isomorphism" or "classification" problems, namely, *orbit equality*, *orbit closure intersection*, and *orbit closure containment*. These capture and relate to a variety of problems within mathematics, physics and computer science, optimization and statistics. These orbit problems extend the more basic null cone problem, whose algorithmic complexity has seen significant progress in recent years.

In this paper, we initiate a study of these problems by focusing on the actions of commutative groups (namely, tori). We explain how this setting is motivated from questions in algebraic complexity, and is still rich enough to capture interesting combinatorial algorithmic problems. While the structural theory of commutative actions is well understood, no general efficient algorithms were known for the aforementioned problems. Our main results are polynomial time algorithms for all three problems. We also show how to efficiently find separating invariants for orbits, and how to compute systems of generating rational invariants for these actions (in contrast, for polynomial invariants the latter is known to be hard). Our techniques are based on a combination of fundamental results in invariant theory, linear programming, and algorithmic lattice theory.

2012 ACM Subject Classification Computing methodologies → Algebraic algorithms; Theory of computation → Algebraic complexity theory

Keywords and phrases computational invariant theory, geometric complexity theory, orbit closure intersection problem

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.32

Related Version *Full Version*: https://arxiv.org/abs/2102.07727

Funding *Peter Bürgisser*: ERC under the European Union's Horizon 2020 Research and Innovation Programme (grant agreement no. 787840).
M. Levent Doğan: ERC under the European Union's Horizon 2020 Research and Innovation Programme (grant agreement no. 787840).
Visu Makam: University of Melbourne and NSF grant CCF-1900460.
Michael Walter: NWO Veni grant no. 680-47-459 and NWO grant OCENW.KLEIN.267.
Avi Wigderson: NSF Grant CCF-1900460.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 32; pp. 32:1–32:30
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Consider the following two problems, which on the face of it have nothing to do with each other:

1. Will the cue ball's trajectory on a billiards table ever end up in a pocket?
2. Given a bipartite graph G, and two functions w, w' assigning weights to edges, is it the case that they assign the same weight to *every* perfect matching M of G?

Both turn out to be orbit problems for torus actions, and exemplify the class of problems we study in this paper.

As our introduction is somewhat long, we break it up as follows. We start with general background to algorithmic invariant theory in §1.1 and discuss general orbit problems in §1.2. In §1.3 we define torus actions, discuss our main results, and explain their motivation from the perspective of algebraic complexity. In §1.4, we give examples of how these orbit problems for torus actions arise in and capture natural problems in physics and optimization. In §1.5, we discuss the organization of the paper and logical structure of our results.

1.1 Algorithms in invariant theory

Computational invariant theory is a subject whose origins can be traced back to "masters of computation" in the 19th century such as Boole, Gordan, Sylvester and Cayley among others. The second half of the 20th century injected a major impetus to both structural and computational aspects of these mathematical areas. On the one hand, the advent of digital computers allowed mathematicians means to study much larger such algebraic structures than could be accessed by hand. On the other, the parallel development of computational complexity provided a mathematical theory with precise computational models for algorithms and their efficiency analysis. This combination has injected many new ideas and questions into invariant theory and related fields, leading to the development of algorithmic techniques such as Gröbner bases and many others, which supported faster and faster algorithms. Texts on this large body of work can be found, for example, in the books [17, 60, 14]. While the computational complexity put focus on polynomial time as the staple of efficiency, it also provided means to argue the likely impossibility of such fast algorithms for certain tasks, through the Cook-Karp-Levin theory [13, 44, 48] of NP-completeness (for Boolean computation) and Valiant's theory of VNP-completeness.

More recently, a further surge in collaboration between algebraists and complexity theorists on these algorithmic questions in invariant theory and representation theory arose from two (related) sources starting in the turn of this century. Both imply that these very algorithmic questions in algebra are deeply entwined with the core complexity questions of P vs. NP and VP vs. VNP. Not surprisingly, new enriching connections between these two research directions are newly found as they develop, providing an exciting collaboration.

The first source is Mulmuley and Sohoni's Geometric Complexity Theory (GCT) [53], which highlights the inherent symmetries of complete problems of these complexity classes, and through these suggests concrete invariant theoretic and representation theoretic attacks on the questions above. This has lead to many new questions, techniques, and much faster algorithms (see, for example, [52, 26, 7, 51]).

The second source is the work of Impagliazzo and Kabanets [42], using Valiant's completeness theory for VP and VNP to again attack these major complexity problems directly through the development of efficient deterministic algorithms for the basic PIT (Polynomial Identity Testing) problem. This problem, which (again, thanks to Valiant's completeness) has natural symmetries, is very similar to basic invariant theory problems. Major progress was recently made on resolving such related algorithmic problems, starting with [33, 27, 40, 41, 19]. Many others continue to follow, see, for example, [21, 1, 28, 10, 9, 11, 8]. We refer to [8] for a recent description of the state-of-art.

1.2 Orbit problems

We now briefly describe the basic setting and problems of interest, postponing some of the technical details to later sections for the sake of brevity. A group homomorphism $\rho\colon G \to \mathrm{GL}(V)$, where V is a vector space (always complex and finite-dimensional) is called a representation of G. One can think of this as a (linear) action of G on V, i.e., a map $G \times V \to V$ where $(g, v) \mapsto \rho(g)v$ satisfies the usual axioms of a group action. For us, groups will always be algebraic and representations rational, that is, morphisms of algebraic groups. We will denote $\rho(g)v$ by gv or $g \cdot v$.

For $v \in V$, we define its *orbit* $O_v := \{gv \mid g \in G\}$ (denoted $O_{G,v}$ if the group is not clear from context) to be the subset of points that can be reached from v by applying a group element. We denote by $\overline{O_v}$ the topological closure of O_v. These notions are extremely basic and in many concrete instances very familiar. One simple example is the action of $\mathrm{GL}_n \times \mathrm{GL}_n$ on $n \times n$ matrices by left and right multiplication: clearly, the orbit of a matrix A consists of the matrices having the same rank as A; moreover, the orbit closure of A is the set of matrices whose rank is at most the rank of A. Another example is the conjugation action of GL_n on $n \times n$ matrices, where the orbits are characterized by Jordan normal forms.[1]

Understanding the space of orbits of a given group action is perhaps the most basic task of invariant theory. The following three basic algorithmic problems will be the focus of this paper.

▶ **Problem 1.1.** *Let $\rho\colon G \to \mathrm{GL}(V)$ be a representation of a group G. Given $v, w \in V$:*

1. ***Orbit equality:*** *Decide if $O_v = O_w$;*
2. ***Orbit closure intersection:*** *Decide if $\overline{O_v} \cap \overline{O_w} \neq \emptyset$;*
3. ***Orbit closure containment:*** *Decide if $w \in \overline{O_v}$.*[2]

As we will discuss the computational complexity of algorithms for these problems, one needs to specify how inputs are given and how we measure their size. We will discuss this, but for now it suffices to think of $n = \dim(V)$, the degree of ρ (assuming it is a polynomial function), and the bit-length of the input vectors v, w as the key size parameters.

The aforementioned problems capture and are related to a natural class of "isomorphism" or "classification" problems across many domains in mathematics, physics and computer science. Examples include the graph isomorphism problem [16], non-commutative rational identity testing [27, 41], equivalence problems on quiver representations [18, 20], matrix and tensor scaling [10, 9], classification of quantum states [4] and module isomorphism problems [6].

To briefly hint at the role of invariant theory, let us take a closer look at problem (2), that is, the problem of orbit closure intersection. We denote by $\mathbb{C}[V]$ the ring of polynomial functions on V. A polynomial function f on V is called *invariant* if it is constant along

[1] The orbit closures of two matrices intersect if and only if the matrices have the same eigenvalues (counted with multiplicity).

[2] The special case of $w = 0$ is called the null cone membership problem. In fact, many of the recent algorithmic advances mentioned above efficiently solve the null cone problem for specific group actions, see [8] and references therein. The motivation of this paper is to extend that understanding to these more general problems.

orbits, i.e., $f(gv) = f(v)$ for all $g \in G$ and $v \in V$. The collection of all invariant polynomials forms a subring $\mathbb{C}[V]^G$, called the *invariant ring.* Since polynomials are continuous, invariant polynomials are constant along orbit closures. In particular, two points v and w are indistinguishable by invariant polynomials when their orbit closures intersect. Amazingly, the converse is also true for a large class of group actions thanks to a result due to Mumford: if the orbit closures of v and w do not intersect, then they can always be distinguished by an invariant polynomial. See Theorem 2.1 for a precise statement.

Mumford's theorem suggests an approach to orbit closure intersection – test if $f(v) = f(w)$ for all invariant polynomials f. For this strategy to be effective, one needs a computational handle on invariant polynomials. Naively there are infinitely many polynomials, but a foundational result of Hilbert helps tackle this issue. A *system of generating polynomial invariants* is a collection of invariant polynomials $\{f_1, \dots, f_r\}$ such that any other invariant polynomial can be written as a polynomial in the f_i's. In particular, to test for orbit closure intersection it suffices to test whether each of the f_i take the same value on both points. Hilbert showed the existence of a finite system of generating polynomial invariants and also gave an algorithm to produce them [36]. Since then, many improvements on the complexity of such algorithms were developed, but even today this task is, in general, infeasible. One basic obstacle is the very description of such a system of generating invariants, coming both from the size of this set and the degree of each polynomial in it.

Nearly a century later, a (singly) exponential bound (in n) on the degrees of a system of generating polynomial invariants was achieved for a very general class of group actions [15], which is unfortunately the best possible in this generality, see [22]. A singly exponential bound is necessary to capture a polynomial with a poly-sized (in n) arithmetic circuit, but is by no means sufficient.[3] Another issue that one has to deal with is the number of invariants in a system of generating polynomial invariants, and it is often the case that there are exponentially many in any system.[4] This led Mulmuley [52] to suggest the notion of a *succinct circuit* as a way to capture a system of generating polynomial invariants with a view towards using them for orbit closure intersection. Unfortunately, this approach does not seem to be computationally feasible either. See [29] where Mulmuley's conjecture [52, Conjecture 5.3] on the existence of succinct circuits was disproved under natural complexity assumptions. What is perhaps most surprising is that this already happens for a *commutative* group action, namely when G is a torus. Further, an example of a group action was given where any system of generating polynomial invariants must contain a VNP-hard polynomial.

The negative result above seem to suggest that the algorithmic tasks at hand are infeasible, even for torus actions, i.e., groups of the form $(\mathbb{C}^\times)^d$. The main results of our paper show the opposite: *all of them are efficiently solvable for torus actions!*

The main novelty on our approach is using *rational invariants* instead of polynomial invariants. A rational invariant is a quotient of polynomials that is invariant, see Section 1.3 for a precise definition. This is a bit unexpected since Mumford's theorem simply does not extend to rational invariants: it is easy to construct examples where two points whose orbit closures intersect are distinguished by a rational invariant. Yet, for representations of tori, we show that (a certain special collection of) rational invariants can be used (in a delicate way) to capture not just orbit closure intersection, but orbit closure containment and orbit equality as well. Moreover, we show that rational invariants are computationally easy in this case, in stark contrast with the aforementioned hardness results for polynomial invariants [29].

[3] For example, the permanent of an $n \times n$ matrix, which has degree n, is believed to require exponential circuit size. This is essentially the content of Valiant's proof that the permanent is complete for the class VNP, combined with the hypothesis that VNP $\neq$ VP.

[4] This is already the case for the matrix scaling action discussed in Section 1.4.

Inspired by the connections to the P vs. NP problem, the GCT program makes several predictions in invariant theory. The setting in which most of the predictions and conjectures are formulated is the setting of rational representations of connected reductive groups (which we will define later). Here, we want to point out that among connected reductive groups, the class of commutative groups happen to be precisely tori. Thus, our main results should be viewed as conclusively verifying several predictions of GCT in the commutative case. Moreover, the barrier result on the computational efficiency of polynomial invariants [29] along with our results on rational invariants suggest that a more thorough investigation of rational invariants is needed in the case where the acting group is non-commutative, e.g., SL_n.

1.3 Torus actions and main results

We now discuss the main contributions of our paper in more detail and precision. Our results concern torus actions, so we specialize the discussion of the preceding section and consider a d-dimensional complex torus $T = (\mathbb{C}^\times)^d$ as the acting group G. The group law is just pointwise multiplication, i.e., $(t_1, \ldots, t_d) \cdot (s_1, \ldots, s_d) = (t_1 s_1, \ldots, t_d s_d)$.

Any linear action of a torus can be described by an integer matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ called the *weight matrix* (where $\mathrm{Mat}_{d,n}(\mathbb{Z})$ denotes the space of $d \times n$ integer matrices). The representation $\rho_M \colon T \to \mathrm{GL}_n(\mathbb{C})$ corresponding to a weight matrix $M = (m_{ij})$ looks as follows:

$$\rho_M(t) = \begin{pmatrix} \prod_{i=1}^d t_i^{m_{i1}} & & \\ & \ddots & \\ & & \prod_{i=1}^d t_i^{m_{in}} \end{pmatrix} \tag{1}$$

Thus any torus action can be viewed as a scaling action, where each coordinate is scaled separately according to a Laurent monomial.[5] The weight matrix (up to reordering of columns) determines the representation. Despite the simple description of commutative torus actions, they as well capture fundamental notions, and the associated orbits can be quite complex. One example is the matrix scaling problem, where the orbits capture weights of perfect matchings (see Problem 1.7).

In this paper, we will assume that a torus action is given by specifying the weight matrix. Thus the bit-length of the entries of the weight matrix are included in the input size of the problems. Moreover, we will allow complex number inputs. These can be described up to finite precision by elements in the field of Gaussian rationals $\mathbb{Q}(i) = \{s + it \mid s, t \in \mathbb{Q}\}$, which will be encoded in the standard way; see, e.g., [52].[6] The following theorem captures the main results of our paper.

▶ **Theorem 1.2.** *Given as input a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ as well as vectors $v, w \in \mathbb{Q}(i)^n$, denote by b the maximal bit-length of the entries of v, w, and M. Then we can in time* $\mathrm{poly}(d, n, b)$*:*

1. *decide whether $O_v = O_w$;*
2. *decide whether $\overline{O_v} \cap \overline{O_w} \neq \emptyset$;*
3. *decide whether $w \in \overline{O_v}$.*

In other words, for rational representations of tori, there are polynomial time algorithms for orbit equality, orbit closure intersection, and orbit closure containment.

[5] We can also describe this action as follows: Identify $v \in \mathbb{C}^n$ with a Laurent polynomial $\sum_{j=1}^n v_j \, z_1^{m_{1j}} \cdots z_d^{m_{dj}}$; then the action of T corresponds precisely to rescaling the variables $z_1, \ldots, z_d$ [34].

[6] In fact, our results hold more generally when the elements in $\mathbb{Q}(i)$ are given in a "floating point" format, namely in the form $(s + it)2^p$, with $s, t \in \mathbb{Q}$ and $p \in \mathbb{Z}$ encoded in binary in the standard way. The same is true for input of the form 2^p, with $p \in \mathbb{Q}$ encoded in binary. See Remark 5.5.

We note that the null cone membership problem mentioned earlier, namely Problems 1.1 (2)/(3) when the input vector w is the 0 vector, was known to have a polynomial time algorithm by a simple reduction to linear programming.[7] There is no known way of doing the same for the orbit problems above, and indeed our theorem above takes an alternative route.

While one might hope for efficient algorithms for Problems 1.1 (1) and (2) in much more general situations than for tori (for general reductive group actions), our efficient algorithm for orbit closure containment is in stark contrast to the known NP-hardness of the general orbit closure containment problem [5]. Our work points to a key difference: namely, for torus group actions, one can use one-parameter subgroups combined with linear programming techniques to reduce orbit closure containment to orbit equality, while this is impossible in this form for general actions. See Section 7 for more details.

A common core underlying all our results is an efficient algorithm for computing invariant Laurent polynomials for torus actions. The key idea is the following. Invariant polynomials for torus actions can be quite complicated. However, suppose that we restrict to vectors of some fixed support, i.e., "nonzero pattern" of the coordinates. This restriction is without loss of generality, since two vectors can only be in the same orbit when their supports coincide. However, it allows us to study a richer class of functions, namely *Laurent polynomials* instead of ordinary polynomials. Allowing for negative exponents makes an important difference: while polynomial invariants naturally form a semigroup, invariant Laurent polynomials form a *lattice*, isomorphic to the integral vectors in the kernel of the weight matrix. Lattices are much better behaved than semigroups, for example they have small bases which can be found efficiently.

Before describing our results, let us define invariant Laurent polynomials more precisely. For a representation $\rho\colon G \to \mathrm{GL}(V)$ of a group G, we have an action of G on the polynomial ring $\mathbb{C}[V]$ defined by $(g \cdot f)(v) := f(\rho(g)^{-1}v)$. When $V = \mathbb{C}^n$, we can identify $\mathbb{C}[V] = \mathbb{C}[x_1, \ldots, x_n]$ with the polynomial ring in n variables. Now consider the set of vectors with nonzero coordinates in $S \subseteq [n]$:

$$X_S = \{v \in \mathbb{C}^n \mid v_j \neq 0 \text{ if and only if } j \in S\}.$$

The Laurent polynomials in the variables x_j for $j \in S$ form the natural class of functions on X_S (since we can always divide by the nonzero coordinates). Accordingly, we will denote their collection by $\mathbb{C}[X_S]$.[8] Now, for a torus action of the form (1), the group T acts on any monomial $x^c = x_1^{c_1} \cdots x_n^{c_n}$ by a simple rescaling. Accordingly, we also have an action of T on the algebra of Laurent polynomials $\mathbb{C}[X_S]$. A Laurent polynomial f is called *invariant* if $g \cdot f = f$ for all $g \in G$. Clearly, if f is invariant, then so are all the Laurent monomials occurring in f. The collection of all invariant Laurent polynomials forms the subalgebra $\mathbb{C}[X_S]^G$ of *invariant Laurent polynomials.* A collection of invariant Laurent polynomials $f_1, \ldots, f_r$ is called a *system of generating invariant Laurent polynomials in the variables* $\{x_j\}_{j \in S}$ if they generate $\mathbb{C}[X_S]^G$ as an algebra. For torus actions, these can always be taken to be Laurent *monomials*, in which case we call them a *system of generating invariant Laurent monomials.* We can then state our key result:

[7] Namely, a vector v is in the null cone if and only if the convex hull of the weights corresponding to the nonzero coordinates of v does not contain the origin.

[8] In the language of algebraic geometry, these are the "regular" functions on X_S.

▶ **Theorem 1.3.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ define an n-dimensional representation of $T = (\mathbb{C}^\times)^d$, and let $S \subseteq [n]$. Assume that the bit-lengths of the entries of M are bounded by b. Then, in $\mathrm{poly}(d, n, b)$-time, we can construct an arithmetic circuit with division $\mathcal{C}$ whose output gates compute a system of generating invariant Laurent monomials $f_1, \ldots, f_r$ in the variables $\{x_j\}_{j \in S}$, where $r \leq n$.*

Here we recall the notion of an *arithmetic circuit with division*, which is a directed acyclic graph as follows. Every node of indegree zero is called an input gate and is labeled by either a variable or a rational (complex) number. Nodes of indegree one and outdegree one are labeled by $^{-1}$ and are called division gates. Nodes of indegree two and outdegree one and is labeled either $+$ or $\times$; in the first case it is a sum gate and in the second a product gate. The only other nodes allowed are output gates which have indegree one and outdegree zero. Given an arithmetic circuit with division, it computes a rational function at each output node in the obvious way. The bit size of such an arithmetic circuit is the total number of nodes plus the total bit-length of the specification of all rational numbers computed in *all* of its gates. The notion of *(division free) arithmetic circuits* is obtained by disallowing division gates. They compute polynomials in the obvious way.

We emphasize that the number of generators produced by Theorem 1.3 is at most n (in particular, independent of the bit-length b), in stark contrast to the situation for monomial invariants. Moreover, the bit-length of $\mathcal{C}$ is polynomially bounded.

As a consequence of Theorem 1.3, we are also able to construct arithmetic circuits that compute a generating set of *rational invariants*. For a representation $\rho \colon G \to \mathrm{GL}(V)$, the action of G on the polynomial ring $\mathbb{C}[V]$ always extends to an action on its field of rational functions, the rational functions $\mathbb{C}(V)$. A rational function $f \in \mathbb{C}(V)$ is called *invariant* if $g \cdot f = f$ for all $g \in G$. The collection of all rational invariants forms the sub-field $\mathbb{C}(V)^G$ of *rational invariants*. A collection of rational invariants $f_1, \ldots, f_r \in \mathbb{C}(V)$ is called a *system of generating rational invariants* if they generate $\mathbb{C}(V)^G$ as a field extension of $\mathbb{C}$. Note that any invariant Laurent polynomial is a rational invariant, but the converse is not necessarily true. Nevertheless:

▶ **Corollary 1.4.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ define an n-dimensional representation of $T = (\mathbb{C}^\times)^d$. Assume that the bit-lengths of the entries of M are bounded by b. Then, in $\mathrm{poly}(d, n, b)$-time, we can construct an arithmetic circuit with division $\mathcal{C}$ whose output gates compute a system of generating rational invariants $f_1, \ldots, f_r \in \mathbb{C}(x_1, \ldots, x_n)^T$, where $r \leq n$.*

This result is in distinct contrast to the impossibility of finding succinct circuits for generating *polynomial* invariants under natural complexity assumptions [29].

Furthermore, we can complement Theorem 1.2 in the following way: if two orbit closures do not intersect, $\overline{O_v} \cap \overline{O_w} \neq \emptyset$, then we can construct in polynomial time an arithmetic circuit computing a separating invariant monomial that can serve as a "witness" of the non-intersection.

▶ **Corollary 1.5.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ define an n-dimensional representation of $T = (\mathbb{C}^\times)^d$. Let $v, w \in \mathbb{Q}(i)$ be such that $\overline{O_v} \cap \overline{O_w} = \emptyset$. Assume the bit-lengths of the entries of v, w and M are bounded by b. Then, in $\mathrm{poly}(d, n, b)$-time, we can construct an arithmetic circuit of bit-length $\mathrm{poly}(d, n, b)$, which computes an invariant monomial f such that $f(v) \neq f(w)$.*

So far, we have discussed orbit problems for complex tori $T = (\mathbb{C}^\times)^d$. It is interesting to ask to which extent our results hold for "compact" tori, which are groups of the form $K = (\mathrm{S}^1)^d$, where $\mathrm{S}^1 = \{z \in \mathbb{C}^\times \mid |z| = 1\}$.[9] Besides the fundamental algorithmic interest in this

[9] Note that K is indeed compact, and a subgroup of T. Moreover, any commutative compact connected

setting, such group actions are important in several areas. For example, the time evolution of periodic systems in Hamiltonian mechanics are naturally given by S^1-actions, and important symmetries in classical and quantum physics are given by compact group actions.

In fact, the results discussed so far can also be used to give an efficient solution for orbit problems for compact tori. Any (continuous) finite-dimensional representation of $(\mathrm{S}^1)^d$ extends to a representation of $(\mathbb{C}^\times)^d$, so representations are specified as before by a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$. Moreover, the compactness implies that orbits $O_{K,v} = \{kv \mid k \in K\}$ are closed and so all three problems mentioned in Problem 1.1 coincide. Therefore, the following corollary solves all three problems for compact tori:

▶ **Corollary 1.6.** *Let the weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ define an n-dimensional representation of $T = (\mathbb{C}^\times)^d$ and put $K = (\mathrm{S}^1)^d$. Further, let $v, w \in \mathbb{Q}(i)^n$ and assume that the bit-lengths of the entries of v, w and M are bounded by b. Then, in* $\mathrm{poly}(d,n,b)$*-time, we can decide if $O_{K,v} = O_{K,w}$.*

To give additional context to this result, we briefly mention some recent results achieving polynomial time algorithms for orbit closure intersection of *specific* group actions. For the left-right action (of $\mathrm{SL}_n \times \mathrm{SL}_n$ on m-tuples of $n \times n$ matrices), one approach to solving the orbit closure intersection problem is to (approximately) reduce to the orbit equality problem for the maximal compact subgroup (which happens to be $\mathrm{SU}(n) \times \mathrm{SU}(n)$, where $\mathrm{SU}(n)$ denotes the group of $n \times n$ unitary matrices with determinant 1), see[1]. This was achieved by using a geodesic convex optimization algorithm. Given the recent advances in this area (see, e.g., [8] and references therein), it is natural to ask if a similar approach could be useful for general reductive group actions. For torus actions, interestingly, we can also go in the other direction. Namely, our result for the orbit equality problem for the maximal compact subgroup, Corollary 1.6, is derived from our main result for complex tori, i.e., Theorem 1.2. More generally, we observe that for arbitrary reductive group actions, the orbit equality problem for the maximal compact subgroup is always equivalent to an orbit closure intersection (or equality) problem for a related action of the larger group, see Theorem 8.2 for a precise statement.

The results in this paper warrant the investigation of several interesting directions that we leave for future work, some of which we will discuss in Section 9.

1.4 Further motivation and algorithmic applications

As we saw above, orbit problems are related to a great number of applications. Despite significant progress, for general reductive group actions it is still an open problem to design fast algorithms for these problems. Our results fully resolve the situation in the case of torus actions and also show how to overcome barriers that had previously been pointed out in the literature [39, 29]. Apart from its fundamental complexity theoretic interest, there are also several algorithmic applications where torus actions arise naturally. One particular application in [37] shows how one can use torus invariants to simplify a system of differential equations with scaling symmetries. We provide and discuss in more detail some other concrete applications to combinatorial optimization and to dynamical systems, which were already mentioned briefly at the beginning of the introduction.

We first explain a link to combinatorial optimization. Consider edge weights w for the complete bipartite graph on $2n$ labeled vertices (n on each side): the weight $w(e)$ of an edge e is assumed to be a rational number, encoded in binary. We define the weight $w(M)$ of a perfect matching M of G as the sum of the weights of the edges occurring in M.

Lie group is of this form.

▶ **Problem 1.7.** *Given edge weights w and w' as above, decide whether they assign the same weight to* every *perfect matching M of G.*

Perhaps surprisingly, this problem can be reformulated as an orbit intersection problem for a torus action (see below). Therefore, Theorem 1.2 implies that Problem 1.7 can be solved in polynomial time. This insight seems far from being obvious!

The relevant torus action here results from from matrix scaling, which has been widely studied and has many applications; see [58] and [12] for more recent developments. Consider $\mathrm{ST}_n := \{(t_1, \ldots, t_n) \in \mathbb{C}^\times \mid t_1 \cdots t_n = 1\}$, which is isomorphic to the algebraic torus $(\mathbb{C}^\times)^{n-1}$. We let $\mathrm{ST}_n \times \mathrm{ST}_n$ act on $\mathrm{Mat}_n(\mathbb{C})$ by left-right multiplication as follows:

$$((t_1, \ldots, t_n), (s_1, \ldots, s_n)) \cdot (v_{ij}) := (t_i v_{ij} s_j)_{ij}. \tag{2}$$

Moreover, we shall identify the edge weights w_{ij}, where $i, j \in [n]$, with the matrix $v_w = (2^{w_{ij}}) \in \mathrm{Mat}_n(\mathbb{C})$.[10] Then one can show that the answer to Problem 1.7 is affirmative if and only if the orbit closures of v_w and $v_{w'}$ in $\mathrm{Mat}_n(\mathbb{C})$ intersect. This follows from Mumford's theorem mentioned earlier, along with the fact that the invariant polynomials for this action are generated by the perfect matchings, namely the monomials $f_\pi = x_{1,\pi(1)} \cdots x_{n,\pi(n)}$ where $\pi \in S_n$ ranges over the permutations [47, Theorem 3]. Indeed, multiplying entries of v_w is the same as summing the corresponding edge weights in the exponent, hence $f_\pi(v_w) = 2^{w(M)}$, where M is the perfect matching defined by the permutation π.

We briefly comment on the 3-dimensional generalization of this action. $\mathrm{ST}_n \times \mathrm{ST}_n \times \mathrm{ST}_n$ acts on 3-tensors in $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \mathbb{C}^n$ in the natural way:

$$((t_1, \ldots, t_n), (s_1, \ldots, s_n), (u_1, \ldots, u_n)) \cdot (v_{ijk}) = (t_i s_j u_k v_{ijk})_{ijk}.$$

In this case, any system of generating polynomial invariants must include the (maximum) 3-dimensional matching monomials $f_{\pi,\tau} = x_{1,\pi(1),\tau(1)} \cdots x_{1,\pi(n),\tau(n)}$ for $\pi, \tau \in S_n$, which led to the barrier result for torus actions in [29]. Of course, in this case there are additional generating invariants, see, e.g., [49]. Our results show that the corresponding orbit problems can nevertheless be solved in polynomial time! Moreover, it is possible to efficiently exhibit *separating* polynomial invariants (whenever they exists) as well as to construct systems of generating invariant *Laurent polynomial* or *rational* invariants.

Our second example concerns a connection to dynamical systems. Consider a (massless) cue ball on a billiard table (assumed to be square to simplify the discussion). We can ask:

▶ **Problem 1.8.** *If we hit the cue ball at a given angle, will its trajectory end up in a pocket?*

It is well-known, and easy to see, that one can map trajectories on an ordinary billiard with reflecting boundaries to a billiard of twice the size with periodic boundaries, say $(\mathbb{R}/2\pi\mathbb{Z})^2$. The trajectory of the ball depends fundamentally on the angle or slope. If the slope is irrational, then the trajectory will be dense, so the answer to Problem 1.8 is trivially yes. Otherwise, the trajectory will be periodic and the problem is nontrivial. We can model it as an orbit problem as follows. Let the compact torus S^1 act on $\mathbb{C}^2$ by

$$t \cdot (x, y) := (t^p x, t^q y),$$

where $s = \frac{q}{p}$ is the slope by which we hit the ball. We can identify points (θ, ν) on the periodic billiard with points $(e^{i\theta}, e^{i\nu}) \in \mathbb{C}^2$. In this way, Problem 1.8 reduces to a constant number of orbit equality problems for this action (one for each pocket). While the problem is

[10] As explained in footnote 6, our results also hold for input of this form, where the w_{ij} are specified in binary.

certainly easy to solve by a variety of methods, one can ask analogous questions for billiards in $n > 2$ dimensions and by allowing a d-dimensional hyperplane worth of allowed cue directions. Such generalizations similarly correspond to orbit problems for compact tori $(\mathrm{S}^1)^d$ on some $\mathbb{C}^n$, and they can all be solved in polynomial time by using Corollary 1.6.

1.5 Organization of the paper

In Section 2, we give an introduction to basic results in invariant theory that we will need to establish our results. In Section 3, we focus on tori, their representations, and their invariants. In particular, we will show that the faces of a natural convex polyhedral "Newton cone" are in one-to-one correspondence with the orbits in an orbit closure, which will be an important ingredient later on.

In Section 4, we discuss the definition and computation of suitable *rational* invariants. As mentioned above, our key result is that for fixed support, a small generating set of invariant Laurent monomials can be computed efficiently. This result, which is Theorem 1.3, is at the heart of our algorithms, and also of independent interest. We achieve this using Smith normal forms. As an easy consequence, this also implies that we can efficiently compute a small generating set of rational invariants for a given representation, that is, Corollary 1.4.

In Section 5, we explain how to use the results of the preceding section to solve the orbit equality problem in polynomial time. This establishes part (1) of Theorem 1.2. Here we rely on known results for testing if a given Laurent monomial (of possibly exponential degree) evaluates to the same value on two given vectors, and we present a brief sketch for completeness.

In Sections 6 and 7, we show how to solve the orbit closure intersection and containment problems by reducing them to orbit equality. This establishes parts (2) and (3) of Theorem 1.2. Here we use the polyhedral description of the structure of orbit closures as furnished by the Newton cone. Furthermore, we show that given two points whose orbit closures do not intersect, we can efficiently construct a separating monomial invariant as a "witness". This proves Corollary 1.5.

In Section 8, we show how to solve the orbit equality problem for compact tori. This establishes Corollary 1.6. We also give, for general reductive groups G, a reduction from orbit equality for a maximally compact subgroup $K \subseteq G$ to orbit equality and orbit closure intersection for G.

In Section 9, we summarize our results and discuss some important open problems and future directions.

Conventions

In this paper, sometimes we work with monomials and sometimes with Laurent monomials. Unless we use the prefix "Laurent", by a monomial, we mean $\prod_j x_j^{c_j}$ where $c_j \in \mathbb{Z}_{\geq 0}$, i.e., all exponents are non-negative. Whenever exponents are allowed to be negative, we will be careful to specify that it is a Laurent monomial.

2 Preliminaries of invariant theory

We will briefly recall the main results in invariant theory that are relevant for us (see [46, 23, 17, 55] for details). We will take our ground field to be $\mathbb{C}$, the field of complex numbers, for simplicity. However, much of this theory works for any algebraically closed field. For a (finite-dimensional) vector space V, we denote by $\mathbb{C}[V]$ the ring of polynomial functions on V. For our purposes, if V is the standard vector space $\mathbb{C}^n$, then $\mathbb{C}[V] = \mathbb{C}[x_1, \dots, x_n]$, the polynomial ring in n variables, where x_i is to be interpreted as the i^{th} coordinate function.

Let G be an algebraic group, i.e., it has the structure of an algebraic variety (not necessarily irreducible) such that the multiplication map $m\colon G \times G \to G$ and the inverse map $\iota\colon G \to G$ are morphisms of varieties.[11] A morphism of algebraic groups $\rho : G \to \mathrm{GL}(V)$ is called a rational representation of G.[12] We write gv or $g \cdot v$ for $\rho(g)v$. For a point $v \in V$, its orbit O_v (or $O_{G,v}$ when the group is not clear from context) is the set of all points that can be reached from v by the action of an element of the group, i.e.,

$$O_v := \{gv \mid g \in G\}.$$

We denote by $\overline{O_v}$ the closure of the orbit O_v. The closure is to be taken either with respect to the Euclidean topology or the Zariski topology. Indeed, the closures in both topologies coincide, a well-known fact that relies on a fundamental result in algebraic geometry due to Chevalley (see [54, I.§10]). A polynomial function $f \in \mathbb{C}[V]$ is called *invariant* if it is oblivious to the group action, i.e., $f(gv) = f(v)$ for all $g \in G$, $v \in V$. The collection of all invariant polynomials forms a subring

$$\mathbb{C}[V]^G := \{f \in \mathbb{C}[V] \mid \forall\, g \in G, v \in V \;\, f(gv) = f(v)\}.$$

One key observation is that invariant functions are constant along orbits and hence constant along orbit closures as well. Hence, if the orbit closures of two points intersect, then they cannot be distinguished by an invariant function. The converse was proved by Mumford for a special class of groups called reductive groups [55] (see also [17, Corollary 2.3.8]). An algebraic group G is called *reductive* if every rational representation is a direct sum of irreducible representations, wherein a representation is called irreducible if it has no non-trivial subrepresentations. Examples of reductive groups include $\mathrm{SL}_n, \mathrm{GL}_n, \mathrm{Sp}_n, \mathrm{O}_n$, finite groups, and most importantly for us, tori (which we define formally in the next section), as well as direct products thereof.[13] Reductive groups have played a central role for a number of mathematical fields for over a century. A particularly important result in the invariant theory of reductive groups is that invariant rings are finitely generated [36, 35, 62].

To state Mumford's result in the generality we need, we will define rational actions on varieties (a notion that naturally generalizes rational representations). Let X be an algebraic variety and let $\mathbb{C}[X]$ denote the ring of regular functions on X. A rational action of an algebraic group G on X is a morphism of varieties $G \times X \to X, (g, x) \mapsto g \cdot x$ satisfying $g \cdot (g' \cdot x) = (gg') \cdot x$ and $e \cdot x = x$ for all $x \in X$, $g, g' \in G$. As in the vector space case, we denote the orbit of a vector $v \in X$ by O_v.

▶ **Theorem 2.1** (Mumford, [55]). *Let G be a reductive group. Let X be an algebraic variety and suppose we have a rational action of G on X. For $v, w \in X$ we have $\overline{O_v} \cap \overline{O_w} = \emptyset$ if and only if there exists $f \in \mathbb{C}[X]^G$ such that $f(v) \neq f(w)$.*

Another well-known important structural result states that every orbit closure $\overline{O_v}$ contains a *unique* closed orbit.

[11] A morphism of varieties simply means that in local coordinates the map is given by ratios of polynomials. For concreteness, the reader may simply think of an algebraic group as a matrix group, i.e., a subgroup of $\mathrm{GL}_n(\mathbb{C})$ that is described as the zero locus of a collection of polynomials.

[12] One can interpret this action as the action of the subgroup $\rho(G) \subseteq \mathrm{GL}(V)$ on V by matrix-vector multiplication, where $\rho(G)$ is parametrized algebraically by an algebraic group G.

[13] The group B_n of upper triangular $n \times n$ invertible matrices is a typical example of a group that is not reductive.

▶ **Theorem 2.2.** *Let $\rho\colon G \to \mathrm{GL}(V)$ be a rational representation of a reductive group G. Then:*

1. *For any $v \in V$, the orbit closure $\overline{O_v}$ contains a unique closed orbit, that we denote by $O_{\widetilde{v}}$.*
2. *If $v, w \in V$, then*

$$\overline{O_v} \cap \overline{O_w} \neq \emptyset \iff O_{\widetilde{v}} = O_{\widetilde{w}}.$$

Proof. (1) The first assertion is [17, Theorem 2.3.6].

(2) For the second assertion, if the orbit closures $\overline{O_v}$ and $\overline{O_w}$ are disjoint, then so are the orbits $O_{\widetilde{v}}$ and $O_{\widetilde{w}}$, which therefore must be different. Conversely, suppose $O_{\widetilde{v}} \neq O_{\widetilde{w}}$. Since these orbits are closed, by Theorem 2.1, there is an invariant $f \in \mathbb{C}[V]^G$ such that $f(\widetilde{v}) \neq f(\widetilde{w})$. By continuity, $f(v) = f(\widetilde{v}) \neq f(\widetilde{w}) = f(w)$, which implies $\overline{O_v} \cap \overline{O_w} = \emptyset$ by another application of Theorem 2.1. ◀

Part(2) of this theorem shows that the orbit closure intersection problem can be reduced to the orbit equality problem, provided we can compute the unique closed orbit $O_{\widetilde{v}}$ contained in $\overline{O_v}$. We will see in Section 6 that if the group G is a torus, this can be achieved in polynomial time.

Another key result in understanding orbit closures is the Hilbert–Mumford criterion. A *one-parameter subgroup* of G is a morphism of algebraic groups $\sigma\colon \mathbb{C}^\times \to G$. For a representation of G on a vector space V, we say that a subset $S \subseteq V$ is G-stable if $g \cdot s \in S$ for all $g \in G$, $s \in S$.

▶ **Theorem 2.3** (Hilbert–Mumford criterion, [35, 55])**.** *Let $\rho\colon G \to \mathrm{GL}(V)$ be a rational representation of a reductive group G. Suppose $S \subseteq V$ is a G-stable closed subvariety of V and let $v \in V$ such that $\overline{O_v} \cap S \neq \emptyset$. Then there exists a one-parameter subgroup $\sigma\colon \mathbb{C}^\times \to G$ such that $\lim_{\epsilon \to 0} \sigma(\epsilon) \cdot v \in S$.*

A particular use of the above theorem is to take $S = \{0\}$ or $S = O_{\widetilde{v}}$. When G is a torus, the set of one-parameter subgroups has the structure of a $\mathbb{Z}$-lattice. We will discuss this further in the next section.

We end this section by introducing a key notion in invariant theory called the *null cone*, whose significance will become clear in later sections. For a collection F of polynomials in $\mathbb{C}[V]$, we denote by $\mathbb{V}(F)$ their common zero locus in V.

▶ **Definition 2.4** (Null cone)**.** *Let $\rho\colon G \to \mathrm{GL}(V)$ be a rational representation of a reductive group G. Then the* null cone *is defined as*

$$\mathcal{N}_G(V) := \mathcal{N}(\rho) := \{v \in V \mid 0 \in \overline{O_v}\}.$$

It can also be defined as the common zero locus of all invariant polynomials without constant part:

$$\mathcal{N}_G(V) := \mathcal{N}(\rho) := \mathbb{V}(\bigcup_{d>0} \mathbb{C}[V]^G_d),$$

where $\mathbb{C}[V]^G_d$ denotes the space of invariant polynomials that are homogeneous of degree d. The equivalence of the two definitions of the null cone follows from Theorem 2.1.

3 Invariants and orbit closures of torus actions

Invariant theory for general reductive groups can get very complicated. However, for representations of tori, that is, *commutative* connected reductive groups, a lot of the theory can be viewed as a combination of linear algebra and the study of convex polytopes. We will collect important results regarding torus actions in this section and refer the reader to [61, 17] for more details. All the results in this section are already known or can be deduced from the existing literature, and we provide proof sketches for completeness. Note that tori are reductive groups, so the results of the previous section hold in this setting.

We will first briefly recall torus actions and the notions of characters/weights, one-parameter subgroups and how weight matrices define a representation. Then, we give a linear algebraic description of invariant rings by determining the monomials that are invariant. Then, we describe a polyhedral perspective on orbits. In particular given a point v in the vector space of the representation, we define a polyhedral cone, called the Newton cone. The Newton cone can be used to determine whether v is in the null cone and moreover we give a correspondence between the faces of the Newton cone to orbits in the orbit closure of v, which is crucial in understanding the orbit closure containment problem.

For this entire section, fix a torus $T = (\mathbb{C}^\times)^d$.[14]

3.1 Representations and invariants

As described in Section 1.3, any representation of a torus T is a "scaling" action (after identifying V with $\mathbb{C}^n$ by an appropriate choice of basis). Namely, each coordinate of $v \in \mathbb{C}^n$ is multiplied by some (Laurent) monomial $\prod_{i=1}^d t_i^{\lambda_i}$ for integers $\lambda_i \in \mathbb{Z}$. These monomials (succinctly described by the so-called weight matrix, see below) together specify the representation. We now make this more precise.

A 1-dimensional (rational) representation is called a *character* or a *weight*. Let $\mathcal{X}(T)$ denote the set of weights of T, which forms a group where the binary operation is (pointwise) multiplication of functions. To each $\lambda = (\lambda_1, \dots, \lambda_d) \in \mathbb{Z}^d$, we associate a weight, also denoted λ by slight abuse of notation, namely

$$\lambda \colon T \to \mathbb{C}^\times, \quad \lambda(t) = \prod_{i=1}^d t_i^{\lambda_i},$$

which gives an identification of abelian groups $\mathbb{Z}^d \cong \mathcal{X}(T)$.

Let $\rho \colon T \to \mathrm{GL}(V)$ be a (rational) representation of T where V is an n-dimensional vector space. We can choose a basis of V consisting of weight vectors, wherein a vector $v \in V$ is called a *weight vector* of weight $\lambda \in \mathcal{X}(T)$ if $t \cdot v = \lambda(t)v$ for all $t \in T$. Once we have chosen a weight basis, using the identification $\mathcal{X}(T) \cong \mathbb{Z}^d$, the corresponding n weights can be collected into a $d \times n$ matrix with integer entries, which we call the *weight matrix* of the representation. Up to permutation of the columns, it is independent of the choice of weight basis, and it classifies the representation up to isomorphism. Concretely, a matrix $M = (m_{ij}) \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ describes the representation $\rho_M \colon T \to \mathrm{GL}_n(\mathbb{C})$ defined in (1). That is, for $t = (t_1, \dots, t_d)$ and $v = (v_1, \dots, v_n) \in \mathbb{C}^n$, we have

[14] Any commutative connected reductive group is isomorphic to some $(\mathbb{C}^\times)^d$. Important examples include T_d, the group of diagonal $d \times d$ invertible matrices and its subgroup ST_d consisting of diagonal matrices with determinant 1.

$$t \cdot v = \rho_M(t)v = \left(\left(\prod_{i=1}^{d} t_i^{m_{i1}} \right) v_1, \left(\prod_{i=1}^{d} t_i^{m_{i2}} \right) v_2, \ldots, \left(\prod_{i=1}^{d} t_i^{m_{in}} \right) v_n \right).$$

The matrix M is the weight matrix for this action. The j^{th} standard basis vector e_j is a weight vector of weight $m^{(j)} = (m_{1j}, m_{2j}, \ldots, m_{dj}) \in \mathbb{Z}^d = \mathcal{X}(T)$. Note that $m^{(j)}$ is the j^{th} column vector of M.

For the rest of this section, we fix an n-dimensional representation $\rho_M \colon T \to \mathrm{GL}_n(\mathbb{C})$ of the torus $T = (\mathbb{C}^\times)^d$ given by a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ with columns $m^{(j)}$ for $j \in [n]$. The following well-known result describes the invariant ring of this action (see, e.g., [22, Section 3]):

▶ **Proposition 3.1.**

1. *Let $c \in \mathbb{Z}_{\geq 0}^n$. A monomial $x^c = \prod_j x_j^{c_j}$ is invariant if and only if $\sum_j c_j m^{(j)} = 0$;*
2. *The invariant ring $\mathbb{C}[x_1, \ldots, x_n]^T$ is spanned as a vector space by the invariant monomials.*

Proof. For the action ρ of G on V, there is a natural induced action of G on the ring of polynomial functions $\mathbb{C}[V]$ defined by the formula $g \cdot f(v) := f(\rho(g)^{-1}v)$. Applying this for the action ρ_M, we get an induced action of T on $\mathbb{C}[x_1, \ldots, x_n]$. It is easy to compute this action: for a monomial x^c and $t \in T$, we have $t \cdot x^c = \lambda(t)^{-1} x^c$, where $\lambda \in \mathcal{X}(T)$ is the character corresponding to $\sum_j c_j m^{(j)} \in \mathbb{Z}^d$. It follows that the monomials which are invariant are precisely the ones for which $\sum_j c_j m^{(j)} = 0$, the trivial character, proving the first part. The second part follows from the observation that a polynomial is invariant if and only if each monomial that occurs in it is invariant. ◀

Part (1) of Proposition 3.1 shows that the invariant monomials are in one-to-one correspondence with the nonnegative integer vectors in the kernel of the weight matrix. Accordingly, they form a semigroup. In general, such semigroups can have a large number of generators, which explains the difficulty of using polynomial invariants [24]. Our key idea to obtain efficient algorithms will be to instead consider invariant Laurent monomials, which form a lattice rather than a semigroup. We will return to this in Section 4.

In turns out that the weights lead to a strong link to convex polyhedral geometry, which in turn characterizes the orbits in an orbit closure. For this, we make the following definitions. The *support* of a vector $v \in \mathbb{C}^n$ is defined as

$$\mathrm{supp}(v) := \{j \in [n] \mid v_j \neq 0\}.$$

Let us record some of the properties of the support. By dimension (of an orbit, orbit closure, algebraic group, etc), we mean the dimension of the underlying variety.

▶ **Lemma 3.2.** *For $v, w \in \mathbb{C}^n$ we have:*

1. *If $O_v = O_w$, then* $\mathrm{supp}(v) = \mathrm{supp}(w)$.
2. *If* $\mathrm{supp}(v) = \mathrm{supp}(w)$*, then* $\dim O_v = \dim O_w$.
3. *If $w \in \overline{O_v}$, then* $\mathrm{supp}(w) \subseteq \mathrm{supp}(v)$. *This inclusion is strict if and only if $w \in \overline{O_v} \setminus O_v$.*

Proof. (1) is clear, since each coordinate simply gets rescaled by a nonzero number by the group action. For (2) we note that the stabilizer group $\mathrm{stab}(v)$ of v only depends on $\mathrm{supp}(v)$. The claim follows using $\dim O_v = d - \dim \mathrm{stab}(v)$. For (3), the inclusion of supports holds since taking limits can never increase the support. Finally, it is known [38, §8.3] that $\overline{O_v} \setminus O_v$ is a Zariski closed subset of dimension strictly less than $\dim O_v$. Hence $w \in \overline{O_v} \setminus O_v$ implies $\dim O_w < \dim O_v$ and therefore $\mathrm{supp}(w) \subsetneq \mathrm{supp}(v)$ by part (2). ◀

3.2 Newton cone and orbit closures

We define the *Newton cone* $C(v)$ of a vector $v \in \mathbb{C}^n$ to be the rational polyhedral cone generated by the weights corresponding to the indices in the support, that is,

$$C(v) := \Big\{ \sum_{j \in \operatorname{supp}(v)} c_j m^{(j)} \mid c_j \geq 0 \Big\} \subseteq \mathbb{R}^d.$$

The *lineality space* of the cone $C(v)$ is defined as $L(v) := C(v) \cap (-C(v))$. Clearly, it is the largest linear subspace contained in $C(v)$. The cone $C(v)$ is called *pointed* iff $L(v) = 0$. (Compare [57] for the structure of polyhedral cones.)

These notions are standard in geometric programming, which essentially studies optimization problems associated with torus actions, albeit often with a different representation and motivation; see, e.g., [11] and references therein. The connection is particularly apparent and useful in the study of polynomial capacities which have important applications to approximate counting [50, 34].

We will see that the Newton cone contains all the information about the orbits contained in an orbit closure. To start, we show that membership in the null cone can be characterized as follows. Define the *essential support* of a vector $v \in V$ as

$$\text{e-supp}(v) := \{ j \in \operatorname{supp}(v) \mid m^{(j)} \in L(v) \}. \tag{3}$$

▶ **Lemma 3.3.** *Let $k \in \operatorname{supp}(v)$. We have $k \in \text{e-supp}(v)$ if and only if there exists an invariant monomial $\prod_{j \in \operatorname{supp}(v)} x_j^{c_j}$ with $c_j \in \mathbb{Z}_{\geq 0}$ such that $c_k > 0$.*

Proof. It is easy to see that $m^{(k)} \in L(v)$ if and only if there is a non-negative integral linear combination $\sum_{j \in \operatorname{supp}(v)} c_j m^{(j)} = 0$ with $c_k > 0$. By Proposition 3.1, this is equivalent to the existence of an invariant monomial $\prod_{j \in \operatorname{supp}(v)} x_j^{c_j}$ with $c_j \in \mathbb{Z}_{\geq 0}$ such that $c_k > 0$. ◀

▶ **Corollary 3.4.** *We have that v is in the null cone $\mathcal{N}(\rho_M)$ if and only if* e-supp$(v) = \emptyset$.

Equivalently, v is in the null cone if and only if $C(v)$ is pointed and $m^{(j)} \neq 0$ for all $j \in \operatorname{supp}(v)$.

In fact, much more can be said. Let us first recall the notion of faces of polyhedral cones. If $C(v)$ is contained in a closed halfspace H_+ of $\mathbb{R}^d$ bounded by a linear hyperplane H, then we call the intersection $F = H \cap C(v)$ a *face* of $C(v)$ when it is non-empty. The cone itself is also considered a face of $C(v)$: by definition, it is the largest face of $C(v)$. On the other hand, each face of $C(v)$ must contain the lineality space $L(v)$, which is therefore the smallest face of $C(v)$.

We will see shortly that the faces of $C(v)$ are in bijective correspondence with the orbits contained in $\overline{O_v}$. For this, we need to introduce some more notation. For a subset $J \subseteq \operatorname{supp}(v)$, we define the *restriction* $v|_J$ to be the vector with entries

$$(v|_J)_j = \begin{cases} v_j & \text{if } j \in J, \\ 0 & \text{otherwise,} \end{cases}$$

as its j-th coordinate. Let now F be a face of $C(v)$ defined by a closed half-space $H_+ = \{y \in \mathbb{R}^d \mid \nu \cdot y \geq 0\}$ for some $\nu \in \mathbb{R}^d$, that is,

$$F = \{ y \in C(V) \mid \nu \cdot y = 0 \}.$$

Since $C(v)$ is rational, we may assume that ν has integer components. We assign to F the subset of indices

$$S_F := \{j \in \operatorname{supp}(v) \mid m^{(j)} \in F\}$$

and define $v_F := v|_{S_F}$. Let us check that the orbit O_{v_F} of v_F is contained in $\overline{O_v}$. The one-parameter subgroup $\sigma\colon \mathbb{C}^\times \to T$ given by $\sigma(\epsilon) = (\epsilon^{\nu_1}, \ldots, \epsilon^{\nu_d})$ satisfies

$$\sigma(\epsilon) \cdot v = \rho_M(\sigma(\epsilon))v = (\epsilon^{\nu \cdot m^{(1)}} v_1, \ldots, \epsilon^{\nu \cdot m^{(n)}} v_n). \tag{4}$$

It follows that $\lim_{\epsilon \to 0} \sigma(\epsilon) \cdot v = v_F$ and hence $v_F \in \overline{O_v}$. The same reasoning shows that $v_F \in \overline{O_{v_{F'}}}$ if F is a face contained in the face F'.

The following result is well known, see e.g., [56, Example 1.3], but we sketch a proof for completeness.

▶ **Proposition 3.5.** *The map $F \mapsto O_{v_F}$ is a bijection between the set of faces of $C(v)$ and the set of orbits contained in $\overline{O_v}$. Moreover, we have*

$$F \subseteq F' \iff \overline{O_{v_F}} \subseteq \overline{O_{v_{F'}}}\ .$$

The proof of surjectivity relies on a strengthening of the Hilbert–Mumford criterion (Theorem 2.3). Recall this states that if we consider a closed subset S that is stable under the group action and intersects the orbit closure of some point v, then there is a one-parameter subgroup that will drive v to a point in S in the limit. However, a subtle point is that this requires S to be closed. In general, orbits are not closed, so a point w could be in the orbit closure of a point v, but the orbit of w may not be closed. In this case, Theorem 2.3 does not apply to $S = O_w$, and indeed the orbit of w need not be reachable from v by a limit of a one-parameter subgroup. The following theorem shows that for torus actions such a phenomenon does not happen. This crucial fact will also prove useful for us algorithmically in Section 7.

▶ **Theorem 3.6** ([46], Kapitel III.2.2). *Let $\rho\colon T \to \operatorname{GL}(V)$ be a rational representation. Suppose $v, w \in V$ are such that $w \in \overline{O_v}$. Then there exists a one-parameter subgroup $\sigma\colon \mathbb{C}^\times \to T$ such that*

$$\lim_{\epsilon \to 0} \sigma(\epsilon) \cdot v \in O_w.$$

Before we prove Proposition 3.5, we discuss a bit about the structure of one-parameter subgroups. For each $\nu \in \mathbb{Z}^d$, we define a one-parameter subgroup of T, namely $\sigma\colon \mathbb{C}^\times \to T$ defined by $\sigma(\epsilon) = (\epsilon^{\nu_1}, \ldots, \epsilon^{\nu_d})$. Any one-parameter subgroup of T is of this form. This gives an identification of abelian groups $\mathbb{Z}^d \cong \mathcal{Y}(T)$, where $\mathcal{Y}(T)$ denotes the collection of all one-parameter subgroups of T.

We leave the proof of the following well known lemma to the reader.

▶ **Lemma 3.7.** *Let $\sigma\colon \mathbb{C}^\times \to T$ be a one-parameter subgroup, so $\sigma(\epsilon) = (\epsilon^{\nu_1}, \ldots, \epsilon^{\nu_d})$ for some $\nu \in \mathbb{Z}^d$, and let $v \in \mathbb{C}^n$.*

1. *The limit $\lim_{t \to 0} \sigma(t) \cdot v$ exists if and only if $m^{(j)} \cdot \sigma \geq 0$ for all $j \in \operatorname{supp}(v)$.*
2. *If the limit exists, then $\lim_{t \to 0} \sigma(t) \cdot v = v|_S$, where $S = \{j \in \operatorname{supp}(v) \mid m^{(j)} \cdot \sigma = 0\}$.*

Proof of Proposition 3.5. We have already verified that O_{v_F} is an orbit contained in $\overline{O_v}$, hence $F \mapsto O_{v_F}$ is well-defined as a map from the set of faces of $C(v)$ to the set of orbits contained in $\overline{O_v}$. To see that it is injective, note that F is the cone generated by

$\operatorname{supp}(v_F) = S_F$. For surjectivity, let O_w be an orbit contained in $\overline{O_v}$ and $\sigma\colon \mathbb{C}^\times \to T$ be a one-parameter subgroup as in Theorem 3.6. There is $\nu \in \mathbb{Z}^d$ such that $\sigma(\epsilon) = (\epsilon^{\nu_1}, \dots, \epsilon^{\nu_d})$. By Lemma 3.7, the existence of $\lim_{\epsilon\to 0} \sigma(\epsilon)\cdot v$ means that $\nu \cdot m^{(j)} \geq 0$ for all $j \in \operatorname{supp}(v)$. In other words, $C(v)$ is contained in the halfspace $\{y \in \mathbb{R}^d \mid \nu \cdot y \geq 0\}$. Moreover, the limit equals v_F, where F is the face $F := \{y \in C(v) \mid \nu \cdot y = 0\}$ of $C(v)$. Therefore, $v_F \in O_w$, hence $O_{v_F} = O_w$, and we have shown surjectivity.

In order to show the remaining equivalence, recall that we argued below (4) that if $F \subseteq F'$ then $v_F \in \overline{O_{v_{F'}}}$. The preceding argument also implies the converse. ◀

As an immediate consequence of Proposition 3.5, we get the following result, which not only reproves Lemma 3.3 but also characterizes the closed orbit in an orbit closure. For this, define

$$\widetilde{v} := v|_{L(v)} = v|_{\text{e-supp}(v)}.$$

▶ **Corollary 3.8.** *The orbit $O_{\widetilde{v}}$ corresponding to the lineality space $L(v)$ is contained in every orbit closure contained in $\overline{O_v}$. Therefore, it is the unique closed orbit contained in $\overline{O_v}$.*

In particular, the orbit O_v is closed if and only if $C(v) = L(v)$, i.e., $C(v)$ equals its linear span. Moreover, v is in the null cone if and only if e-supp$(v) = \emptyset$.

4 Generating Laurent polynomials and rational invariants

In this section, we discuss the computation of suitable rational invariants, which is the heart of our algorithms, and the main novelty of this paper. As explained in the introduction, the starting point is the simple observation that two orbits can only be equal when they have the same support (Lemma 3.2). But once we restrict to vectors of fixed support, it is natural to consider a larger class of invariants, namely Laurent polynomials, which are polynomials that can also have negative exponents. In Section 4.1 we will see that the invariant Laurent polynomials for a given support naturally form a lattice that can be computed from the weight matrix. This allows us to give an efficient algorithm for computing small sets of generators. As a consequence, we can also efficiently compute a system of generating rational invariants.

For the rest of this section, we fix an n-dimensional representation $\rho_M\colon T \to \mathrm{GL}_n(\mathbb{C})$ of the torus $T = (\mathbb{C}^\times)^d$ given by a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ with columns $m^{(j)}$ for $j \in [n]$.

4.1 Invariant Laurent polynomials

For $S \subseteq [n]$, consider the set of vectors with support S, that is, the variety

$$X_S = \{v \in \mathbb{C}^n \mid \operatorname{supp}(v) = S\} = \{v \in \mathbb{C}^n \mid v_j \neq 0 \text{ if and only if } j \in S\}. \tag{5}$$

The ring of regular functions on X_S, denoted $\mathbb{C}[X_S]$, is naturally identified with the ring of Laurent polynomials in variables $\{x_j\}_{j\in S}$. That is,

$$\mathbb{C}[X_S] = \mathbb{C}[x_j, x_j^{-1} \mid j \in S].$$

We observe that ρ_M restricts to an action of T on X_S, and induces an action on $\mathbb{C}[X_S]$. The proposition below shows that the algebra $\mathbb{C}[X_S]^T$ of *invariant Laurent polynomials* can be succinctly described in terms of the lattice

$$L_S = \Big\{c \in \mathbb{Z}^S \mid \sum_{j\in S} c_j m^{(j)} = 0\Big\} = \ker(M_S) \cap \mathbb{Z}^{|S|}, \tag{6}$$

where $\mathbb{Z}^S := \{c \in \mathbb{R}^n \mid c_j = 0 \text{ for all } j \notin S\} \cong \mathbb{Z}^{|S|}$, and M_S denotes the submatrix of the weight matrix M, obtained by removing all columns except those labeled by S.

▶ **Proposition 4.1.**

1. *Let $c \in \mathbb{Z}^S$. A Laurent monomial $x^c = \prod_{j \in S} x_j^{c_j}$ is invariant if and only if $c \in L_S$.*
2. *The algebra of invariant Laurent polynomials $\mathbb{C}[X_S]^T$ is spanned as a vector space by the invariant Laurent monomials.*
3. *If $\{c^{(1)}, c^{(2)}, \dots, c^{(r)}\}$ is a lattice basis of L_S, then $\mathbb{C}[X_S]^T$ is generated as an algebra by the invariant Laurent monomials $\{x^{c^{(1)}}, \dots, x^{c^{(r)}}\}$.*

Proof. The first two parts are shown using an argument similar to the proof of Proposition 3.1. The third statement is an immediate consequence. ◀

It is instructive to compare this with the discussion below Proposition 3.1, where we saw that the invariant polynomials are similarly described by the *semigroup* of nonnegative vectors in the kernel of the weight matrix. By working with vectors of fixed support, we instead obtain a natural lattice structure, which simplifies the situation considerably. For example, the lattice L_S and hence the algebra of invariant Laurent polynomials $\mathbb{C}[X_S]^T$ have at most $|S| \leq n$ generators – in stark contrast to the situation for invariant polynomials.

We now discuss how to compute lattice bases as in Proposition 4.1. It is well known that every integer matrix M can be diagonalized by multiplying from left and right with unimodular matrices. This is known as the *Smith normal form* [59]. The Smith normal form can be computed in polynomial time [43]. We record these facts in the following theorem.

▶ **Theorem 4.2** (Smith normal form). *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$. Then, there exist unimodular matrices $U \in \mathrm{Mat}_{d,d}(\mathbb{Z}), W \in \mathrm{Mat}_{n,n}(\mathbb{Z})$ such that*

$$UMW = \begin{bmatrix} \alpha_1 & 0 & 0 & & \dots & & 0 \\ 0 & \alpha_2 & 0 & & \dots & & 0 \\ 0 & 0 & \ddots & & & & 0 \\ & & & \alpha_r & & & \vdots \\ \vdots & \vdots & & & 0 & & \\ & & & & & \ddots & \\ 0 & 0 & 0 & \dots & & & 0 \end{bmatrix}$$

and the diagonal elements satisfy $\alpha_i \mid \alpha_{i+1}$ for $i = 1, 2, \dots, r-1$, where r equals the rank of M. The matrix UMW is unique and called the Smith normal form of M.

Moreover, if the bit-lengths of the entries of M are bounded by b, then the matrices U, W, and UMW can be computed in $\mathrm{poly}(d, n, b)$-time.

Using the Smith normal form it is easy to compute a basis of the lattice L_S. We state this in the following algorithm and corollary.

■ **Algorithm 1** Computation of a basis of the lattice of invariant Laurent monomials.

Input $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ and $S \subseteq [n]$.
Step 1 Compute the submatrix M_S of M obtained by deleting all columns except those in S.
Step 2 Compute the Smith normal form UM_SW of M_S (as in Theorem 4.2).
Step 3 Return $\{w^{(r+1)}, w^{(r+2)}, \dots, w^{(n)}\}$, where $w^{(j)}$ denotes the j^{th} column of W.

▶ **Corollary 4.3.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ and $S \subseteq [n]$, and suppose the bit-lengths of the entries of M are bounded by b. Then Algorithm 1 computes a basis for the lattice L_S defined in* (6) *in* $\mathrm{poly}(d,n,b)$*-time. In particular, each $w^{(j)}$ has bit-length* $\mathrm{poly}(d,n,b)$.

Alternatively, one can use lattice algorithms; we refer the interested reader to [31, Corollary 5.4.10].

▶ Remark 4.4. It is easy to see that given an exponent vector $c = (c_1, \dots, c_n) \in \mathbb{Z}^n_{\geq 0}$, where the bit-lengths of the c_is are bounded by b, an arithmetic circuit computing the monomial x^c of size $\mathrm{poly}(n,b)$ can be constructed in $\mathrm{poly}(n,b)$-time. Similarly, if $c \in \mathbb{Z}^n$, an arithmetic circuit with division computing the Laurent monomial x^c can be constructed in $\mathrm{poly}(n,b)$-time.

Proof of Theorem 1.3. This follows from Proposition 4.1, Corollary 4.3, and Remark 4.4. ◀

4.2 Rational invariants

In the remainder of this section we will discuss rational invariants. For $V = \mathbb{C}^n$, recall that $\mathbb{C}[V] = \mathbb{C}[x_1, \dots, x_n]$ is the polynomial ring in n variables. Let $\mathbb{C}(V) = \mathbb{C}(x_1, \dots, x_n)$ the field of rational functions (its fraction field). In other words, any element in $\mathbb{C}(V)$ is a ratio of two polynomials. The action of T on $\mathbb{C}[V]$ extends to $\mathbb{C}(V)$. Then $\mathbb{C}(V)^T$ is the field of *rational invariants.* Clearly, any invariant Laurent polynomial is a rational invariant, but the converse need not be the case.

Nevertheless, we can show that the invariant Laurent polynomials in all variables (that is, for support $S = [n]$) generate the rational invariants as a field.

▶ **Proposition 4.5.** *Let $A := \mathbb{C}[X_{[n]}] = \mathbb{C}[x_1, x_1^{-1}, \dots, x_n, x_n^{-1}]^T$ denote the algebra of invariant Laurent polynomials, and let $F := \mathbb{C}(x_1, \dots, x_n)^T$ denote the field of rational invariants. Then, A generates F as a field, i.e., the field of fractions of A is F.*

Proof. Let $f \in F^\times$ and write $f = \frac{p}{q}$, where $p, q \in \mathbb{C}[x_1, \dots, x_n]$ have no common factors. Since f is invariant, we have for any $t \in T$ that

$$\frac{t \cdot p}{t \cdot q} = t \cdot f = f = \frac{p}{q}.$$

Accordingly, $t \cdot p = \alpha(t)p$ and $t \cdot q = \alpha(t)q$ for some $\alpha(t) \in \mathbb{C}^\times$. Thus, p and q span one-dimensional representations. This in turn implies that $\alpha \colon T \to \mathbb{C}^\times$ is a character, as discussed in Section 3.1, and further that p (and also q) is a sum of monomials with the same weight, i.e., $p = \sum_e p_e x^e$ such that $t \cdot x^e = \alpha(t)x^e$ for $p_e \neq 0$. In particular, $f_e = \frac{q}{x^e}$ is a Laurent polynomial *invariant* if $p_e \neq 0$, and we can write

$$f = \frac{p}{q} = \sum_e p_e \frac{x^e}{q} = \sum_e p_e \frac{1}{f_e},$$

which concludes the proof. ◀

As a direct consequence, any system of generating invariant Laurent polynomials (as an algebra) also serves as a system of generating rational invariants (as a field extension of $\mathbb{C}$). Thus we obtain:

Proof of Corollary 1.4. This follows from Theorem 1.3 (with $S = [n]$) and Proposition 4.5. ◀

5 Orbit equality problem

In this section, we will give a polynomial time algorithm for the orbit equality problem. Given two points, the strategy is to compute a small collection of invariant Laurent monomials (using the result of Section 4) whose evaluations at the two given points will determine whether the two points are in the same orbit. The efficient testing of whether two Laurent monomials evaluate to the same value actually requires an idea: this has already been studied in the literature and we briefly sketch in Section 5.1 how to do this.

We still assume an n-dimensional representation $\rho_M : T \to \mathrm{GL}_n(\mathbb{C})$ of the torus $T = (\mathbb{C}^\times)^d$ given by a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ with columns $m^{(j)}$ for $j \in [n]$.

In general, invariants can only decide orbit closure intersection, not orbit equality. However, the crucial point is that in the varieties (5) consisting of vectors of fixed support any T-orbit is closed.

▶ **Proposition 5.1.** *Let $S \in [n]$, X_S be the variety defined in* (5)*, and $v \in X_S$. Then the orbit O_v is a closed subset of X_S.*

Proof. By Lemma 3.2 (3) we have $O_v = \overline{O_v} \cap X_S$ which implies that the orbits are closed in X_S. ◀

Orbit equality in V can always be reduced to orbit equality in some X_S, since equality of supports is a necessary condition (Lemma 3.2 (1)). The importance of the above result is that the latter orbit equality and orbit closure intersection are equivalent in X_S. Together with Theorem 2.1 we obtain the following result.

▶ **Corollary 5.2.** *Suppose* $\mathrm{supp}(v) = \mathrm{supp}(w) = S$*. Then, $O_v \neq O_w$ if and only if there is an invariant Laurent monomial $f = \prod_{j \in S} x_j^{c_j}$ such that $f(v) \neq f(w)$.*

Thus, we obtain the following algorithm for the orbit equality problem.

Algorithm 2 Deciding orbit equality.

```
Input  $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ and $v, w \in \mathbb{Q}(i)^n$.
Step 1  Check if $\mathrm{supp}(v) = \mathrm{supp}(w)$. If not, $O_v \neq O_w$, so we can stop.
Step 2  Use Algorithm 1 to compute a lattice basis $\mathcal{B}$ for the lattice $L_S$ defined in (6).
Step 3  For each $e \in \mathcal{B}$, we check if $v^e = w^e$ (as described in Section 5.1 below).
    If they are all equal, then $O_v = O_w$. Else, $O_v \neq O_w$.
```

Proof of Theorem 1.2, part (1). The correctness of Algorithm 2 follows from Proposition 4.1 and Corollary 5.2. We now analyze its runtime. Clearly, the first step can be implemented efficiently. For the second step, we can appeal to Corollary 4.3. For step 3, we first observe that, again by Corollary 4.3, the exponents e have bit-length $\mathrm{poly}(d, n, b)$. Then Proposition 5.4 below shows that this step can also be implemented in time $\mathrm{poly}(d, n, b)$. ◀

5.1 Laurent monomial equivalence

We now discuss how to test if a Laurent monomial x^e evaluates to the same value at two points v and w. In our context, where each component e_j of the exponent vector $e = (e_1, \ldots, e_n)$ has poly-sized bit-lengths, it is unreasonable to evaluate the Laurent monomials explicitly, because the answer may very well require exponentially large bit-length. Yet, it is possible to check if $v^e = w^e$ efficiently. We describe a simple algorithm based on g.c.d.'s, which has appeared before (see, for example, [25]) in the case where the entries of v and w are in $\mathbb{Z}$ (or equivalently $\mathbb{Q}$). The result is much older; for example, it follows from the results in [3], as mentioned in [30], which gives a generalization to number fields.[15] Here we present a short self-contained proof and then follow up with the rather simple extension to Gaussian rationals.

▶ **Lemma 5.3.** *Suppose $a_1, \ldots, a_k, b_1, \ldots, b_r \in \mathbb{Q}$ and $e_1, \ldots, e_k, f_1, \ldots, f_r \in \mathbb{Z}$ have bit-lengths at most s. Then, in $\operatorname{poly}(k, r, s)$-time, we can decide if $\prod_{i=1}^k a_i^{e_i} = \prod_{j=1}^r b_j^{f_j}$.*

Proof. By clearing denominators, we may assume that $a_1, \ldots, a_k, b_1, \ldots, b_r$ are integers. By moving terms to the other side, we can further assume w.l.o.g.that all $e_i, f_j \geq 0$. Pick some a_l and some b_m that are not coprime. Then, consider $d = \gcd(a_l, b_m) \geq 2$. W.l.o.g., we can assume $e_l \geq f_m$. Then, test if $d^{e_l - f_m}(a_l')^{e_l} \prod_{i \neq l} a_i^{e_i} = (b_m')^{f_m} \prod_{j \neq m} b_j^{f_j}$, where $a_l' = a_l/d$ and $b_m' = b_m/d$. This is an iterative procedure which stops when each a_i is coprime to b_j. At which point, unless all a_i's and b_j's are equal to 1, both sides cannot be equal.

The question is how long does such an iterative procedure take. Consider the quantity $P := |a_1 \cdots a_k b_1 \cdots b_r|$. After applying one step, the resulting quantity P' satisfies $P' = P/d^2 \leq P/4$. Since initially, P is $2^{\operatorname{poly}(k,r,s)}$-sized, there are at most a polynomial number of iterative steps. Hence, the entire procedure takes $\operatorname{poly}(k, r, s)$-time. ◀

An analogous result with the same proof holds for the ring $\mathbb{Z}[i]$ of Gaussian integers and its quotient field $\mathbb{Q}(i)$ of Gaussian rationals, using that this ring has unique factorization into irreducible elements. In the following proposition, we assume that a Gaussian rational $a = \alpha + i\beta \in \mathbb{Q}(i)$ is described by giving the encodings of α and β in binary.

▶ **Proposition 5.4.** *Suppose $a_1, \ldots, a_k, b_1, \ldots, b_r \in \mathbb{Q}(i)$ and $e_1, \ldots, e_k, f_1, \ldots, f_r \in \mathbb{Z}$ all have bit-lengths bounded by s. Then, in $\operatorname{poly}(k, r, s)$-time, we can decide if $\prod_{i=1}^k a_i^{e_i} = \prod_{j=1}^r b_j^{f_j}$.*

▶ Remark 5.5. For computational purposes, in many instances, numbers are described by their "floating point" representations. The floating point description of a Gaussian rational $a \in \mathbb{Q}(i)$ is described by giving the binary encodings of $\alpha, \beta \in \mathbb{Q}$ and $p \in \mathbb{Z}$ such that $a = (\alpha + i\beta)2^p$. If we assume that $a_1, \ldots, a_k, b_1, \ldots, b_r \in \mathbb{Q}(i)$ in the proposition above are given by their floating point descriptions, we can still decide monomial equivalence in polynomial time. Indeed, if we write each $a_j = (\alpha_j + i\beta_j)2^{p_j}$ and $b_j = (\gamma_j + i\delta_j)2^{q_j}$, then deciding whether $\prod_{j=1}^k a_j^{e_j} = \prod_{j=1}^r b_j^{f_j}$ simplifies to deciding if

$$\left(\prod_{j=1}^k (\alpha_j + i\beta_j)^{e_j}\right) \cdot 2^{\sum_{j=1}^k e_j p_j} = \left(\prod_{j=1}^r (\gamma_j + i\delta_j)^{f_j}\right) \cdot 2^{\sum_{j=1}^r f_j q_j},$$

which can again be interpreted as an instance of Proposition 5.4 and hence can be checked in polynomial time. Since all other computations in our algorithms only involve supports of vectors, it follows that all results in this paper generalize to this input model, as claimed in footnote 6.

An even easier special case arises for numbers of the form $a = 2^p$, with $p \in \mathbb{Q}$ specified by its binary encoding, as in the perfect matching application discussed in Section 1.4. Indeed, if $a_j = 2^{p_j}$ and $b_j = 2^{q_j}$ for $j \in [n]$, then deciding whether $\prod_{j=1}^k a_j^{e_j} = \prod_{j=1}^r b_j^{f_j}$ simply amounts to verifying whether $\sum_{j=1}^n p_j e_j = \sum_{j=1}^n q_j f_j$, which is clearly possible in polynomial time.

[15] In particular Ge's result [30] implies that Theorem 1.2 extends to the case where the entries of v and w are taken from some algebraic number field.

6 Orbit closure intersection and explicit separating invariants

In this section, we discuss how to solve the orbit closure intersection problem in polynomial time by efficiently reducing it to the orbit equality problem. The problem of orbit closure intersection has a manifestly analytic point of view, but also an algebraic point of view by Mumford's theorem, Theorem 2.1. In other words, when orbit closures of two points do not intersect, there is an invariant polynomial that takes different values on both points, serving as a "witness" to the fact that the orbit closures do not intersect. Accordingly, given two vectors whose orbit closures do not intersect, we also explain how to efficiently construct an arithmetic circuit which computes an invariant monomial separating the two vectors.

6.1 Reduction to orbit equality

The key idea is the following. Recall from Theorem 2.2 that any orbit closure $\overline{O_v}$ contains as unique closed orbit $O_{\tilde{v}}$, and that two orbit closures intersect if and only if they contain the same closed orbit. In Corollary 3.8, we showed that the unique closed orbit has a concrete polyhedral characterization: we can take $\tilde{v} = v|_{\text{e-supp}(v)}$, the restriction of the vector v to its essential support. Accordingly, the map $v \mapsto \tilde{v}$ provides a reduction of the orbit closure intersection problem for ρ_M to the the orbit equality problem for ρ_M. The following lemma shows that the essential support (and hence the reduction map) can be computed in polynomial time by using linear programming.

▶ **Lemma 6.1.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ define an n-dimensional representation of $T = (\mathbb{C}^\times)^d$, and let $v \in \mathbb{C}^n$. For $k \in \operatorname{supp}(v)$, we have $k \in \text{e-supp}(v)$ if and only if there is a non-negative linear combination $\sum_{j\in\operatorname{supp}(v)} c_j m^{(j)} = 0$ such that $c_k > 0$. If the bit-lengths of the entries of M are bounded by b, the latter can be decided in $\operatorname{poly}(d,n,b)$-time by using linear programming.*

Proof. The characterization follows from Proposition 3.1 and Lemma 3.3. It amounts to a basic decisional problem of linear programming, which is well known to be solvable in polynomial time, see [31]. ◀

The above proof also shows that a nonvanishing invariant monomial as in Lemma 6.1 can be computed in polynomial time. As explained above, we arrive at the following algorithm and results.

Algorithm 3 Reduction of orbit closure intersection to orbit equality.

Input $M \in \mathrm{Mat}_{d,n}(\mathbb{Z}), v, w \in \mathbb{Q}(i)^n$.

Step 1 Compute e-supp(v) in the following way: For each $k \in \operatorname{supp}(v)$, use linear programming to determine if there is a non-negative linear combination $\sum_{j\in\operatorname{supp}(v)} c_j m^{(j)} = 0$ with $c_k > 0$. The set e-supp(v) consists of all $k \in \operatorname{supp}(v)$ for which this is the case.

Step 2 Compute e-supp(w) in the same way.

Step 3 Return $\tilde{v} = v|_{\text{e-supp}(v)}$ and $\tilde{w} = w|_{\text{e-supp}(w)}$.

▶ **Corollary 6.2.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ describe an n-dimensional representation of $T = (\mathbb{C}^\times)^d$. Further, let $v, w \in \mathbb{Q}(i)^n$ and assume the bit-lengths of the entries of $M, v,$ and w are bounded by b. Then there is a $\operatorname{poly}(d,n,b)$-time reduction that reduces the problem of deciding $\overline{O_v} \cap \overline{O_w} \neq \emptyset$ to the problem of deciding if $O_{\widetilde{v}} = O_{\widetilde{w}}$, where $\tilde{v}$ and $\tilde{w}$ have bit-lengths bounded by b.*

Proof of Theorem 1.2, part (2). This follows from part (1), combined with Corollary 6.2. ◀

6.2 Explicit separating invariant

For torus actions, our reduction of orbit closure intersection to orbit equality will give us an invariant Laurent monomial that takes different values on the two points. But a separating invariant Laurent monomial itself does *not* serve as a witness (at least not naively, one needs further properties about the support of the Laurent monomial for it to serve as a witness). We now prove Corollary 1.5, which asserts that given two vectors we can nevertheless efficiently construct an arithmetic circuit which computes an invariant *monomial* separating them.

Proof of Corollary 1.5. We already noted that, by linear programming, we can compute the essential supports of v and w in $\operatorname{poly}(d,n,b)$-time. We distinguish two cases.

Case 1: $\text{e-supp}(v) \neq \text{e-supp}(w)$

Suppose $k \in \text{e-supp}(v) \setminus \text{e-supp}(w)$ without loss of generality. By Lemma 3.3 there is an invariant monomial $f = \prod_{j \in \operatorname{supp}(v)} x_j^{c_j}$ such that $c_k > 0$. Let us verify that $f(v) \neq f(w)$. We clearly have $f(v) \neq 0$. On the other hand, $f(w) = f(\widetilde{w}) = 0$, since $\widetilde{w} \in \overline{O_w}$, but k is not contained in $\operatorname{supp}(\widetilde{w}) = \text{e-supp}(w)$. So we indeed have $f(v) \neq f(w)$. In addition, we can find $(c_1, \ldots, c_n)$ in $\operatorname{poly}(d,n,b)$-time by linear programming (Lemma 6.1), so we can construct an arithmetic circuit for f in $\operatorname{poly}(d,n,b)$-time by Remark 4.4.

Case 2: $\text{e-supp}(v) = \text{e-supp}(w)$

Let $S := \text{e-supp}(v) = \text{e-supp}(w)$. We assume that $\overline{O_v} \cap \overline{O_w} = \emptyset$, which implies $O_{\widetilde{v}} \cap O_{\widetilde{w}} = \emptyset$. Thus, by Corollary 5.2, there is an invariant Laurent monomial $f = x^e$ with the property that $f(\widetilde{v}) \neq f(\widetilde{w})$, and hence $f(v) \neq f(w)$. Just like in Algorithm 2, we can in $\operatorname{poly}(d,n,b)$-time compute such an exponent vector $e \in \mathbb{Z}^n$, with bit-length of the e_i bounded above by $\operatorname{poly}(d,n,b)$.

Our goal is to produce an invariant monomial that separates v and w, so we need to modify f so as to get rid of the negative exponents. In the process, we must ensure that the bit-length of the circuit does not explode. By Lemma 3.3, for each $k \in S$, there exists $c^{(k)} \in \mathbb{Z}^n_{\geq 0}$ such that $\sum_{j \in \operatorname{supp}(v)} c_j^{(k)} m^{(j)} = 0$ and $c_k^{(k)} > 0$. We can compute $c^{(k)}$ in $\operatorname{poly}(d,n,b)$-time by linear programming. Let $m_k = x^{c^{(k)}}$ denote the corresponding invariant monomial. Put $S_- := \{j \in S \mid e_j < 0\}$. If $m_j(v) \neq m_j(w)$ for some $j \in S_-$, then m_j is an explicit separating invariant monomial and we are done by Remark 4.4. Assume now $m_j(v) = m_j(w)$ for all $j \in S_-$. Then $\widetilde{f} := x^d := f \cdot \prod_{j \in S_-} m_j^{-e_j}$ is a Laurent monomial that separates v and w. We verify now that the exponent vector d has non-negative entries. By construction, we have for $k \in S_-$,

$$d_k = e_k + (-e_k) c_k^{(k)} + \sum_{j \in S_-, j \neq k} (-e_j) \cdot c_k^{(j)} \geq 0,$$

since $e_k < 0$ and $e_j < 0$ for all $j \in S_-$, while $c_k^{(k)} \geq 1$, and $c_k^{(j)} \geq 0$. For $k \in [n] \setminus S_-$, we have

$$d_k = e_k + \sum_{j \in S_-} (-e_j) \cdot c_k^{(j)} \geq 0,$$

since $e_k \geq 0$ for $k \in S \setminus S_-$ and $e_k = 0$ for $k \notin S$, while $e_j < 0$ for $j \in S_-$. Altogether, we have shown that indeed all components of d are non-negative. We finally note that d can be computed in polynomial time, in particular, it has bit-length $\operatorname{poly}(d,n,b)$. So by Remark 4.4, we can construct an arithmetic circuit of size $\operatorname{poly}(d,n,b)$ that computes $\widetilde{f}$ in $\operatorname{poly}(d,n,b)$-time. ◀

7 Orbit closure containment

In this section, we discuss how to solve the the orbit closure containment problem in polynomial time by efficiently reducing it to the orbit equality problem.

The notion of orbit closure containment is in general quite tricky to capture. Polynomial invariants do not suffice, since two orbit closures can intersect (hence all polynomial invariants agree) with neither being contained in the other – this is precisely the difference between the orbit closure intersection and the orbit closure containment problem. Instead, the key idea for the reduction comes from one-parameter subgroups. We already discussed in Section 3 that if $w \in \overline{O_v}$ then O_w can be reached from v by a one-parameter subgroup. The following proposition gives a concrete polyhedral description of the relevant one-parameter subgroups.

▶ **Lemma 7.1.** *Let $M \in \operatorname{Mat}_{d,n}(\mathbb{Z})$ define an n-dimensional representation of $T = (\mathbb{C}^\times)^d$, and let $v, w \in \mathbb{C}^n$. Then $w \in \overline{O_v}$ if and only if there exists a one-parameter subgroup $\sigma \colon \mathbb{C}^\times \to T$, so $\sigma(\epsilon) = (\epsilon^{\nu_1}, \dots, \epsilon^{\nu_d})$ for some $\nu \in \mathbb{Z}^d$, such that*

1. *$\{j \in \operatorname{supp}(v) \mid m^{(j)} \cdot \nu = 0\} = \operatorname{supp}(w)$ and $m^{(k)} \cdot \nu > 0$ for all $k \in \operatorname{supp}(v) \setminus \operatorname{supp}(w)$;*
2. *$O_{(v|_{\operatorname{supp}(w)})} = O_w$.*

Proof. If $w \in \overline{O_v}$, then by Theorem 3.6, we know that there is a one-parameter subgroup σ such that $\lim_{t \to 0} \sigma(t)v \in O_w$. In particular this implies that $\lim_{t\to 0} \sigma(t)v$ has the same support as w and has the same orbit as w. Now, both (1) and (2) follow from Lemma 3.7.

For the converse, note that, again by Lemma 3.7, (1) implies that $\lim_{t\to 0} \sigma(t)v = v|_{\operatorname{supp}(w)} \in \overline{O_v}$, hence it follows that $O_w = O_{(v|_{\operatorname{supp}(w)})} \subseteq \overline{O_v}$ by (2). ◀

Now, we can give our algorithm to test if w is in the orbit closure of v.

Algorithm 4 Orbit closure containment.

Input $M \in \operatorname{Mat}_{d,n}(\mathbb{Z})$ and $v, w \in \mathbb{Q}(i)^n$.
Step 1 Check if $\operatorname{supp}(w) \subseteq \operatorname{supp}(v)$. If not, $w \notin \overline{O_v}$, so we can stop.
Step 2 Using linear programming, determine whether there exists a solution $y \in \mathbb{R}^d$ to the collection of linear equalities $m^{(j)} \cdot \nu = 0$ for each $j \in \operatorname{supp}(w)$ and linear inequalities $m^{(k)} \cdot \nu > 0$ for all $k \in \operatorname{supp}(v) \setminus \operatorname{supp}(w)$. If there is no solution, then $w \notin \overline{O_v}$, so we can stop.
Step 3 Use Algorithm 2 check whether $O_{(v|_{\operatorname{supp}(w)})} = O_w$. If yes, then $w \in \overline{O_v}$. Else, it is not.

Proof of Theorem 1.2, part (3). The correctness of Algorithm 4 follows from Lemma 7.1. Indeed, condition (1) in the lemma is satisfied if and only if the algorithm passes the first two steps, and then condition (2) is tested in the last step.

We still need to argue about the efficiency of the algorithm. Clearly, Step 1 can be done in linear time. Step 2 can be done in $\operatorname{poly}(d, n, b)$-time by linear programming. Step 3 appeals to the orbit equality problem, which by part (1) of the theorem can be done in $\operatorname{poly}(d, n, b)$-time. ◀

8 Orbit problems for compact tori

So far, we have studied orbit problems for algebraic tori, that is, groups of the form $T = (\mathbb{C}^\times)^d$. In this section we consider the groups $K = (\mathrm{S}^1)^d$, where $\mathrm{S}^1 = \{z \in \mathbb{C}^\times \mid |z| = 1\}$. Such groups are often called *compact tori.* Indeed, any commutative compact connected Lie group is of this form. Besides the fundamental algorithmic interest in this setting, it is also important in applications. For example, in physics, symmetries are often given by compact group actions, such as compact tori [32, 2]. We give further complexity-theoretic motivation below.

The compactness implies that orbits are closed and so the three problems in Problem 1.1 coincide. In this section, we show how to solve the orbit equality problem for a compact torus by reducing it to orbit equality for the corresponding algebraic torus. Subsequently, we give an alternative reduction that works not only for tori but in fact for any connected reductive group such as SL_n.

To start, we note that it is known that any (continuous) finite-dimensional representation of $K = (\mathrm{S}^1)^d$ extends to a representation of $T = (\mathbb{C}^\times)^d$ [62]. In particular, representations can be specified as before by a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$. Then we have the following result:

▶ **Proposition 8.1.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ define an n-dimensional representation of $T = (\mathbb{C}^\times)^d$ and $K = (\mathrm{S}^1)^d$. Let $v, w \in \mathbb{C}^n$. Then, $O_{K,v} = O_{K,w}$ if and only if $O_{T,v} = O_{T,w}$ and $|v_j| = |w_j|$ for all j.*

Proof. Since $K \subseteq T$, it is clear that if $O_{K,v} = O_{K,w}$, then $O_{T,v} = O_{T,w}$ and $|v_j| = |w_j|$ for all j.

Conversely, suppose $O_{T,v} = O_{T,w}$ and $|v_j| = |w_j|$ for all j. Then, there is some $t \in T$ such that $t \cdot v = w$. Write $t = (t_1, \ldots, t_d)$ and write each $t_i = r_i \cdot e^{\mathrm{i}\theta_i}$, with $r_i > 0$ and $\theta_i \in \mathbb{R}$. Then, it is easy to see that we must have $(e^{\mathrm{i}\theta_1}, \ldots, e^{\mathrm{i}\theta_d}) \cdot v = w$. Thus v and w are in the same K-orbit. ◀

Proof of Corollary 1.6. We are given $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ and $v, w \in \mathbb{Q}(i)^n$. By the above proposition, we need to check if $O_{T,v} = O_{T,w}$ and if $|v_j| = |w_j|$ for all j. The former can be done in polynomial time by Theorem 1.2 and the latter can clearly be done in polynomial time. ◀

Before proceeding we give some further context and motivation. Algorithms for the null cone membership problem (given a rational representation $\rho : G \to \mathrm{GL}(V)$ of a reductive group G and $v \in V$, decide if $0 \in \overline{O_v}$) based on optimization methods have emerged in recent years. They take advantage of the fact that $0 \in \overline{O_v}$ if and only if one can drive the norm to 0 along the orbit O_v. This can be viewed as an optimization problem where one tries to minimize (infimize) the norm along the orbit. While this is not a convex optimization problem, it is geodesically convex by the Kempf-Ness theory [45], which allows for many of the ideas to be modified appropriately. As far as the orbit closure intersection problem is concerned, the natural extension of this idea is as follows: Given $v, w \in V$, first use an optimization algorithm to approximately find a point in each orbit closure with minimal norm; let us call these points $\check{v}$, $\check{w}$. Then, appealing to the Kempf-Ness theory again, we have that $\overline{O_v} \cap \overline{O_w} \neq \emptyset$ if and only if $\check{v}$ and $\check{w}$ are in the same orbit for a maximal compact subgroup K of G. In this way, the orbit closure intersection problem for G can be reduced to the orbit equality problem for the maximal compact subgroup K. In fact, for the so-called left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on matrix-tuples, this idea was carried out successfully to obtain a polynomial-time algorithm for orbit closure intersection [1]. This further emphasizes the importance of the orbit equality problem for compact Lie group actions.

Here we report on an interesting phenomenon, which provides a kind of converse to the strategy explained above. Namely, for any action of a connected reductive group G, the orbit equality problem for the maximal compact subgroup $K \subseteq G$ is equivalent to an orbit intersection (or equality) problem for a related action of G! As this result is not crucial to the rest of the paper and requires significantly different background, we will be brief in our explanations. We denote by V^* the contragredient or dual representation of V.

▶ **Theorem 8.2.** *Let $\rho\colon G \to \mathrm{GL}(V)$ be a finite-dimensional representation of a connected reductive group G. Let K be a maximal compact subgroup of G, and $\langle\cdot,\cdot\rangle$ be a K-invariant Hermitian inner product on V. For $v \in V$, let $\widehat{v} \in V^*$ be defined by $\widehat{v}(w) := \langle v, w\rangle$. Then, for $v, w \in V$, the following are equivalent:*

1. *$O_{K,v} = O_{K,w}$;*
2. *$O_{G,(v,\widehat{v})} = O_{G,(w,\widehat{w})}$ in $V \oplus V^*$;*
3. *The G-orbit closures of $(v, \widehat{v})$ and $(w, \widehat{w})$ in $V \oplus V^*$ intersect.*

Proof. Let $\mathrm{Lie}(G) \subseteq L(V)$ denote the Lie algebra of G. For any linear action of G on a vector space U, we get an induced action of $\mathrm{Lie}(G)$ on U. Given a K-invariant Hermitian form $\langle\cdot,\cdot\rangle$ on U, we define the so-called moment map $\mu_U\colon U \to \mathrm{Lie}(G)^*$ by the formula $\mu_U(u)(X) = \langle u, X \cdot u\rangle$ for $u \in U$ and $X \in \mathrm{Lie}(G)$ (up to a scalar which is not relevant for our purposes). The celebrated Kempf-Ness theorem says that if $\mu_U(u) = 0$ then the G-orbit of u is closed. Moreover, it asserts that if $u' \in U$ is another point such that $\mu_U(u') = 0$, then $O_{G,u} = O_{G,u'}$ if and only if $O_{K,u} = O_{K,u'}$.

Applying the preceding to $(v, \widehat{v})$ and $(w, \widehat{w})$ in $U = V \oplus V^*$, a simple calculation shows that the moment map vanishes at either point, so the two orbits are closed. This shows the equivalence between (2) and (3). The equivalence between (1) and (2) follows immediately from the second part of the Kempf-Ness theorem, using that $k\widehat{v} = \widehat{kv}$ for any $k \in K$, since K acts unitarily. ◀

9 Concluding remarks, future directions, and open problems

To better understand the context of our results and their potential impact on future progress, we briefly discuss some results in literature and then suggest further research directions. In very high level, we feel that the following aspects are highlighted by this work: the relative power and interplay between algebraic and analytical algorithms, the importance of understanding commutative actions as a stepping stone towards understanding general actions, the role of rational (as opposed to polynomial) invariants, and the subtlety of “no go” results, which evidently can be surpassed.

There has been an explosion of interest over the last decade in understanding invariant theory from a complexity theoretic perspective (we survey some of this literature in the introduction). This rapidly developing field can be seen as an endeavour to classifying computational problems in invariant theory according to their difficulty, finding efficient algorithms whenever possible, as well as connecting to applications in mathematics, physics, optimization, and statistics.

Invariant theory in the setting of a rational representation of a connected reductive group is the most relevant for complexity theory. The commutative case of tori is an important special case. Despite the well-understood structural simplicity of the corresponding invariant theory, even basic algorithmic problems are non-trivial. Null cone membership, arguably the most basic problem, has long been known to have an efficient algorithm, as it reduces to linear programming, which non-trivially admits polynomial-time algorithms. The problems of orbit equality, orbit closure intersection, and orbit closure containment have polynomial time algorithms, as shown in this paper. We stress that while efficient algorithms for linear programming are "continuous" or "analytic" in nature, our algorithms use a combination of *both* analytic *and* algebraic techniques. The more general problem of succinct circuits for generating polynomial invariants, which is one of the basic challenges proposed in [52], has recently shown to be impossible under natural complexity assumptions [29]. Yet, in this paper, we bypass this negative result, and see that *rational* invariants for torus actions can be captured in a computationally efficient way without the need for succinct circuits. It is an interesting open problem to determine if there are succinct circuits for separating invariants or null cone definers, see [29, Problems 1.14, 1.15].

The invariant theory of non-commutative groups has a different flavor from, and is far more complex than, the commutative case, see, for example, [38]. Many interesting problems in computational invariant theory remain open in the non-commutative case. We list a few. First and foremost, the results in this paper motivate the investigation of the computational efficiency of systems of generating *rational* invariants. Further, it is natural to wonder if rational invariants can help capture orbit closure intersection and orbit equality for non-commutative group actions. Another open problem is to give *any* polynomial time algorithm for orbit closure intersection (and the subproblem of null cone membership). An intermediate challenge is to ascertain whether null cone membership is in NP ∩ co-NP. Note that in [5] it is shown that the general orbit closure containment problem is NP-hard.

References

1 Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *STOC'18—Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 172–181. ACM, New York, 2018. doi:10.1145/3188745.3188942.

2 Michele Audin. *Torus actions on symplectic manifolds*, volume 93. Birkhäuser, 2012.

3 Eric Bach, James R. Driscoll, and Jeffrey O. Shallit. Factor refinement. In David S. Johnson, editor, *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California, USA*, pages 201–211. SIAM, 1990. URL: http://dl.acm.org/citation.cfm?id=320176.320199.

4 Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical review letters*, 70(13):1895, 1993.

5 Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. Variety membership testing, algebraic natural proofs, and geometric complexity theory. arXiv, 2020. arXiv:1911.02534.

6 Peter A. Brooksbank and Eugene M. Luks. Testing isomorphism of modules. *Journal of Algebra*, 320(11):4020–4029, 2008. Computational Algebra. doi:10.1016/j.jalgebra.2008.07.014.

7 Peter Bürgisser, Matthias Christandl, Ketan D. Mulmuley, and Michael Walter. Membership in moment polytopes is in NP and coNP. *SIAM J. Comput.*, 46(3):972–991, 2017. doi:10.1137/15M1048859.

8 Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Towards a theory of non-commutative optimization: geodesic first and second order methods for moment maps and polytopes. In *60th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2019*, pages 845–861. IEEE Computer Soc., Los Alamitos, CA, 2019. arXiv:1910.12375.

9 Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In *59th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2018*, pages 883–897. IEEE Computer Soc., Los Alamitos, CA, 2018. doi:10.1109/FOCS.2018.00088.

10 Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. In *9th Innovations in Theoretical Computer Science*, volume 94 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 24, 20. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.

11 Peter Bürgisser, Yinan Li, Harold Nieuwboer, and Michael Walter. Interior-point methods for unconstrained geometric programming and scaling problems. arXiv, 2020. `arXiv:2008.12110`.

12 Michael B Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained Newton's method and interior point methods. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS 2017)*, pages 902–913. IEEE, 2017. `arXiv:1704.02310`.

13 Stephen A. Cook. The complexity of theorem proving procedures. In *Proc. 3rd ACM STOC*, pages 151–158, 1971.

14 David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997.

15 Harm Derksen. Polynomial bounds for rings of invariants. *Proc. Amer. Math. Soc.*, 129(4):955–963, 2001. `doi:10.1090/S0002-9939-00-05698-7`.

16 Harm Derksen. The graph isomorphism problem and approximate categories. *J. Symb. Comput.*, 59:81–112, 2013. `doi:10.1016/j.jsc.2013.06.002`.

17 Harm Derksen and Gregor Kemper. *Computational invariant theory*, volume 130 of *Encyclopaedia of Mathematical Sciences*. Springer, Heidelberg, enlarged edition, 2015. With two appendices by Vladimir L. Popov, and an addendum by Norbert A'Campo and Popov, Invariant Theory and Algebraic Transformation Groups, VIII. `doi:10.1007/978-3-662-48422-7`.

18 Harm Derksen and Visu Makam. Generating invariant rings of quivers in arbitrary characteristic. *J. Algebra*, 489:435–445, 2017. `doi:10.1016/j.jalgebra.2017.06.035`.

19 Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. *Adv. Math.*, 310:44–63, 2017. `doi:10.1016/j.aim.2017.01.018`.

20 Harm Derksen and Visu Makam. Degree bounds for semi-invariant rings of quivers. *J. Pure Appl. Algebra*, 222(10):3282–3292, 2018. `doi:10.1016/j.jpaa.2017.12.007`.

21 Harm Derksen and Visu Makam. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *Algebra Number Theory*, 14(10):2791–2813, 2020. `doi:10.2140/ant.2020.14.2791`.

22 Harm Derksen and Visu Makam. An exponential lower bound for the degrees of invariants of cubic forms and tensor actions. *Adv. Math.*, 368:107136, 25, 2020. `doi:10.1016/j.aim.2020.107136`.

23 Igor Dolgachev. *Lectures on invariant theory*, volume 296 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2003. `doi:10.1017/CBO9780511615436`.

24 Arnaud Durand, Miki Hermann, and Laurent Juban. On the complexity of recognizing the Hilbert basis of a linear Diophantine system. *Theoret. Comput. Sci.*, 270(1-2):625–642, 2002. `doi:10.1016/S0304-3975(01)00017-2`.

25 Kousha Etessami, Alistair Stewart, and Mihalis Yannakakis. A note on the complexity of comparing succinctly represented integers, with an application to maximum probability parsing. *ACM Trans. Comput. Theory*, 6(2):9:1–9:23, 2014. `doi:10.1145/2601327`.

26 Michael A. Forbes and Amir Shpilka. Explicit Noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, randomization, and combinatorial optimization*, volume 8096 of *Lecture Notes in Comput. Sci.*, pages 527–542. Springer, Heidelberg, 2013. `doi:10.1007/978-3-642-40328-6_37`.

27 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *57th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2016*, pages 109–117. IEEE Computer Soc., Los Alamitos, CA, 2016. `doi:10.1109/FOCS.2016.95`.

28 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Operator scaling: theory and applications. *Found. Comput. Math.*, 20(2):223–290, 2020. doi:10.1007/s10208-019-09417-z.

29 Ankit Garg, Christian Ikenmeyer, Visu Makam, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Search problems in algebraic complexity, GCT, and hardness of generators for invariant rings. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.12.

30 Guoqiang Ge. Testing equalities of multiplicative representations in polynomial time (extended abstract). In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 422–426. IEEE Computer Society, 1993. doi:10.1109/SFCS.1993.366845.

31 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993. doi:10.1007/978-3-642-78240-4.

32 Victor Guillemin and Shlomo Sternberg. *Symplectic techniques in physics*. Cambridge university press, 1990.

33 Leonid Gurvits. Classical complexity and quantum entanglement. *J. Comput. Syst. Sci.*, 69(3):448–484, 2004. doi:10.1016/j.jcss.2004.06.003.

34 Leonid Gurvits. Combinatorial and algorithmic aspects of hyperbolic polynomials. *arXiv preprint*, 2004. arXiv:math/0404474.

35 D. Hilbert. Über die vollen Invariantensysteme. *Math. Ann.*, 42:313–373, 1893. URL: http://eudml.org/doc/157652.

36 David Hilbert. Über die Theorie der algebraischen Formen. *Math. Ann.*, 36(4):473–534, 1890. doi:10.1007/BF01208503.

37 Evelyne Hubert and George Labahn. Scaling invariants and symmetry reduction of dynamical systems. *Foundations of Computational Mathematics*, 13, 2013. doi:10.1007/s10208-013-9165-9.

38 James E. Humphreys. *Linear algebraic groups*. Springer-Verlag, New York-Heidelberg, 1975. Graduate Texts in Mathematics, No. 21.

39 Christian Ikenmeyer, Ketan D Mulmuley, and Michael Walter. On vanishing of Kronecker coefficients. *Computational Complexity*, 26(4):949–992, 2017.

40 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative Edmonds' problem and matrix semi-invariants. *Comput. Complexity*, 26(3):717–763, 2017. doi:10.1007/s00037-016-0143-x.

41 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Comput. Complexity*, 27(4):561–593, 2018. doi:10.1007/s00037-018-0165-7.

42 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. doi:10.1007/s00037-004-0182-6.

43 Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979. doi:10.1137/0208040.

44 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103, 1972.

45 George Kempf and Linda Ness. The length of vectors in representation spaces. In *Algebraic geometry (Proc. Summer Meeting, Univ. Copenhagen, Copenhagen, 1978)*, volume 732 of *Lecture Notes in Math.*, pages 233–243. Springer, Berlin, 1979.

46 Hanspeter Kraft. *Geometrische Methoden in der Invariantentheorie*. Aspects of Mathematics, D1. Friedr. Vieweg & Sohn, Braunschweig, 1984. `doi:10.1007/978-3-322-83813-1`.

47 David B Leep and Gerry Myerson. Marriage, magic, and solitaire. *The American Mathematical Monthly*, 106(5):419–429, 1999.

48 L. A. Levin. Universal enumeration problems. *Problemy Peredači Informacii*, 9(3):115–116, 1973.

49 Nathan Linial and Zur Luria. On the vertices of the d-dimensional Birkhoff polytope. *Discrete & Computational Geometry*, 51(1):161–170, 2014.

50 Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica*, 20(4):545–568, 2000.

51 Visu Makam and Avi Wigderson. Singular tuples of matrices is not a null cone (and, the symmetries of algebraic varieties). *CoRR*, abs/1909.00857, 2019. `arXiv:1909.00857`.

52 Ketan D. Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *J. Amer. Math. Soc.*, 30(1):225–309, 2017. `doi:10.1090/jams/864`.

53 Ketan D Mulmuley and Milind Sohoni. Geometric complexity theory I: An approach to the P vs. NP and related problems. *SIAM Journal on Computing*, 31(2):496–526, 2001.

54 David Mumford. *The red book of varieties and schemes*, volume 1358 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1988. `doi:10.1007/978-3-662-21581-4`.

55 David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory, Third Edition*, volume 34 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, 1994.

56 Vladimir L Popov. Two orbits: When is one in the closure of the other? *Proceedings of the Steklov Institute of Mathematics*, 264(1):146–158, 2009.

57 Alexander Schrijver. *Theory of linear and integer programming*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Ltd., Chichester, 1986. A Wiley-Interscience Publication.

58 R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35:876–879, 1964.

59 Henry J. Stephen Smith. On systems of linear indeterminate equations and congruences. *Philosophical Transactions of the Royal Society of London*, 151:293–326, 1861. URL: `http://www.jstor.org/stable/108738`.

60 Bernd Sturmfels. *Algorithms in Invariant Theory*. Texts & Monographs in Symbolic Computation. Springer, 2008. `doi:10.1007/978-3-211-77417-5`.

61 David Wehlau. Constructive invariant theory for tori. *Annales de l'institut Fourier*, 43(4):1055–1066, 1993. URL: `http://eudml.org/doc/75025`.

62 Hermann Weyl. *The Classical Groups. Their Invariants and Representations*. Princeton University Press, Princeton, N.J., 1939.

Pseudodistributions That Beat All Pseudorandom Generators (Extended Abstract)

Edward Pyne
Harvard University, Cambridge, MA, USA

Salil Vadhan
Harvard University, Cambridge, MA, USA

Abstract

A recent paper of Braverman, Cohen, and Garg (STOC 2018) introduced the concept of a *weighted pseudorandom generator (WPRG)*, which amounts to a pseudorandom generator (PRG) whose outputs are accompanied with real coefficients that scale the acceptance probabilities of any potential distinguisher. They gave an explicit construction of WPRGs for ordered branching programs whose seed length has a better dependence on the error parameter ε than the classic PRG construction of Nisan (STOC 1990 and Combinatorica 1992).

In this work, we give an explicit construction of WPRGs that achieve parameters that are *impossible* to achieve by a PRG. In particular, we construct a WPRG for *ordered permutation branching programs of unbounded width* with a single accept state that has seed length $\tilde{O}(\log^{3/2} n)$ for error parameter $\varepsilon = 1/\operatorname{poly}(n)$, where n is the input length. In contrast, recent work of Hoza et al. (ITCS 2021) shows that any PRG for this model requires seed length $\Omega(\log^2 n)$ to achieve error $\varepsilon = 1/\operatorname{poly}(n)$.

As a corollary, we obtain explicit WPRGs with seed length $\tilde{O}(\log^{3/2} n)$ and error $\varepsilon = 1/\operatorname{poly}(n)$ for ordered permutation branching programs of width $w = \operatorname{poly}(n)$ with an arbitrary number of accept states. Previously, seed length $o(\log^2 n)$ was only known when both the width and the reciprocal of the error are subpolynomial, i.e. $w = n^{o(1)}$ and $\varepsilon = 1/n^{o(1)}$ (Braverman, Rao, Raz, Yehudayoff, FOCS 2010 and SICOMP 2014).

The starting point for our results are the recent space-efficient algorithms for estimating random-walk probabilities in directed graphs by Ahmadenijad, Kelner, Murtagh, Peebles, Sidford, and Vadhan (FOCS 2020), which are based on spectral graph theory and space-efficient Laplacian solvers. We interpret these algorithms as giving WPRGs with large seed length, which we then derandomize to obtain our results. We also note that this approach gives a simpler proof of the original result of Braverman, Cohen, and Garg, as independently discovered by Cohen, Doron, Renard, Sberlo, and Ta-Shma (these proceedings).

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization

Keywords and phrases pseudorandomness, space-bounded computation, spectral graph theory

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.33

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2021/019/` [28]

Funding *Edward Pyne*: Supported by NSF grant CCF-1763299.
Salil Vadhan: Supported by NSF grant CCF-1763299 and a Simons Investigator Award.

Acknowledgements We thank Jack Murtagh and Sumegha Garg for insightful discussions, and Oded Goldreich and the CCC '21 reviewers for feedback that improved our presentation.

1 Introduction

The notion of a **pseudorandom generator (PRG)** [7, 35, 26] is ubiquitous in theoretical computer science, with vast applicability in cryptography and derandomization. (See the texts [17, 34] for more background on pseudorandomness.) A recent work of Braverman, Cohen, and Garg [9] introduced the following intriguing generalization of a PRG, in which we attach real coefficients to the outputs of the generator:

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 33; pp. 33:1–33:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Definition 1.** *Let $\mathcal{B}$ be a class of boolean functions $B\colon \{0,1\}^n \to \{0,1\}$. An **$\varepsilon$-weighted pseudorandom generator (WPRG)** for $\mathcal{B}$ is a function $(G,\rho)\colon \{0,1\}^s \to \{0,1\}^n \times \mathbb{R}$ such that for every $B \in \mathcal{B}$,*

$$\left| \mathop{\mathbb{E}}_{x \leftarrow U_{\{0,1\}^n}}[B(x)] - \mathop{\mathbb{E}}_{x \leftarrow U_{\{0,1\}^s}}[\rho(x) \cdot B(G(x))] \right| \le \varepsilon.$$

*The value s is the **seed length** of the WPRG, and n is the **output length** of the WPRG. We say that the WPRG is **(mildly)**[1] **explicit** if given x, $G(x)$ and $\rho(x)$ are computable in space $O(s)$, and $\rho(x)$ has absolute value at most $2^{O(s)}$.*

Above and throughout, we use the standard definition of space-bounded complexity, which counts the working, read-write memory of the algorithm, and does not include the length of the read-only input or write-only output, which can be exponentially longer than the space bound.

In the original work of Braverman, Cohen, and Garg [9] and previous versions of this paper [28], generators as above were called **pseudorandom pseudodistributions (PRPDs)**. The terminology of weighted pseudorandom generators (WPRGs) was introduced by Cohen et al. [14], and we find it more intuitive (and it avoids the double use of the "pseudo-" prefix).

With Definition 1, a PRG is a special case of a WPRG with $\rho(x) = 1$. The power of WPRGs comes from allowing the coefficients to be negative, which yields cancellations. Indeed, an explicit ε-WPRG with seed length s in which all of the coefficients are nonnegative can be converted into an explicit $O(\varepsilon)$-PRG with seed length $O(s + \log(1/\varepsilon))$. A general WPRG can be converted into a linear combination of two unweighted generators. That is, for every explicit WPRG $(G,\rho)\colon \{0,1\}^s \to \{0,1\}^n \times \mathbb{R}$, there are explicit generators $G_+ : \{0,1\}^{s'} \to \{0,1\}^n$ and $G_- : \{0,1\}^{s'} \to \{0,1\}^n$ with seed length $s' = O(s + \log(1/\varepsilon))$ and coefficients $\rho_+, \rho_- \in \mathbb{R}$ such that for every function $B : \{0,1\}^n \to \{0,1\}$, we have:

$$\mathop{\mathbb{E}}_{x}[\rho(x) \cdot B(G(x))] = \rho_+ \cdot \mathop{\mathbb{E}}_{x}[G_+(x)] - \rho_- \cdot \mathop{\mathbb{E}}_{x}[G_-(x)] \pm \varepsilon.$$

The motivation for WPRGs is that they can be used to derandomize algorithms in the same way as a PRG: we can estimate the acceptance probability of any function $B \in \mathcal{B}$ by enumerating over the seeds x of the WPRG (G,ρ) and calculating the average of the values $\rho(x) \cdot B(G(x))$. Furthermore, [9] observe that if (G,ρ) is an ε-WPRG for a model then G is an **ε-hitting set generator (HSG)**. That is, if B is any function in $\mathcal{B}$ with $\Pr[B(U_n) = 1] > \varepsilon$, then there exists an $x \in \{0,1\}^s$ such that $B(G(x)) = 1$.

Given this motivation, it is natural to ask whether WPRGs are more powerful than PRGs. That is, can ε-WPRGs achieve a shorter seed length than ε-PRGs for a natural computational model $\mathcal{B}$? (There are simple constructions of artificial examples.) As discussed below, Braverman, Cohen, and Garg [9] gave an explicit construction of WPRGs achieving a shorter seed length than the *best known* construction of PRGs for ordered branching programs, but not beating the best possible seed length for that model (given by a non-explicit application of the Probabilistic Method). In this work, we give an explicit construction of WPRGs for a natural computational model (ordered permutation branching programs of unbounded width) with a seed length that beats all possible PRGs for that model.

[1] We consider this definition to correspond to *mild* explicitness because requiring that the generator be computable in space linear in its seed length only implies that it is computable in time exponential in its seed length (i.e. time polynomial in the size of its truth table), which is mildly explicit according to the terminology in [34]. *Strong* explicitness, in contrast, would require that each bit of the truth table is computable in time polynomial in s.

1.1 Ordered Branching Programs

The work of Braverman, Cohen, and Garg [9], as well as our paper, focuses on WPRGs for classes $\mathcal{B}$ of functions computable by *ordered branching programs*, a nonuniform model that captures how a space-bounded randomized algorithm accesses its random bits.

▶ **Definition 2.** *An **(ordered) branching program** B of length n and width w computes a function $B : \{0,1\}^n \to \{0,1\}$. On an input $\sigma \in \{0,1\}^n$, the branching program computes as follows. It starts at a fixed start state $v_0 \in [w]$. Then for $r = 1, \ldots, n$, it reads the next symbol σ_r and updates its state according to a transition function $B_r : [w] \times \{0,1\} \to [w]$ by taking $v_t = B_r(v_{t-1}, \sigma_t)$. Note that the transition function B_r can differ at each time step.*

*The branching program **accepts** σ, denoted $B(\sigma) = 1$, if $v_n \in V_{acc}$, where $V_{acc} \subseteq [w]$ is the set of accept states, and otherwise it **rejects**, denoted $B(\sigma) = 0$. Thus an ordered branching program is specified by the transition functions $B_1, \ldots, B_n$, the start state v_0 and the set V_{acc} of accept states.*

An ordered branching program of length n and width w can compute the output of an algorithm that uses $\log w$ bits of memory and n random bits, by taking the state at each layer as the contents of memory at that time. We note that we can convert any ordered branching program into one with a single accept state by collapsing all of V_{acc} to a single state.

Using the probabilistic method, it can be shown that there *exists* an ε-PRG for ordered branching programs of length n and width w with seed length $s = O(\log(nw/\varepsilon))$. The classic construction of Nisan [25] gives an explicit PRG with seed length $s = O(\log n \cdot \log(nw/\varepsilon))$, and this bound has not been improved except for extreme ranges of w, namely when w is at least quasipolynomially larger than (n/ε) [27, 5, 22] or when $w \leq 3$ [8, 32, 19, 24]. Braverman, Cohen, and Garg [9] gave an explicit construction of a WPRG that achieves improved dependence on the error parameter ε, with seed length

$$s = \tilde{O}\left(\log n \cdot \log(nw) + \log(1/\varepsilon)\right).$$

In particular, for error $\varepsilon = n^{-\log n}$ and width $w = \text{poly}(n)$, their seed length improves Nisan's from $O(\log^3 n)$ to $\tilde{O}(\log^2 n)$. Chatthopadhyay and Liao [12] gave a simpler construction of WPRGs with a slightly shorter seed length than [9], with an additive dependence on $O(\log(1/\varepsilon))$ rather than $\tilde{O}(\log(1/\varepsilon))$.

1.2 Permutation Branching Programs

Due to the lack of progress in constructing improved PRGs for general ordered branching programs as well as some applications, attention has turned to more restricted classes of ordered branching programs. In this work, our focus is on *permutation* branching programs:

▶ **Definition 3.** *An **(ordered) permutation branching program** is an ordered branching program B where for all $t \in [n]$ and $\sigma \in \{0,1\}$, $B_t(\cdot, \sigma)$ is a permutation on $[w]$.*

This can be thought of as the computation being time-reversible on any fixed input σ. We note that we cannot assume without loss of generality that a permutation branching program has a single accept state, as merging a set of accept states will destroy the permutation property. Nevertheless, ordered permutation branching programs with a single accept state can compute interesting functions, such as testing whether a $\sum_{i \in S} x_i \equiv 0 \pmod{m}$, for any $m \leq w$ and any $S \subseteq [n]$. An ordered permutation branching program with a single accept state can also test whether $x|_T = \pi(x|_S)$ for any permutation $\pi : \{0,1\}^\ell \to \{0,1\}^\ell$ and any two subsets $S, T \subseteq [n]$ of size ℓ such that all elements of T are larger than all elements of S, provided that $w \geq 2^\ell$ [20].

Previous works on various types of PRGs for permutation branching programs [30, 29, 10, 23, 15, 33, 20] have achieved seed lengths that are logarithmic or nearly logarithmic in the length n of the branching program, improving the $\log^2 n$ bound in Nisan's generator. In particular, Braverman, Rao, Raz, and Yehudayoff [10] gave a PRG for the more general model of *regular* branching programs (with an arbitrary number of accept states) with seed length

$$s = O\left(\log n \cdot (\log w + \log(1/\varepsilon) + \log\log n)\right).$$

For getting a HSG, they also showed how how to eliminate the $\log\log n$ and $\log(1/\varepsilon)$ terms at the price of a worse dependence on w,[2] achieving a seed length of

$$s \leq \log(n+1) \cdot w.$$

For the specific case of permutation branching programs, Koucký, Nimbhorkar, and Pudlák [23], De [15], and Steinke [33] showed how to remove the $\log\log n$ term in the Braverman et al. PRG at the price of a worse dependence on w, achieving seed length

$$s = O(\log n \cdot (\mathrm{poly}(w) + \log(1/\varepsilon))).$$

Most recently, Hoza, Pyne, and Vadhan [20] showed that the dependence on the width w could be entirely eliminated if we restrict to permutation branching programs with a *single accept state*, constructing a PRG with seed length

$$s = O(\log n \cdot (\log\log n + \log(1/\varepsilon)).$$

In particular, they show that this seed length is provably better than what is achieved by the Probabilistic Method; that is, a random function with seed length $o(n)$ fails to be a PRG for unbounded-width permutation branching programs with high probability. Like the prior PRGs for bounded-width permutation branching programs, the seed length has a term of $O(\log n \cdot \log(1/\varepsilon))$. However, in contrast to the bounded-width case, this cannot be improved to $O(\log(n/\varepsilon))$ by a non-explicit construction. Hoza et al. prove that seed length $\Omega(\log n \cdot \log(1/\varepsilon))$ is *necessary* for any ε-PRG against unbounded-width permutation branching programs. For hitting-set generators (HSGs), they show that seed length $O(\log(n/\varepsilon))$ is possible via the Probabilistic Method, thus leaving an explicit construction as an open problem.

1.3 Our Results

In this paper, we construct an explicit WPRG for permutation branching programs of unbounded width and a single accept state that beats the aforementioned lower bounds for PRGs:

▶ **Theorem 4.** *For all $n \in N$ and $\varepsilon \in (0, 1/2)$, there is an explicit ε-WPRG (and hence ε-HSG) for ordered permutation branching programs of length n, arbitrary width, and a single accept state, with seed length*

$$s = O\left(\log(n)\sqrt{\log(n/\varepsilon)}\sqrt{\log\log(n/\varepsilon)} + \log(1/\varepsilon)\log\log(n/\varepsilon)\right).$$

In particular, when $\varepsilon = 1/\mathrm{poly}(n)$, we achieve seed length $\tilde{O}(\log^{3/2} n)$, while a PRG requires seed length $\Omega(\log^2 n)$ [20].

[2] The lack of dependence on ε can be explained by the observation of Braverman et al. that any regular branching program that has nonzero acceptance probability has acceptance probability at least $1/2^{w-1}$, so WLOG $\varepsilon > 1/2^w$, i.e. $w > \log(1/\varepsilon)$.

As noted in [20], an ε-WPRG for branching programs with a single accept state is also an $(a \cdot \varepsilon)$-WPRG for branching programs with at most a accept states. For bounded-width permutation branching programs, we can take $a = w$ and obtain:

▶ **Corollary 5.** *For all $n, w \in \mathbb{N}$ and $\varepsilon \in (0, 1/2)$, there is an explicit ε-WPRG (and hence ε-HSG) for ordered permutation branching programs of length n and width w (and any number of accept states), with seed length*

$$s = O\left(\log(n)\sqrt{\log(nw/\varepsilon)}\sqrt{\log\log(nw/\varepsilon)} + \log(w/\varepsilon)\log\log(nw/\varepsilon)\right).$$

In particular for $w = \operatorname{poly}(n)$ and $\varepsilon = 1/\operatorname{poly}(n)$, we achieve seed length $\tilde{O}(\log^{3/2} n)$. Note that the previous explicit PRGs (or even HSGs) for permutation branching programs (as mentioned in Subsection 1.2) achieved seed length $o(\log^2 n)$ only when both $w = n^{o(1)}$ and $\varepsilon = 1/n^{o(1)}$. With seed length $o(\log^2 n)$, Corollary 5 can handle width as large as $w = n^{\tilde{\Omega}(\log n)}$ and error as small as $\varepsilon = 1/n^{-\tilde{\Omega}(\log(n))}$. We summarize these results in a table.

Citation	Type	Model	Seed Length
Non-explicit (folklore)	PRG	General	$\Theta(\log(nw/\varepsilon)$
[25, 21]	PRG	General	$O(\log n \cdot \log(nw/\varepsilon))$
[10]	PRG	Regular	$\tilde{O}(\log n \cdot \log(w/\varepsilon))$
[10]	HSG	Regular	$\log(n+1) \cdot w$
[23, 15, 33]	PRG	Permutation	$O(\log n \cdot (\operatorname{poly}(w) + \log(1/\varepsilon))$
[9, 12, 28]	WPRG	General	$\tilde{O}(\log n \cdot \log nw + \log(1/\varepsilon))$
[20]	PRG	Permutation (1 accept)	$\tilde{\Theta}(\log n \cdot \log(1/\varepsilon))$
Non-explicit [20]	HSG	Permutation (1 accept)	$O(\log(n/\varepsilon))$
Theorem 4	WPRG	Permutation (1 accept)	$\tilde{O}(\log n\sqrt{\log(n/\varepsilon)} + \log(1/\varepsilon))$
Corollary 5	WPRG	Permutation	$\tilde{O}(\log n\sqrt{\log(nw/\varepsilon)} + \log(w/\varepsilon))$

2 Overview of Proofs

The starting point for our results are the recent space-efficient algorithms for estimating random-walk probabilities in directed graphs by Ahmadenijad, Kelner, Murtagh, Peebles, Sidford, and Vadhan [2], which are based on spectral graph theory and space-efficient Laplacian solvers. We interpret these algorithms as giving WPRGs with large seed length, which we then derandomize to obtain our results.

The specific problem considered by Ahmadenijad et al. is the following: given a directed graph $\mathcal{G} = (V, E)$, two vertices $s, t \in V$, a walk-length $k \in \mathbb{N}$, and an error parameter $\varepsilon > 0$, estimate the probability that a random walk of length k started at s ends at t to within $\pm\varepsilon$. Such an algorithm can be applied to the following graph in order to estimate the acceptance probability of an ordered branching program:

▶ **Definition 6.** *Given a length n, width w branching program B with transition functions $(B_1, \ldots, B_n)$ with start vertex $v_0 \in [w]$, and a single accept vertex v_{acc}, the* ***(layered) graph associated with B*** *is the graph $\mathcal{G}$ with vertex set $\{0, 1, \ldots, n\} \times [w]$ and directed edges from $(i-1, v)$ to $(i, B_i(v, 0))$ and $(i, B_i(v, 1))$ for every $i = 1, \ldots, n$ and $v \in [w]$.*

Applying the algorithms of Ahmadenijad et al. to the graph $\mathcal{G}$ with $s = (0, v_0)$, $t = (n, v_{\mathrm{acc}})$, and $k = n$, we obtain an estimate of the acceptance probability of B to within $\pm\varepsilon$, just like an ε-WPRG for B would allow us to obtain. But a WPRG (G, ρ) is much more constrained than an arbitrary space-efficient algorithm, which can directly inspect the graph. Instead, a WPRG is limited to generating $S = 2^s$ walks of length n in the layered graph, described by sequences $G(x_1), \ldots, G(x_S) \in \{0,1\}^n$ of edge labels, and then combining the indicators $B(G(x_1)), \ldots, B(G(x_n))$ of whether the walks ended at t via a linear combination with fixed coefficients $\rho(x_1), \ldots, \rho(x_S) \in \mathbb{R}$.

Note that if B is a permutation branching program, then the graph $\mathcal{G}$ above is 2-regular (except for layer 0 which has no incoming edges and layer n which has no outgoing edges). Thus, the basis for Theorem 4 is the (main) result of Ahmadenijad et al., which applies to regular (or more generally, Eulerian) directed graphs G. However, they also give a new algorithm for estimating random-walk probabilities in arbitrary directed graphs. This algorithm is not as space-efficient as the ones for regular graphs, but is significantly simpler, so we begin by describing how to obtain a WPRG based on that algorithm. The resulting WPRG matches the parameters of the WPRG of Braverman, Cohen, and Garg [9], but has a significantly simpler proof (and is also simpler than the construction of Chatthopadhyay and Liao [12]). A similar construction was independently discovered by Cohen, Doron, Renard, Sberlo, and Ta-Shma [14].

2.1 WPRG for Arbitrary Ordered Branching Programs

Let B be an arbitrary width w, length n ordered branching program, with associated layered graph $\mathcal{G}$ as in Definition 6. The algorithm of Ahmadenijad et al. starts with the $(n+1)w \times (n+1)w$ random-walk transition matrix $\mathbf{W}$ of $\mathcal{G}$, which has the following block structure:

$$\mathbf{W} = \begin{bmatrix} 0 & \mathbf{B}_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{B}_2 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & \mathbf{B}_n \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Here entry $((i,u),(j,v))$ is the probability that taking one random step in $\mathcal{G}$ from vertex (i,u) ends at (j,v). Thus $\mathbf{B}_i$ is the $w \times w$ transition matrix for the random walk from layer $i-1$ to i in the branching program. (Note that the matrix $\mathbf{W}$ is not quite stochastic due to layer n having no outgoing edges.)

Ahmadenijad et al. consider the Laplacian $\mathbf{L} = \mathbf{I}_{(n+1)w} - \mathbf{W}$. Its inverse $\mathbf{L}^{-1} = (\mathbf{I}_{(n+1)w} - \mathbf{W})^{-1} = \mathbf{I}_{(n+1)w} + \mathbf{W} + \mathbf{W}^2 + \mathbf{W}^3 + \cdots$ sums up random-walks of all lengths in G, and thus has the following form:

$$\mathbf{L}^{-1} = \begin{bmatrix} \mathbf{B}_{0\ldots 0} & \mathbf{B}_{0\ldots 1} & \mathbf{B}_{0\ldots 2} & \cdots & \mathbf{B}_{0\ldots n} \\ 0 & \mathbf{B}_{1\ldots 1} & \mathbf{B}_{1\ldots 2} & \cdots & \mathbf{B}_{1\ldots n} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & \mathbf{B}_{n-1\ldots n} \\ 0 & 0 & 0 & \cdots & \mathbf{B}_{n\ldots n} \end{bmatrix},$$

where

$$\mathbf{B}_{i\ldots j} = \mathbf{B}_{i+1}\mathbf{B}_{i+2}\cdots\mathbf{B}_j.$$

In particular, the $(0,n)$'th block of $\mathbf{L}^{-1}$ gives the random-walk probabilities from layer 0 to layer n, and thus the acceptance probability of G is exactly the (v_0, v_{acc})'th entry of the $(0,n)$'th block of $\mathbf{L}^{-1}$. Therefore, the task reduces to producing a sufficiently good estimate of $\mathbf{L}^{-1}$.

Ahmadenijad et al. estimate $\mathbf{L}^{-1}$ in two steps. First, they observe that the Saks–Zhou derandomization of logspace [31] can be used to produce, in deterministic space $O(\log(nw)\sqrt{\log(n)})$, approximations $\widetilde{\mathbf{B}_{i\ldots j}}$ of the blocks $\mathbf{B}_{i\ldots j}$ to within entrywise error $1/\operatorname{poly}(nw)$, resulting in an approximate pseudoinverse

$$\widetilde{\mathbf{L}^{-1}} = \begin{bmatrix} \widetilde{\mathbf{B}_{0\ldots 0}} & \widetilde{\mathbf{B}_{0\ldots 1}} & \widetilde{\mathbf{B}_{0\ldots 2}} & \cdots & \widetilde{\mathbf{B}_{0\ldots n}} \\ 0 & \widetilde{\mathbf{B}_{1\ldots 1}} & \widetilde{\mathbf{B}_{1\ldots 2}} & \cdots & \widetilde{\mathbf{B}_{1\ldots n}} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & \widetilde{\mathbf{B}_{n-1\ldots n}} \\ 0 & 0 & 0 & \cdots & \widetilde{\mathbf{B}_{n\ldots n}} \end{bmatrix}, \tag{1}$$

with the property that

$$\left\|\mathbf{I}_{(n+1)w} - \widetilde{\mathbf{L}^{-1}}\mathbf{L}\right\|_1 \leq 1/nw,$$

where $\|\cdot\|_1$ denotes the ℓ_1 operator norm on row vectors, ie $\|\mathbf{M}\|_1 = \sup_{x\neq 0}\|x\mathbf{M}\|_1/\|x\|_1$.

Next, Ahmadenijad et al. reduce the approximation error to an arbitrary $\varepsilon < 1/(nw)^{O(1)}$ by using preconditioned Richardson iterations, as captured by the following lemma:

▶ **Lemma 7** (preconditioned Richardson iteration, [2] Lemma 6.2). *Let $\|\cdot\|$ be a submultiplicative norm on $N \times N$ real matrices. Given matrices $\mathbf{A}, \mathbf{P}_0 \in \mathbb{R}^{N\times N}$ such that $\|\mathbf{I}_N - \mathbf{P}_0\mathbf{A}\| \leq \alpha$ for some constant $\alpha > 0$, let $\mathbf{P}_m = \sum_{i=0}^{m}(\mathbf{I}_N - \mathbf{P}_0\mathbf{A})^i\mathbf{P}_0$. Then $\|\mathbf{I}_N - \mathbf{P}_m\mathbf{A}\| \leq \alpha^{m+1}$.*

Setting $N = (n+1)w$, $\mathbf{A} = \mathbf{L}$, $\mathbf{P}_0 = \widetilde{\mathbf{L}^{-1}}$, and $\alpha = 1/nw$, and $m = O(\log_{nw}(1/\varepsilon))$, we obtain $\widetilde{\mathbf{L}_\varepsilon} = \mathbf{P}_m$ such that $\|\mathbf{I}_N - \widetilde{\mathbf{L}_\varepsilon}\mathbf{L}\|_1 \leq \varepsilon/(nw)^{O(1)}$, which implies that $\widetilde{\mathbf{L}_\varepsilon}$ and $\mathbf{L}^{-1}$ are entrywise equal up to $\pm\varepsilon$, for

$$\widetilde{\mathbf{L}_\varepsilon} = \sum_{i=0}^{m}(\mathbf{I}_N - \widetilde{\mathbf{L}^{-1}}\mathbf{L})^i\widetilde{\mathbf{L}^{-1}} \tag{2}$$

In particular, the (v_0, v_{acc})'th entry of the $(0, n)$'th block of $\widetilde{\mathbf{L}_\varepsilon}$ is an estimate of the acceptance probability of the branching program to within $\pm\varepsilon$. Computing $\widetilde{\mathbf{L}_\varepsilon}$ from $\mathbf{L}$ and $\widetilde{\mathbf{L}^{-1}}$ can be done in space $O((\log nw)\cdot\log m)$, yielding Ahmadenijad et al.'s space bound of

$$O(\log(nw)\sqrt{\log(n)} + (\log nw)\cdot\log\log_{nw}(1/\varepsilon)).$$

Now we show how, with appropriate an modification, we can interpret this algorithm of Ahmadenijad et al. as a WPRG (albeit with large seed length). We replace the use of the Saks–Zhou algorithm (which requires looking at the branching program) with Nisan's pseudorandom generator. Specifically, we take $\widetilde{\mathbf{B}_{i\ldots j}}$ to be the matrix whose (u, v)'th entry is the probability that, if we start at state u in the the i'th layer and use a random output of Nisan's pseudorandom generator to take $j - i$ steps in the branching program, we end at state v in the j'th layer. For $\widetilde{\mathbf{B}_{i\ldots j}}$ to approximate $\mathbf{B}_{i\ldots j}$ to within error $\pm 1/\operatorname{poly}(nw)$ as above, Nisan's pseudorandom generator requires seed length

$$s_{\text{Nisan}} = O(\log(j-i)\cdot\log nw) = O(\log n\cdot\log nw).$$

Observe that for every i, $\widetilde{\mathbf{B}_{i\ldots i}} = \mathbf{I}_w = \mathbf{B}_{i\ldots i}$. Without loss of generality, we may also assume that $\widetilde{\mathbf{B}_{(i-1)\ldots i}} = \mathbf{B}_{(i-1)\ldots i}$, since taking one step only requires one random bit.

Next, we observe from Equation 2 that the matrix $\widetilde{\mathbf{L}_\varepsilon}$ is a polynomial of degree $2m+1$ in the matrices $\mathbf{L}$ and $\widetilde{\mathbf{L}^{-1}}$. In particular the $(0,n)$'th block of $\widetilde{\mathbf{L}_\varepsilon}$ is a polynomial of degree at most $2m+1$ in the matrices $\widetilde{\mathbf{B}_{i\dots j}}$. Specifically, using the upper-triangular structure of the matrices $\mathbf{L}$ and $\widetilde{\mathbf{L}^{-1}}$ and noting that the product of d $(n+1)\times(n+1)$ block matrices expands into a sum of $(n+1)^{d-1}$ terms, each of which is a product of d individual blocks, we show:

▶ **Observation 8.** *The $(0,n)$'th block of $\widetilde{\mathbf{L}_\varepsilon}$ is equals the sum of at most $(n+1)^{O(m)}$ terms, each of which is of the form*

$$\pm\widetilde{\mathbf{B}_{i_0\cdots i_1}}\widetilde{\mathbf{B}_{i_1\cdots i_2}}\cdots\widetilde{\mathbf{B}_{i_{r-1}\cdots i_r}}, \tag{3}$$

where $0=i_0<i_1<i_2<\cdots<i_r=n$ and $r\le 2m+1$.

Notice that, up to the sign, each term as expressed in Equation (3) is the transition matrix for a pseudorandom walk from layer 0 to layer n of the branching program, where we use $r\le m+1$ independent draws from Nisan's generator, with the j'th draw being used to walk from layer i_{j-1} to layer i_j. In particular, the (v_0,v_{acc}) entry of Equation (3) equals the acceptance probability of the branching program on such a pseudorandom walk (up to the $\pm$ sign). Thus the algorithm now has the form required of a WPRG.

The seed length for the WPRG is the sum of the seed length s_{sum} needed to select a random term in the sum (using the coefficients of the WPRG to rescale the sum into a expectation) and the seed length s_{term} to generate a walk for the individual term. To select a random term in the sum requires a seed of length

$$s_{\mathrm{sum}}=\log n^{O(m)}=O(m\cdot\log(n))=O(\log_{nw}(1/\varepsilon)\cdot\log(n))=O(\log(1/\varepsilon)).$$

The seed length needed for an individual term is at most

$$s_{\mathrm{term}}=O(m)\cdot s_{\mathrm{Nisan}}=O(\log_{nw}(1/\varepsilon)\cdot\log(n)\cdot\log nw)=O(\log(1/\varepsilon)\cdot\log(n)).$$

The latter offers no improvement over Nisan's PRG. (Recall that $\varepsilon<1/nw$.) To obtain a shorter seed length, we just need to derandomize the product in Equation (3). Instead of using r independent seeds, we use dependent seeds generated using the Impagliazzo–Nisan–Wigderson pseudorandom generator [21]. Specifically, we can produce a pseudorandom walk that approximates the product to within entrywise error $\pm\gamma$ using a seed of length

$$s'_{\mathrm{term}}=s_{\mathrm{Nisan}}+O((\log r)\cdot\log(rw/\gamma)).$$

The entrywise error of γ in each term may accumulate over the $n^{O(m)}$ terms, so to achieve a WPRG error of $O(\varepsilon)$, we should set $\gamma=\varepsilon/n^{O(m)}=1/\varepsilon^{O(1)}$. Recalling that $r\le 2m+1=O(\log_{nw}(1/\varepsilon))$, we attain a seed length of

$$\begin{aligned}s_{\mathrm{sum}}+s'_{\mathrm{term}} &= O(\log(1/\varepsilon))+O(\log n\cdot\log nw)+O(\log\log_{nw}(1/\varepsilon)\cdot\log(1/\varepsilon))\\ &= O(\log n\cdot\log nw+\log(1/\varepsilon)\cdot\log\log_{nw}(1/\varepsilon)),\end{aligned}$$

which slightly improves over the bound of Braverman, Cohen, and Garg [9], and is incomparable to that of Chattopadhyay and Liao [12]. Specifically, our first term of $O(\log n\cdot\log nw)$ is better than [12] by a factor of $\log\log(nw)$, but our second term of $O(\log(1/\varepsilon)\cdot\log\log_{nw}(1/\varepsilon))$ is worse by a factor of $\log\log_{nw}(1/\varepsilon)$.

2.2 WPRG for Permutation Branching Programs

Now we give an overview of our WPRG for permutation branching programs, as stated in Theorem 4. This is based on the the algorithm of Ahmademnijad et al. that estimates random-walk probabilities in *regular* (or even Eulerian) digraphs with better space complexity than the algorithm described in Subsection 2.1. As before, we will review their algorithm as applied to the $((n+1)\cdot w)$-vertex graph $\mathcal{G}$ associated with an ordered branching program B of length n and width w. Since we assume that the branching program B is a permutation program, the graph $\mathcal{G}$ will be 2-regular at all layers other than 0 and n. For the spectral graph-theoretic machinery used by Ahmadenijad et al., it is helpful to work with random-walk matrices that correspond to strongly connected digraphs, so we also add a complete bipartite graph of edges from layer n back to layer 0, resulting in the following modified version of the matrix $\mathbf{W}$:

$$\mathbf{W}_0 = \begin{bmatrix} 0 & \mathbf{B}_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{B}_2 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & \mathbf{B}_n \\ \mathbf{J}_w & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{4}$$

where the $\mathbf{J}_w$ in the lower-left corner is the $w \times w$ matrix in which every entry is $1/w$ (corresponding to the complete bipartite graph we added). Notice that the matrix $\mathbf{J}_w$ is identically zero when applied to any vector that is orthogonal to the uniform distribution, so it is not very different than having 0 in the lower-left block as we had before. Indeed, the powers of $\mathbf{W}$ look as follows:

$$\mathbf{W}_0^2 = \begin{bmatrix} 0 & 0 & \mathbf{B}_{0..2} & 0 & 0 \\ \vdots & 0 & 0 & \ddots & 0 \\ 0 & & \vdots & & \mathbf{B}_{n-2..n} \\ \mathbf{J}_w & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{J}_w & 0 & \cdots & 0 \end{bmatrix}, \ldots, \mathbf{W}_0^n = \begin{bmatrix} 0 & 0 & \cdots & 0 & \mathbf{B}_{0..n} \\ \mathbf{J}_w & 0 & & & 0 \\ 0 & \ddots & & & 0 \\ \vdots & 0 & \mathbf{J}_w & 0 & 0 \\ 0 & 0 & 0 & \mathbf{J}_w & 0 \end{bmatrix} \tag{5}$$

where

$$\mathbf{B}_{i\ldots j} = \mathbf{B}_{i+1}\mathbf{B}_{i+2}\cdots\mathbf{B}_j.$$

Notice in particular that $\mathbf{W}_0^{n+1}$ will be a block-diagonal matrix with $\mathbf{J}_w$'s on the diagonal (i.e. $\mathbf{W}_0^{n+1} = \mathbf{I}_{n+1} \otimes \mathbf{J}_w$), and thus has no dependence on the branching program B.

Now the Laplacian $\mathbf{I}_{(n+1)w} - \mathbf{W}_0$ is no longer invertible (the uniform distribution is in the kernel). In [2], they instead estimate the Moore-Penrose pseudoinverse of $\mathbf{I}_{(n+1)w} - \mathbf{W}_0$. We instead scale $\mathbf{W}_0$ by a factor $c = 1 - 1/(n+1)$, and consider the Laplacian $\mathbf{L}_0 = \mathbf{I}_{(n+1)w} - c\mathbf{W}_0$. Looking ahead, this scaling factor ensures that the condition number of $\mathbf{L}_0$ depends only on n, allowing us to obtain a seed length independent of w. Then, by the expressions above for the powers of $\mathbf{W}_0$, it can be shown that from

$$\mathbf{L}_0^{-1} = \mathbf{I}_{(n+1)w} + c\mathbf{W}_0 + c^2\mathbf{W}_0^2 + c^3\mathbf{W}_0^3 + \ldots$$

we can compute $\mathbf{B}_{0..n}$, which appears in $\mathbf{W}_0^n$ with a scaling factor $c^n \geq 1/4$.

So again to estimate the acceptance probability of B, it suffices to compute a sufficiently good approximation to $\mathbf{L}_0^{-1}$. As before, it suffices to compute a matrix $\widetilde{\mathbf{L}_0^{-1}}$ such that $\|\mathbf{I}_N - \widetilde{\mathbf{L}_0^{-1}}\mathbf{L}_0\| \leq \alpha$ for some constant $\alpha < 1$ and a submultiplicative matrix norm $\|\cdot\|$, because then we can use preconditioned Richardson iterations (Lemma 7) to estimate $\mathbf{L}_0$ to within arbitrary entrywise accuracy.

Unfortunately, we don't know how to directly obtain such an initial approximation $\widetilde{\mathbf{L}_0^{-1}}$ efficiently enough for our result. Instead, following Ahmadenijad et al., we tensor $\mathbf{W}_0$ with a sufficiently long directed cycle. Specifically, we let $\mathbf{C}_i$ be the directed cycle on 2^i vertices, and consider $\mathbf{C}_q$ for $q = \log(n+1)$ (which we assume is an integer WLOG). We consider the *cycle lift*, whose transition matrix is

$$\mathbf{C}_q \otimes \mathbf{W}_0 = \begin{bmatrix} 0 & \mathbf{W}_0 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{W}_0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & \mathbf{W}_0 \\ \mathbf{W}_0 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

Then, we seek to invert the Laplacian $\mathbf{L} = \mathbf{I}_{2^q N} - c\mathbf{C}_q \otimes \mathbf{W}_0$. Similarly to the above, we have:

$$\begin{aligned}\mathbf{L}^{-1} &= (\mathbf{I}_{2^q N} - c\mathbf{C}_q \otimes \mathbf{W}_0)^{-1} \\ &= \left(\mathbf{I}_{2^q N} - c^{n+1}\mathbf{C}_q^{n+1} \otimes \mathbf{W}_0^{n+1}\right)^{-1} \cdot \left(\mathbf{I}_{2^q N} + c\mathbf{C}_q \otimes \mathbf{W}_0 + c^2\mathbf{C}_q^2 \otimes \mathbf{W}_0^2 + \cdots c^n\mathbf{C}_q^n \otimes \mathbf{W}_0^n\right) \\ &= \left(\mathbf{I}_{2^q N} - c^{n+1}\mathbf{C}_q^{n+1} \otimes (\mathbf{I}_{n+1} \otimes \mathbf{J}_w)\right)^{-1} \cdot \left(\mathbf{I}_{2^q N} + c\mathbf{C}_q \otimes \mathbf{W}_0 + c^2\mathbf{C}_q^2 \otimes \mathbf{W}_0^2 + \cdots c^n\mathbf{C}_q^n \otimes \mathbf{W}_0^n\right).\end{aligned}$$

Thus, letting

$$\mathbf{M} = \mathbf{I}_{2^q N} - c^{n+1}\mathbf{C}_q^{n+1} \otimes (\mathbf{I}_{n+1} \otimes \mathbf{J}_w) = \mathbf{I}_{2^q N} - c^{n+1}\mathbf{I}_{2^q} \otimes (\mathbf{I}_{n+1} \otimes \mathbf{J}_w),$$

which has no dependence on the branching program, we have:

$$\begin{aligned}\mathbf{M} \cdot \mathbf{L}^{-1} &= \mathbf{I}_{2^q N} + c\mathbf{C}_q \otimes \mathbf{W}_0 + c^2\mathbf{C}_q^2 \otimes \mathbf{W}_0^2 + \cdots c^n\mathbf{C}_q^n \otimes \mathbf{W}_0^n \\ &= \begin{bmatrix} \mathbf{I}_N & c\mathbf{W}_0 & c^2\mathbf{W}_0^2 & \cdots & c^n\mathbf{W}_0^n \\ c^n\mathbf{W}_0^n & \mathbf{I}_N & c\mathbf{W}_0 & \cdots & c^{n-1}\mathbf{W}_0^{n-1} \\ \vdots & & \ddots & & \vdots \\ c^2\mathbf{W}_0^2 & c^3\mathbf{W}_0^3 & c^4\mathbf{W}_0^4 & \ddots & c\mathbf{W}_0 \\ c\mathbf{W}_0^1 & c^2\mathbf{W}_0^2 & c^3\mathbf{W}_0^3 & \cdots & \mathbf{I}_N \end{bmatrix}\end{aligned}$$

Thus, if we can accurately estimate $\mathbf{L}^{-1}$, we can obtain an accurate estimate of $\mathbf{W}_0^n$, whose upper-right block equals $\mathbf{B}_{0..n}$ and thus contains the acceptance probability of the branching program.

To compute an approximate inverse of $\mathbf{L} = \mathbf{I}_{2^q N} - c\mathbf{C}_q \otimes \mathbf{W}_0$, Ahmadenijad et al. provide a recursive formula expressing $(\mathbf{I}_{2^q N} - c\mathbf{C}_q \otimes \mathbf{W}_0)^{-1}$ in terms of $(\mathbf{I}_{2^{q-1}N} - c^2\mathbf{C}_{q-1} \otimes \mathbf{W}_0^2)^{-1}$ and some applications of the matrix $\mathbf{W}_0$. That is, computing the inverse of the Laplacian of the cycle lift of $\mathbf{W}_0$ reduces to computing the inverse of the Laplacian of a cycle lift of $\mathbf{W}_0^2$ with a cycle of half the length. At the bottom of the recursion (after q levels of recursion), we need to compute the inverse of

$$\mathbf{I}_N - c^{2^q}\mathbf{W}_0^{2^q} = \mathbf{I}_N - c^{n+1}\mathbf{W}_0^{n+1} = \mathbf{I}_N - c^{n+1}\mathbf{I}_{n+1} \otimes \mathbf{J}_w,$$

which is easy (and does not depend on the branching program). The resulting formula for $(\mathbf{I}_{2^qN} - c\mathbf{C}_q \otimes \mathbf{W}_0)^{-1}$ is a polynomial in $\mathbf{W}_0, \mathbf{W}_0^2, \mathbf{W}_0^4, \ldots, \mathbf{W}_0^{2^{q-1}}$. However, computing these high powers of $\mathbf{W}_0$ exactly is too expensive in space usage.

Thus, instead Ahmadenijad et al. use the *derandomized square* [30] which allows for computing a sequence $\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_q$ where $\mathbf{W}_i$ a sparsification of $\mathbf{W}_{i-1}^2$ with the property that $\mathbf{W}_q$ can be constructed in deterministic space

$$O(\log nw + q \cdot \log(1/\delta))$$

for an error parameter δ, rather than the space $O(q \cdot \log nw)$ of exact repeated squaring. They also introduce a new notion of spectral approximation, called *unit-circle approximation*, and show that the derandomized square $\mathbf{W}_i$ is a unit-circle approximation of $\mathbf{W}_{i-1}^2$ to within error δ. Using repeated derandomized squaring in the recursion, Ahmadenijad et al. obtain an approximate inverse $\widetilde{\mathbf{L}^{-1}}$ with the properties that:

1. The $N \times N$ blocks of $\mathbf{M} \cdot \widetilde{\mathbf{L}^{-1}}$ are each of the form $\mathbf{W}_{i_1}\mathbf{W}_{i_2} \cdots \mathbf{W}_{i_r}$ where $r = O(q)$
2. There is a submultiplicative matrix norm $\| \cdot \|_{\mathbf{F}}$ such that $\|\mathbf{I}_{2^qN} - \widetilde{\mathbf{L}^{-1}}\mathbf{L}\|_{\mathbf{F}} = O(q^2\delta)$. Moreover, achieving an $\varepsilon/\operatorname{poly}(n)$ approximation in $\mathbf{F}$-norm implies an ε approximation of $\mathbf{M} \cdot \mathbf{L}^{-1}$ in max-norm. Ahmadenijad et al. actually lose a factor of $\operatorname{poly}(nw)$ in moving from $\mathbf{F}$-norm to approximation in max-norm, but we improve this bound to $\operatorname{poly}(n)$ by our choice of scaling factor $c = 1 - 1/(n+1)$.

Item 1 allows for constructing $\mathbf{M} \cdot \widetilde{\mathbf{L}^{-1}}$ from $\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_q$ in space

$$O(\log q \cdot \log nw).$$

By Item 2, if we take $\delta < 1/O(q^2)$, we can apply preconditioned Richardson iterations (Lemma 7) with degree $m = O(\log(n/\varepsilon)/\log(1/q\delta))$ to obtain $\widetilde{\mathbf{L}_\varepsilon} = \mathbf{P}_m$ such that $\mathbf{M} \cdot \widetilde{\mathbf{L}_\varepsilon}$ approximates $\mathbf{M}\mathbf{L}^{-1}$ to within entrywise error ε. The preconditioned Richardson iterations have an additive space cost of:

$$O(\log m \cdot \log nw).$$

Taking $\delta = 1/O(q^2)$ and recalling that $q = \log(n+1)$, the final space complexity is

$$O(\log(nw) + q \log q) + O(\log q \cdot \log nw) + O(\log\log(n/\varepsilon) \cdot \log nw) = O(\log nw \cdot \log\log(n/\varepsilon)).$$

To view this algorithm as a WPRG for permutation branching programs, we use the equivalence between the Impagliazzo–Nisan–Wigderson (INW) generator on permutation branching programs and the derandomized square of the corresponding graph, as established in [30, 20]. Using this correspondence, the matrix $\mathbf{W}_i$ has the same structure as $\mathbf{W}^{2^i}$ (see Equation 5), except that each block of the form $\mathbf{B}_{j..j+2^i}$ is replaced with a matrix $\widetilde{\mathbf{B}_{j..j+2^i}}$ that is the transition matrix of a pseudorandom walk from layer j of the branching program to layer $j + 2^i$ using the INW generator. The seed length to generate this pseudorandom walk is

$$s_{\text{INW}} = O(q \log(q/\delta)),$$

which, as highlighted in [20], is independent of the width w of the branching program. This is the place where we use the fact that B is a permutation branching program rather than a regular branching program. Even though the algorithm Ahmadenijad et al. works for regular directed graphs (and hence regular branching programs), the derandomized square operations used in that case can no longer be viewed as being obtained by using a pseudorandom generator to derandomize walks in the graph.

Then, again assuming without loss of generality that $\widetilde{\mathbf{B}_{(j-1)\ldots j}} = \mathbf{B}_{(j-1)\ldots j}$ for $j = 1, \ldots, n$, we have the following analogue of Observation 8:

▶ **Observation 9.** *The upper-right $w \times w$ block of $\mathbf{M} \cdot \widetilde{\mathbf{L}_\varepsilon}$ equals the sum of at most $n^{O(m)}$ terms, each of which is of the form*

$$\pm\widetilde{\mathbf{B}_{i_0\cdots i_1}}\widetilde{\mathbf{B}_{i_1\cdots i_2}}\cdots\widetilde{\mathbf{B}_{i_{r-1}\cdots i_r}}, \tag{6}$$

where $0 = i_0 < i_1 < i_2 < \cdots < i_r = n$ and $r = O(qm)$.

As in Subsection 2.1, the algorithm now has the form required of a WPRG and our only remaining challenge is to keep the seed length small. The seed length for the WPRG is the sum of the seed length needed to select a random term in the sum (using the coefficients of the WPRG to rescale the sum into a expectation) and the seed length to generate a walk for the individual term. To select a random term in the sum requires a seed of length

$$s_{\text{sum}} = \log(n^{O(m)}).$$

The seed length needed for an individual term is at most

$$s_{\text{term}} = O(qm) \cdot s_{\text{INW}},$$

which again would be too expensive for us. To derandomize the product in Equation (6), we again use the INW generator, but rely on the analysis in [20] for permutation branching programs to maintain a seed length that is independent of the width. Specifically, we can produce a pseudorandom walk that approximates the product to within entrywise error $\pm\gamma$ using a seed of length

$$s'_{\text{term}} = s_{\text{INW}} + O((\log r) \cdot \log(\log(r)/\gamma)) = s_{\text{INW}} + O(\log qm \cdot \log(\log(qm)/\gamma)).$$

The entrywise error of γ in each term may accumulate over the $n^{O(m)}$ terms, so to achieve a WPRG error of $O(\varepsilon)$, we should set $\gamma = \varepsilon/n^{O(m)}$, which means that $s'_{\text{term}} \geq s_{\text{sum}}$.

All in all, we attain a seed length of

$$\begin{aligned} s_{\text{sum}} + s'_{\text{term}} &= O(m\log n) + s_{\text{INW}} + O((\log qm) \cdot \log(\log(qm)/\gamma)) \\ &= O(q\log(q/\delta)) + \tilde{O}(m\log n) + O(\log qm \cdot \log(n/\varepsilon)) \\ &= \tilde{O}\left(\log n \cdot \log(1/\delta) + \frac{\log(n/\varepsilon)}{\log(1/(\delta\log n))} \cdot \log n + \log\log(n/\varepsilon) \cdot \log(n/\varepsilon)\right) \end{aligned}$$

Optimizing the choice of δ as $\delta = \exp(-\tilde{\Theta}(\sqrt{\log(n/\varepsilon)}))$, we get a seed length of

$$\tilde{O}(\log n\sqrt{\log(n/\varepsilon)} + \log(1/\varepsilon)).$$

Note that the choice of δ here is much smaller than in the Ahmadenijad et al. algorithm, which used $\delta = 1/\operatorname{polylog}(n)$. The reason we need the smaller choice of δ is to reduce the effect of the $\log(n^{O(m)})$ price we pay in s_{sum} and s'_{term}, which does not have an analogue in the algorithm of Ahmadenijad et al.

2.3 Perspective

Some intuition for the ability of WPRGs to beat the parameters of PRGs can come from the study of *samplers* [16]. A *sampler* for a class $\mathcal{F}$ of functions $f : \{0,1\}^m \to \mathbb{R}$ is randomized algorithm Samp that is given oracle access to a function $f \in \mathcal{F}$ and, with probability at least $1 - \delta$, outputs an estimate of $\mathbb{E}[f(U_n)]$ to within additive error $\pm\varepsilon$. Most often, the class $\mathcal{F}$ is taken to be the class of all bounded functions $f : \{0,1\}^m \to [0,1]$, but some works have considered the general definition and other classes, such as the class $\mathcal{F}$ of unbounded functions f such that the random variable $f(U_n)$ has subgaussian tails [6, 1]. Two key complexity parameters of a sampler are its *randomness complexity* (the number of coin tosses it uses, typically as a function of m, δ, and ε) and its *sample complexity* (the number of queries it makes to oracle f). An *averaging sampler* is one that has a restricted form, where it uses its coin tosses to generate (possibly correlated) samples $x_1, \dots, x_S$, and then outputs the average of f on the samples, i.e. $(f(x_1) + \cdots + f(x_S))/S$.

As noted by Cheng and Hoza [13], PRGs and WPRGs can be viewed as deterministic averaging samplers (i.e. with randomness complexity and failure probability zero). Specifically, a PRG $G : \{0,1\}^s \to \{0,1\}^m$ for a class $\mathcal{F}$ is a deterministic averaging sampler for the class $\mathcal{F}$ with sample complexity $S = 2^s$. Indeed, the sampler simply outputs the set of all $S = 2^s$ outputs of G. A WPRG as a more general form of a nonadaptive deterministic sampler for the class $\mathcal{F}$, one that is restricted to output a linear combination of the function values.

So comparing the power of PRGs vs. WPRGs is a special case of the more general problem of comparing the power of averaging samplers vs. more general nonadaptive samplers. In this more general framing, there are some natural examples of classes $\mathcal{F}$ where nonadaptive samplers can have smaller sample complexity than any averaging sampler. Specifically, if we consider the class $\mathcal{F}$ of *unbounded* functions $f : \{0,1\}^m \to \mathbb{R}$ with bounded variance, i.e. $\mathrm{Var}[f(U_n)] \leq 1$, then the best sample complexity for an averaging sampler is $\Theta(\min\{1/\varepsilon^2\delta, 2^m\})$. (Essentially, Chebychev's Inequality is tight for such functions.) However, there is a nonadaptive sampler with sample complexity $O(\log(1/\delta)/\varepsilon^2)$, namely the *median-of-averages sampler*, which outputs the median of $O(\log(1/\delta))$ averages, with each average being on $O(1/\varepsilon^2)$ samples.

This example suggests two areas of investigation. First, can we gain further benefits in seed length by considering further generalizations of PRGs that are allowed to estimate acceptance probability with more general functions than linear combinations (or possibly even with adaptive queries)? Some examples are the line of work on converting hitting-set generators for circuits [3, 4, 11, 18] or ordered branching programs [13] into deterministic samplers. Second, is there a benefit in the study of samplers in restricting attention to ones that output linear combinations like WPRGs? Perhaps these still retains some of the useful composition properties and connections to other pseudorandom objects that are enjoyed by averaging samplers (cf. [36, 34, 1]), while allowing for gains in sample and/or randomness complexity.

References

1. Rohit Agrawal. Samplers and extractors for unbounded functions. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, volume 145 of *LIPIcs*, pages 59:1–59:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.59.
2. AmirMahdi Ahmadinejad, Jonathan A. Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil P. Vadhan. High-precision estimation of random walks in small space. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1295–1306. IEEE, 2020. doi:10.1109/FOCS46700.2020.00123.
3. Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. A new general derandomization method. *Journal of the ACM*, 45(1):179–213, 1998. doi:10.1145/273865.273933.

4 Alexander E. Andreev, Andrea E. F. Clementi, José D. P. Rolim, and Luca Trevisan. Weak random sources, hitting sets, and BPP simulations. *SIAM Journal on Computing*, 28(6):2103–2116 (electronic), 1999.

5 Roy Armoni. On the derandomization of space-bounded computations. In *Randomization and approximation techniques in computer science (Barcelona, 1998)*, volume 1518 of *Lecture Notes in Comput. Sci.*, pages 47–59. Springer, Berlin, 1998.

6 Jaroslaw Blasiok. Optimal streaming and tracking distinct elements with high probability. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2432–2448. SIAM, 2018. `doi:10.1137/1.9781611975031.156`.

7 Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. `doi:10.1137/0213053`.

8 Andrej Bogdanov, Zeev Dvir, Elad Verbin, and Amir Yehudayoff. Pseudorandomness for width 2 branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:70, 2009. URL: `http://eccc.hpi-web.de/report/2009/070`.

9 Mark Braverman, Gil Cohen, and Sumegha Garg. Hitting sets with near-optimal error for read-once branching programs. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 353–362. ACM, 2018. `doi:10.1145/3188745.3188780`.

10 Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. In *FOCS*, pages 40–47. IEEE Computer Society, 2010. `doi:10.1109/FOCS.2010.11`.

11 Harry Buhrman and Lance Fortnow. One-sided two-sided error in probabilistic computation. In *STACS 99 (Trier)*, volume 1563 of *Lecture Notes in Comput. Sci.*, pages 100–109. Springer, Berlin, 1999.

12 Eshan Chattopadhyay and Jyun-Jie Liao. Optimal error pseudodistributions for read-once branching programs. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 25:1–25:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.CCC.2020.25`.

13 Kuan Cheng and William M. Hoza. Hitting sets give two-sided derandomization of small space. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 10:1–10:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.CCC.2020.10`.

14 Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted prgs against read once branching programs. In *36th Computational Complexity Conference, CCC 2021, July 19-23, 2021, Toronto, Ontario (Virtual Conference)*, 2021. To appear.

15 Anindya De. Pseudorandomness for permutation and regular branching programs. In *IEEE Conference on Computational Complexity*, pages 221–231. IEEE Computer Society, 2011. `doi:10.1109/CCC.2011.23`.

16 Oded Goldreich. A sample of samplers - a computational perspective on sampling (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, 4(20), 1997. URL: `http://eccc.hpi-web.de/eccc-reports/1997/TR97-020/index.html`.

17 Oded Goldreich. *A primer on pseudorandom generators*, volume 55 of *University Lecture Series.* American Mathematical Society, Providence, RI, 2010.

18 Oded Goldreich, Salil Vadhan, and Avi Wigderson. Simplified derandomization of bpp using a hitting set generator. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay of Randomness and Computation*, volume 6650 of *Lecture Notes in Computer Science*, pages 59–67. Springer, 2011.

19 Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators via milder pseudorandom restrictions. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*. IEEE, 20–23 October 2012.

20 William M. Hoza, Edward Pyne, and Salil P. Vadhan. Pseudorandom generators for unbounded-width permutation branching programs. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 7:1–7:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.ITCS.2021.7`.

21 Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 356–364, Montréal, Québec, Canada, 23–25 May 1994.

22 Daniel M. Kane, Jelani Nelson, and David P. Woodruff. Revisiting norm estimation in data streams. *CoRR*, abs/0811.3648, 2008. `arXiv:0811.3648`.

23 Michal Koucký, Prajakta Nimbhorkar, and Pavel Pudlák. Pseudorandom generators for group products: extended abstract. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 263–272. ACM, 2011. `doi:10.1145/1993636.1993672`.

24 Raghu Meka, Omer Reingold, and Avishay Tal. Pseudorandom generators for width-3 branching programs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 626–637. ACM, 2019.

25 Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

26 Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, October 1994.

27 Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, February 1996.

28 Edward Pyne and Salil Vadhan. Pseudodistributions that beat all pseudorandom generators. ECCC preprint TR21-019, 2021.

29 Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom walks in regular digraphs and the RL vs. L problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC '06)*, pages 457–466, 21–23 May 2006. Preliminary version as *ECCC* TR05-22, February 2005.

30 Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM '05)*, number 3624 in Lecture Notes in Computer Science, pages 436–447, Berkeley, CA, August 2005. Springer.

31 Michael Saks and Shiyu Zhou. $\mathrm{BP_H SPACE}(S) \subseteq \mathrm{DSPACE}(S^{3/2})$. *Journal of Computer and System Sciences*, 58(2):376–403, 1999.

32 Jirí Síma and Stanislav Zák. Almost k-wise independent sets establish hitting sets for width-3 1-branching programs. In Alexander S. Kulikov and Nikolay K. Vereshchagin, editors, *CSR*, volume 6651 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2011. `doi:10.1007/978-3-642-20712-9_10`.

33 Thomas Steinke. Pseudorandomness for permutation branching programs without the group theory. Technical Report TR12-083, Electronic Colloquium on Computational Complexity (ECCC), July 2012. URL: `http://eccc.hpi-web.de/report/2012/083/`.

34 Salil P Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

35 Andrew C. Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, 3–5 November 1982. IEEE.

36 David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997.

GSF-Locality Is Not Sufficient For Proximity-Oblivious Testing

Isolde Adler
School of Computing, University of Leeds, UK

Noleen Köhler
School of Computing, University of Leeds, UK

Pan Peng
Department of Computer Science, University of Sheffield, UK

Abstract

In Property Testing, *proximity-oblivious testers (POTs)* form a class of particularly simple testing algorithms, where a basic test is performed a number of times that may depend on the proximity parameter, but the basic test itself is independent of the proximity parameter.

In their seminal work, Goldreich and Ron [STOC 2009; SICOMP 2011] show that the graph properties that allow constant-query proximity-oblivious testing in the bounded-degree model are precisely the properties that can be expressed as a *generalised subgraph freeness (GSF)* property that satisfies the *non-propagation* condition. It is left open whether the non-propagation condition is necessary. Indeed, calling properties expressible as a generalised subgraph freeness property *GSF-local properties*, they ask whether all GSF-local properties are non-propagating. We give a negative answer by exhibiting a property of graphs that is GSF-local and propagating. Hence in particular, our property does not admit a POT, despite being GSF-local. We prove our result by exploiting a recent work of the authors which constructed a first-order (FO) property that is not testable [SODA 2021], and a new connection between FO properties and GSF-local properties via neighbourhood profiles.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms; Theory of computation → Finite Model Theory

Keywords and phrases Property testing, proximity-oblivous testing, locality, first-order logic, lower bound

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.34

1 Introduction

Graph property testing is a framework for studying sampling-based graph algorithms. Given a graph property $\mathcal{P}$, the goal is to design a (randomised) algorithm, called *tester*, that distinguishes between graphs that satisfy $\mathcal{P}$ from those that are "far" from satisfying $\mathcal{P}$, where the notion "being far" depends on the underlying query access model and is always parametrised by a *proximity parameter* $\varepsilon > 0$. The query model also specifies the class of graphs and the types of queries allowed by the algorithm. The two most well known models for graph property testing are the *dense graph model* and the *bounded-degree graph model* (see [9]). Towards an understanding of which graph properties are testable with a constant number of queries in each model, much progress has been made since the framework of property testing was introduced [24, 10]. To illustrate, a full characterization of the properties that are testable with a constant number of queries in the dense graph model has been obtained by Alon, Fischer, Newman, and Shapira [2].

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 34; pp. 34:1–34:27
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Typical property testers make decisions regarding the global property of the graph from the local views. In the extreme case, a tester could make local views independent of the distance to a predetermined set of graphs. Motivated by this, Goldreich and Ron [13] initiated the study of (one-sided error) *proximity-oblivious testers (POTs)* for graphs, where a tester simply repeats a basic test for a number of times that depends on the proximity parameter, and the basic tester is oblivious of the proximity parameter. They gave characterizations of graph properties that can be tested with constant query complexity by a POT in both dense graph model and the bounded-degree model. In each model, it is known that the class of properties that have constant-query POTs is a strict subset of the class of properties that are testable (by standard testers).

In this paper, we focus on the bounded-degree graph model [12]. In this model, the algorithm is given query access to an input graph with maximum degree bounded by d, where d is some constant. For any specified query v and an index $i \leq d$, the algorithm can obtain the i-th neighbor of v if it exists, and a special symbol $\perp$ otherwise. Given a proximity parameter $\varepsilon > 0$, an n-vertex graph with maximum degree at most d is said to be ε-far from a property $\mathcal{P}$ if one needs to add and/or delete more than εdn edges to make it satisfy $\mathcal{P}$. A property is said to be *testable* if there exists a tester that makes only a *constant* number of queries to the input graph G, and distinguishes if G satisfies the property $\mathcal{P}$ or is ε-far from satisfying $\mathcal{P}$, with success probability at least $\frac{2}{3}$. Here the constant is a number that might depend on ε and d, but is independent of the size of the input graph. It has been known that many properties are testable, such as subgraph-freeness, k-edge connectivity, cycle-freeness, being Eulerian, degree-regularity [12], minor-freeness [3, 16, 20], hyperfinite properties [22], k-vertex connectivity [25, 7], and subdivision-freeness [19].

Turning to POTs, informally, a (one-sided error) POT for a property $\mathcal{P}$ is a tester that always accepts a graph G if it satisfies $\mathcal{P}$, and rejects G with probability that is a monotonically increasing function of the distance of G from the property $\mathcal{P}$. We say $\mathcal{P}$ is *proximity-oblivious testable* if such a tester exists for $\mathcal{P}$ with constant query complexity. To characterise the class of proximity-oblivious testable properties in the bounded-degree model, Goldreich and Ron [13] introduced a notion of generalized subgraph freeness (GSF), that extends the notions of induced subgraph freeness and (non-induced) subgraph freeness. A graph property is called a *GSF-local* property if it is expressible as a GSF property. It has been shown in [13] that a graph property is constant-query proximity-oblivious testable if and only if it is a GSF-local property that satisfies a so-called *non-propagation* condition. Informally, a GSF-local property $\mathcal{P}$ is non-propagating if repairing a graph G that does not satisfy $\mathcal{P}$ does not trigger a global "chain reaction" of necessary modifications. We refer Section 2.3 for formal definitions.

A major question that is left open is whether every GSF-local property satisfies the non-propagation condition.

1.1 Our contribution

In this paper, we resolve the aforementioned open question raised in [13] by showing the following negative result.

▶ **Theorem 1** (Main result). *There exists a GSF-local property that is not testable in the bounded-degree graph model. Thus, not all GSF-local properties are non-propagating.*

We expect our result would shed some light on a full characterization of testable properties in the bounded degree model. Indeed, in the recent work by Ito, Khoury and Newman [18], the authors gave a characterization of testable *monotone* graph properties and testable *hereditary* graph properties with one-sided error in the bounded-degree graph model; and they asked the open question "*is every property that is defined by a set of forbidden configurations testable?*" Since their definition of a property defined by a set of "forbidden configuration" is equivalent to a GSF-local property, our main result also gives a negative answer to their question.

1.2 Proof outline

The starting point of our proof is a recent result of the authors that there exists a first-order (FO) property that is not testable in the bounded degree graph model [1], where a property $\mathcal{P}$ is said to be an FO property if it can be expressed by an FO formula, i. e. a quantified formula whose variables represent graph vertices, with predicates for equality and adjacency. Intuitively, each structure in the property given in [1] is a hybridization of a sequence of expander graphs and a tree structure, where the expander graphs are recursively constructed by the zig-zag product introduced by Reingold et al. [23]. Here each level of the tree structure forms one member of the recursive sequence of expander graphs. It was shown that this property is both an FO property and a family of expanders, and the latter implies it is not testable (see e. g. [6]). We refer to Section 4 and [1] for a detailed description of the property.

By Gaifman's locality theorem [8], it is known that FO can only express local properties. Indeed, Hanf's Theorem [15] implies that we can understand this locality as prescribing upper and lower bounds for the occurrence of certain local neighbourhood (isomorphism) types.

On the other hand, a GSF-local property as defined in [13] refers to the freeness of some constant-size *marked* graphs, where a mark graph F specifies an induced subgraph and how it "interacts" with the rest of the graph (see Definition 3). Intuitively, such a property just specifies a condition that the local neighbourhoods of a graph G should satisfy, i.e., certain types of local neighbourhoods cannot not occur in G, or equivalently, these types have 0 occurrences.

Building upon the above observations, we establish a formal connection between FO properties and GSF-local properties. We first encode the possible bounds on occurrences of local neighbourhood types into what we call *neighbourhood profiles*, and characterise FO definable properties of bounded degree relational structures as finite unions of properties defined by neighbourhood profiles (Lemma 9). We then show that every FO formula defined by a non-trivial finite union of properties which in turn is defined by a so-called 0*-profiles*, i. e. the prescribed lower bounds are all 0, is GSF-local (Theorem 11). Given the fundamental roles of local properties in graph theory, graph limits [21], we believe this new connection is of independent interest.

For technical reasons, we make use of a property $\mathcal{P}_{\textcircled{z}}$ of *relational structures* that can be expressed by some FO formula while it is *not* testable in the bounded-degree model, instead of directly using the non-testable graph property from [1]. We further prove that a minor variant of the relational structure property $\mathcal{P}_{\textcircled{z}}$, which we denote by $\mathcal{P}'_{\textcircled{z}}$, can be defined by 0-profiles (Lemma 20). Finally, we construct a non-testable *graph* property $\mathcal{P}_{\text{graph}}$ by a *local* reduction from the σ-structure property $\mathcal{P}'_{\textcircled{z}}$ (Lemma 24). In the reduction we maintain being definable by 0-profiles which proves GSF-locality of the graph property $\mathcal{P}_{\text{graph}}$ (Lemma 25). Intuitively, the property $\mathcal{P}_{\text{graph}}$ encodes the property $\mathcal{P}_{\textcircled{z}}$ in undirected graphs. Again, $\mathcal{P}_{\text{graph}}$ is a family of expanders (which guarantees non-testability), where in addition the local neighbourhoods satisfy the aforementioned features which guarantee that it is an FO property and also GSF-local.

1.3 Other related work

The notion of POT was implicitly defined in [4]. Goldreich and Shinkar [14] studied two-sided error POTs for both dense graph and bounded-degree graph models. Goldreich and Kaufman [11] investigated the relation between local conditions that are invariant in an adequate sense and properties that have a constant-query proximity-oblivious testers. Fichtenberger et al. [6] showed that every testable property is either finite or contains an infinite hyperfinite subproperty.

2 Preliminaries

2.1 Graphs, relational structures and first-order logic

We will briefly introduce structures and first-order logic and point the reader to [5] for a more detailed introduction. A (relational) *signature* is a finite set $\sigma = \{R_1, \dots, R_\ell\}$ of relation symbols R_i. Every relation symbol R_i has an arity $\text{ar}(R_i) \in \mathbb{N}_{>0}$. A *$\sigma$-structure* is a tuple $A = (U(A), R_1(A), \dots, R_\ell(A))$, where $U(A)$ is a *finite* set, called the *universe* of A and $R_i(A) \subseteq U(A)^{\text{ar}(R_i)}$ is an $\text{ar}(R_i)$-ary relation on $U(A)$. Note that if $\sigma = \{E_1, \dots, E_\ell\}$ is a signature where each E_i is a binary relation symbol, then σ-structures are directed graphs with ℓ edge-colours. Let $\sigma_{\text{graph}} := \{E\}$ be a signature with one binary relation symbol E. Then we can understand undirected graphs as σ_{graph}-structures for which the relation E is symmetric (every undirected edge is represented by two tuples). Using this we can transfer all notions defined below for graphs. Typically we name graphs G, H, F, we denote the set of vertices of a graph G by $V(G)$, the set of edges by $E(G)$ and vertices are typically named $u, v, w, u', v', w', \dots$. In contrast when we talk about a general relational structure we use A, B and $a, b, a', b', \dots$ to denote elements from the universe.

In the following we let σ be a relational signature. Two σ-structures A and B are *isomorphic* if there is a bijective map from $U(A)$ to $U(B)$ that preserves all relations. For a σ-structure A and a subset $S \subseteq U(A)$, we let $A[S]$ denote the *substructure* of A *induced* by S, i. e. $A[S]$ has universe S and $R(A[S]) := R(A) \cap S^{\text{ar}(R)}$ for all $R \in \sigma$. The *degree* of an element $a \in U(A)$ denoted by $\deg_A(a)$ is defined to be the number of tuples in A containing a. We define the *degree* of A, denoted by $\deg(A)$, to be the maximum degree of its elements. Given a signature σ and a constant d, we let $\mathcal{C}_{\sigma,d}$ be the class of bounded-degree d σ-structures and $\mathcal{C}_d$ the set of all bounded-degree d graphs. Note that the degree of a graph differs by exactly a factor 2 from the degree of the corresponding σ_{graph}-structure.

Syntax and semantic of FO is defined in the usual way (see e. g. [5]). We use $\exists^{\geq m} x\, \varphi$ (and $\exists^{= m} x\, \varphi$, $\exists^{\leq m} x\, \varphi$, respectively) as a shortcut for the FO formula expressing that the number of witnesses x satisfying φ is at least m (exactly m, at most m, respectively). We say that a variable occurs *freely* in an FO formula if at least one of its occurrences is not bound by any quantifier. We use $\varphi(x_1, \dots, x_k)$ to express that the set of variables which occur freely in the FO formula φ is a subset of $\{x_1, \dots, x_k\}$. For a formula $\varphi(x_1, \dots, x_k)$, a σ-structure A and $a_1, \dots, a_k \in U(A)$ we write $A \models \varphi(a_1, \dots, a_k)$ if φ evaluates to true after assigning a_i to x_i, for $1 \leq i \leq k$. A *sentence* of FO is a formula with no free variables. For an FO sentence φ we say that A is a *model* of φ or A satisfies φ if $A \models \varphi$.

The *Gaifman graph* of a σ-structure A is the undirected graph $G(A) = (U(A), E)$, where $\{v, w\} \in E$, if $v \neq w$ and there is an $R \in \sigma$ and a tuple $\overline{a} = (a_1, \dots, a_{\text{ar}(R)}) \in R(A)$, such that $v = a_j$ and $w = a_k$ for some $1 \leq k, j \leq \text{ar}(R)$. We use $G(A)$ to apply graph theoretic notions to relational structures. Note that for any graph the Gaifman graph of the corresponding symmetric σ_{graph}-structure is the graph itself. For two elements $a, b \in U(A)$, we define the *distance* between a and b in A, denoted by $\mathrm{dist}_A(a,b)$, as the length of a shortest path form a to b in $G(A)$, or ∞ if there is no such path. For $r \in \mathbb{N}$ and $a \in U(A)$, the *r-neighbourhood* of a is the set $N_r^A(a) := \{b \in U(A) : \mathrm{dist}_A(a,b) \leq r\}$. We define $\mathcal{N}_r^A(a) := A[N_r^A(a)]$ to be the substructure of A induced by the r-neighbourhood of a. For $r \in \mathbb{N}$ an *r-ball* is a tuple (B,b), where B is a σ-structure, $b \in U(B)$ and $U(B) = N_r^B(b)$, i. e. B has radius r and b is the centre. Note that by definition $(\mathcal{N}_r^A(a), a)$ is an r-ball for any σ-structure A and $a \in U(A)$. Two r-balls $(B,b),(B',b')$ are isomorphic if there is an isomorphism of σ-structure from B to B' that maps b to b'. We call the isomorphism classes of r-balls *r-types*. For an r-type τ and an element $a \in U(A)$ we say that *a has* (r-)type τ if $(\mathcal{N}_r^A(a), a) \in \tau$. Moreover, given such an r-type τ, there is a formula $\varphi_\tau(x)$ such that for every σ-structure A and for every $a \in U(A)$, $A \models \varphi_\tau(a)$ iff $(\mathcal{N}_r^A(a), a) \in \tau$. A *Hanf-sentence* is a sentence of the form $\exists^{\geq m} x \varphi_\tau(x)$, for some $m \in \mathbb{N}_{>0}$, where τ is an r-type. An FO sentence is in *Hanf normal form*, if it is a Boolean combination[1] of Hanf sentences. Two formulas $\varphi(x_1,\dots,x_k)$ and $\psi(x_1,\dots,x_k)$ of signature σ are called *d-equivalent*, if they are equivalent on $\mathcal{C}_{\sigma,d}$, i. e. for all $A \in \mathcal{C}_{\sigma,d}$ and $(a_1,\dots,a_k) \in U(A)^k$ we have $A \models \varphi(a,\dots,a_k)$ iff $A \models \psi(a_1,\dots,a_k)$. Hanf's locality theorem for first-order logic [15] implies the following.

▶ **Theorem 2** (Hanf [15]). *Let $d \in \mathbb{N}$. Every sentence of first-order logic is d-equivalent to a sentence in Hanf normal form.*

2.2 Property testing

In the following, we give definitions of two models for property testing - the bounded-degree model for graphs and the bounded-degree model for relational structures. For notational convenience, $\mathcal{C}$ will either denote a class of graphs of bounded-degree d, or a class of σ-structures of bounded-degree d for some signature σ and some $d \in \mathbb{N}$. We will further refer to both graphs and σ-structures as structures. A *property* $\mathcal{P}$ in $\mathcal{C}$ is a subset of $\mathcal{C}$ which is closed under isomorphism. We say that a structure A has property $\mathcal{P}$ if $A \in \mathcal{P}$. For $\epsilon \in (0,1)$ we say that a structure $\mathcal{A}$ on n vertices/elements is *ϵ-close* to $\mathcal{P}$ if there is a structure $A' \in \mathcal{P}$ such that A and A' differ in at most ϵdn edges/tuples. We say that $A \in \mathcal{C}$ is ϵ-far from $\mathcal{P}$ if A is not ϵ-close to $\mathcal{P}$.

A property tester accesses a structure via oracle queries. A *query* to a σ-structure A of bounded-degree d has the form (a,i) for an element $a \in U(A)$, $i \in \{1,\dots,d\}$ and is answered by $\mathrm{ans}(a,i) := (R, a_1,\dots,a_{\mathrm{ar}(R)})$ where $(a_1,\dots,a_{\mathrm{ar}(R)})$ is the i-th tuple containing a and $(a_1,\dots,a_{\mathrm{ar}(R)}) \in R(A)$. A *query* to a graph G of bounded-degree d has the form (v,i) for $v \in V(G)$, $i \in \{1,\dots,d\}$ and is answered by $\mathrm{ans}(v,i) := w$ where w is the i-th neighbour of v.

Let $\mathcal{P}_n$ be the subset of $\mathcal{P}$ with n vertices/elements. Thus $\mathcal{P} = \cup_{n \in \mathbb{N}} \mathcal{P}_n$. We give the formal definitions of standard property testing and proximity-oblivious testing in Appendix A.

2.3 Generalised subgraph freeness

Now we present the formal definition of generalised subgraph freeness, GSF-local properties and the notion of non-propagation, which were introduced in [13].

[1] By Boolean combination we always mean *finite* Boolean combination.

▶ **Definition 3** (Generalized subgraph freeness (GSF)). *A* marked *graph is a graph with each vertex marked as either* "full" *or* "semifull" *or* "partial". *An* embedding *of a marked graph F into a graph G is an injective map $f : V(F) \to V(G)$ such that for every $v \in V(F)$ the following three conditions hold.*

1. *If v is marked "full", then $N_1^G(f(v)) = f(N_1^F(v))$.*
2. *If v is marked "semifull", then $N_1^G(f(v)) \cap f(V(F)) = f(N_1^F(v))$.*
3. *If v is marked "partial", then $N_1^G(f(v)) \supseteq f(N_1^F(v))$.*

The graph G is called F-free if there is no embedding of F into G. For a set of marked graphs $\mathcal{F}$, a graph G is called $\mathcal{F}$-free if it is F-free for every $F \in \mathcal{F}$.

Based on the above definition of GSF, we can define GSF-local properties.

▶ **Definition 4** (GSF-local properties). *Let $\mathcal{P} = \cup_{n\in\mathbb{N}}\mathcal{P}_n$ be a graph property where $\mathcal{P}_n = \{G \in \mathcal{P} \mid |V(G)| = n\}$ and $\overline{\mathcal{F}} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ a sequence of sets of marked graphs. $\mathcal{P}$ is called* $\overline{\mathcal{F}}$-local *if there exists an integer s such that for every n the following conditions hold.*

1. *$\mathcal{F}_n$ is a set of marked graphs, each of size at most s.*
2. *$\mathcal{P}_n$ equals the set of n-vertex graphs that are $\mathcal{F}_n$-free.*

$\mathcal{P}$ is called GSF-local *if there is a sequence $\overline{\mathcal{F}} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ of sets of marked graphs such that $\mathcal{P}$ is $\overline{\mathcal{F}}$-local.*

The following notion of non-propagating condition of a sequence of sets of marked graphs was introduced to study constant-query POTs.

▶ **Definition 5** (Non-propagating). *Let $\overline{\mathcal{F}} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ be a sequence of sets of marked graphs.*

- *For a graph G, a subset $B \subset V(G)$* covers *$\mathcal{F}_n$ in G if for every marked graph $F \in \mathcal{F}_n$ and every embedding of F in G, at least one vertex of F is mapped to a vertex in B.*
- *The sequence $\overline{\mathcal{F}}$ is* non-propagating *if there exists a (monotonically non-decreasing) function $\tau : (0, 1] \to (0, 1]$ such that the following two conditions hold.*
  1. *For every $\epsilon > 0$ there exists $\beta > 0$ such that $\tau(\beta) < \epsilon$.*
  2. *For every graph G and every $B \subset V(G)$ such that B covers $\mathcal{F}_n$ in G, either G is $\tau(|B|/n)$-close to being $\mathcal{F}_n$-free or there are no n-vertex graphs that are $\mathcal{F}_n$-free.*

  A GSF-local property $\mathcal{P}$ is non-propagating *if there exists a non-propagating sequence $\overline{\mathcal{F}}$ such that $\mathcal{P}$ is $\overline{\mathcal{F}}$-local.*

In the above definition, the set B can be viewed as the set involving necessary modifications for repairing a graph G that does not satisfy the property $\mathcal{P}$ that is $\overline{\mathcal{F}}$-local, and the second condition says we do not need to modify G "much beyond" B. In particular, it implies we can repair G without triggering a global "chain reaction". Goldreich and Ron gave the following characterization for the proximity-oblivious testable properties in the bounded-degree graph model.

▶ **Theorem 6** (Theorem 5.5 in [13]). *A graph property $\mathcal{P}$ has a constant-query proximity-oblivious tester if and only if $\mathcal{P}$ is GSF-local and non-propagating.*

The following open question was raised in [13].

▶ **Open Question 7** (Are all GSF-local properties non-propagating?). *Is it the case that for every GSF-local property $\mathcal{P} = \cup_{n\in\mathbb{N}}\mathcal{P}_n$, there is a sequence $\overline{\mathcal{F}} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ that is non-propagating and $\mathcal{P}$ is $\overline{\mathcal{F}}$-local?*

3 Relating different notions of locality

In this section we define properties by prescribing upper and lower bounds on the number of occurrence of neighbourhood types. These bounds are given by *neighbourhood profiles* which we will define formally below. We use these properties to give a natural characterization of FO properties of bounded-degree structures in Lemma 9, which is a straightforward consequence of Hanf's Theorem (Theorem 2). We use this characterization to establish links between FO definability and GSF-locality. This connection is the key ingredient in the proof of our main theorem.

Observe that for fixed $r, d \in \mathbb{N}$ and σ, there are only finitely many r-types in structures in $\mathcal{C}_{\sigma,d}$. For any signature σ and $d, r \in \mathbb{N}$ we let $n_{d,r,\sigma} \in \mathbb{N}$ be the number of different r-types of σ-structures of degree at most d. Assuming that for all $d, r \in \mathbb{N}$ the r-neighbourhood-types of σ-structures of degree at most d are ordered, we let $\tau^i_{d,r,\sigma}$ denote the i-th such neighbourhood type, for $i \in \{1, \dots, n_{d,r,\sigma}\}$. With each σ-structure $A \in \mathcal{C}_{\sigma,d}$ we associate its *r-histogram vector* $\overline{v}_{d,r,\sigma}(A)$, given by

$$(\overline{v}_{d,r,\sigma}(A))_i := |\{a \in U(A) \mid \mathcal{N}^A_r(a) \in \tau^i_{d,r,\sigma}\}|.$$

We let

$$\mathfrak{I} := \{[k,l], [k,\infty) \mid k \leq l \in \mathbb{N}\}$$

be the set of all closed or half-closed, infinite intervals with natural lower/upper bounds.

▶ **Definition 8.** *Let σ be a signature and $d, r \in \mathbb{N}$.*

1. *An* r-neighbourhood profile *of degree d is a function $\rho : \{1, \dots, n_{d,r,\sigma}\} \to \mathfrak{I}$.*
2. *For a structure $A \in \mathcal{C}_{\sigma,d}$, we say A obeys ρ, denoted by $A \sim \rho$, if*

   $$(\overline{v}_{d,r,\sigma}(A))_i \in \rho(i) \text{ for all } i \in \{1, \dots, n_{d,r,\sigma}\}.$$

   Let $\mathcal{P}_\rho$ be the set of structures A that obey ρ, i.e., $\mathcal{P}_\rho = \{A \in \mathcal{C}_{\sigma,d} \mid A \sim \rho\}$.
3. *We say that a property $\mathcal{P}$ is* defined by a finite union of neighbourhood profiles *if there is $k \in \mathbb{N}$ such that $\mathcal{P} = \bigcup_{1 \leq i \leq k} \mathcal{P}_{\rho_i}$ where ρ_i is an r_i-neighbourhood profile and $r_i \in \mathbb{N}$ for every $i \in \{1, \dots, k\}$.*

We let $n_{d,r} := n_{d,r,\sigma_{\text{graph}}}$ denote the total number of r-type of undirected graphs of degree at most d, and let $\tau^i_{d,r} := \tau^i_{d,r,\sigma_{\text{graph}}}$ be the i-th r-type of bounded degree d, for any $i \in \{1, \dots, n_{d,r}\}$. Further, for a graph G let $\overline{v}_{d,r}(G)$ denote the r-histogram vector of G. Note that for any type $\tau^i_{d,r}$ where the edge relation is not symmetric we have that $(\overline{v}_{d,r}(G))_i = 0$ and therefore in any r-neighbourhood profile ρ for graphs we have $\rho(i) = [0,0]$ for any type $\tau^i_{d,r}$ which is not symmetric.

We now give a lemma showing that bounded-degree FO properties can be equivalently defined as finite unions of properties defined by neighbourhood profiles. Here the technicalities that arise are due to Hanf normal form not requiring the locality-radius of all Hanf-sentences to be the same. The proof of Lemma 9 is deferred to Appendix C.

▶ **Lemma 9.** *For every non-empty property $\mathcal{P} \subseteq \mathcal{C}_{\sigma,d}$, $\mathcal{P}$ is FO definable on $\mathcal{C}_{\sigma,d}$ if and only if $\mathcal{P}$ can be obtained as a finite union of properties defined by neighbourhood profiles.*

3.1 Relating FO properties to GSF-local properties

We now prove that FO properties which arise as unions of neighbourhood profiles of a particularly simple form are GSF-local. For this let

$$\mathfrak{I}_0 := \{[0,\infty), [0,k] \mid k \in \mathbb{N}\} \subset \mathfrak{I}.$$

We call any neighbourhood profile ρ with codomain $\mathfrak{I}_0$ a 0-*profile*, as all lower bounds for the occurrence of types are 0.

▶ **Observation 10.** *Let ρ be a 0-profile. If two structures $A, A' \in \mathcal{C}_{\sigma,d}$ satisfy $(\overline{v}_{d,r,\sigma}(A))_i \leq (\overline{v}_{d,r,\sigma}(A'))_i$ for every $i \in \{1,\dots,n_{d,r,\sigma}\}$ and $A' \sim \rho$, then $A \sim \rho$.*
In particular, the existence of an r-type cannot be expressed by a 0-profile.

▶ **Theorem 11.** *Every finite union of properties defined by 0-profiles is GSF-local.*

Proof. We prove this in two parts (Claim 12 and Claim 13). We first argue that every property $\mathcal{P}_\rho$ defined by some 0-profile $\rho : \{1,\dots,n_{d,r,\sigma}\} \to \mathfrak{I}_0$ is GSF-local. For this it is important to note that we can express a forbidden r-type τ by a forbidden generalised subgraph. For $(B,b) \in \tau$, the set of all graphs with no vertex of neighbourhood type τ is the set of all B-free graphs where every vertex in $V(B)$ of distance less than r to b is marked "full" and every vertex in $V(B)$ of distance r to b is marked "semifull". Since a profile of the form $\rho : \{1,\dots,n_{d,r,\sigma}\} \to \mathfrak{I}_0$ can express that some neighbourhood type τ can appear at most k times for some fixed $k \in \mathbb{N}$, we need to forbid all marked graphs in which type τ appears $k+1$ times. We will formalise this in the following claim.

▷ Claim 12. For every r-neighbourhood profile $\rho : \{1,\dots,n_{d,r}\} \to \mathfrak{I}_0$, there is a finite set $\mathcal{F}$ of marked graphs such that $\mathcal{P}_\rho$ is exactly the property of $\mathcal{F}$-free graphs.

Proof. Assume τ is an r-type and $k \in \mathbb{N}_{>0}$. Then we say that a marked graph F is a *k-realisation* of τ if F has the following properties.

1. There are k distinct vertices $v_1,\dots,v_k$ in F such that $(\mathcal{N}_r^F(v_i), v_i) \in \tau$ for every $i = 1,\dots,k$.
2. Every vertex v in F has distance less or equal to r to at least one vertex v_i.
3. Every vertex v in F of distance less than r to at least one v_i is marked as "full".
4. Every vertex v in F of distance greater or equal to r to every v_i is marked as "semifull".

We denote by $S^k(\tau)$ the set of all k-realisations of τ.

Now we can define the set $\mathcal{F}$ of forbidden subgraphs to be

$$\mathcal{F} := \bigcup_{k\in\mathbb{N}, 1\leq i\leq n_{d,r,\sigma}:\rho(i)=[0,k]} S^{k+1}(\tau_{d,r}^i).$$

Let $\mathcal{P}$ be the property of all $\mathcal{F}$-free graphs. We first prove that the property $\mathcal{P}$ is contained in $\mathcal{P}_\rho$. Towards a contradiction assume that $G \in \mathcal{C}_d$ is $\mathcal{F}$-free but not contained in $\mathcal{P}_\rho$. As G is not contained in $\mathcal{P}_\rho$ there must be an index $i \in \{1,\dots,n_{d,r}\}$ such that $(\overline{v}_{d,r}(G))_i \notin \rho(i)$. Since $\rho(i) \in \mathfrak{I}_0$ there is $k \in \mathbb{N}$ such that $\rho(i) = [0,k]$ and hence $(\overline{v}_{d,r}(G))_i > k$. Hence there must be $k+1$ vertices $v_1,\dots,v_{k+1}$ in G such that $(\mathcal{N}_r^G(v_i), v_i) \in \tau_{d,r}^i$. We define the marked graph F to be the subgraph of G induced by the r-neighbourhoods of $v_1,\dots,v_{k+1}$, i. e. $G[\cup_{1\leq i\leq k+1} N_r^G(v_i)]$, in which every vertex of distance less than k to at least one of the v_i is marked as "full" and every other vertex is marked as "semifull". Then F is by definition a $(k+1)$-realisation of $\tau_{d,r}^i$ and hence $F \in \mathcal{F}$. We now argue that F can be embedded into G. Since F is an induced subgraph of G the identity map gives us a natural embedding $f : F \to G$. Let v be any vertex marked "full" in F. Then by construction of F, there is $i \in \{1, \dots, k+1\}$ such that $f(v)$ is of distance less than r to v_i in G. But then $N_1^G(f(v))$ is a subset of $N_r^G(v_i)$. As F without the marking is the subgraph of G induced by $\cup_{1 \leq i \leq k+1} N_r^G(v_i)$ this implies that $f(N_1^F(v)) = N_1^G(f(v))$. Furthermore, assume v is a vertex marked "semifull" in F. Then $f(N_1^F(v)) = N_1^G(f(v)) \cap f(V(F))$ holds as F without the markings is an induced subgraph of G. This proves that G is not F-free by Definition 3. This is a contradiction to our assumption that G is $\mathcal{F}$-free and $F \in \mathcal{F}$.

Similarly, we can show that $\mathcal{P}_\rho \subseteq \mathcal{P}$ by assuming $G \in \mathcal{C}_d$ is in $\mathcal{P}_\rho$ but not $\mathcal{F}$-free, and showing that the embedding of any graph of $\mathcal{F}$ into G yields an amount of vertices of a certain type contradicting containment in $\mathcal{P}_\rho$. ◁

Next we prove that classes defined by excluding finitely many marked graphs are closed under finite unions.

▷ Claim 13. Let $\mathcal{F}_1, \mathcal{F}_2$ be two finite sets of marked graphs. For $i \in \{1, 2\}$, let $\mathcal{P}_i$ be the property of $\mathcal{F}_i$-free graphs. Then there is a set $\mathcal{F}$ of generalised subgraphs such that $\mathcal{P}_1 \cup \mathcal{P}_2$ is the property of $\mathcal{F}$-free graphs.

Proof. We say that a marked graph F is a (not necessarily disjoint) union of marked graphs F_1, F_2 if

1. there is an embedding f_i of F_i into the graph F without its markings as in Definition 3 for every $i \in \{1, 2\}$.
2. for every vertex v in F there is $i \in \{1, 2\}$ and a vertex w in F_i such that $f_i(w) = v$.
3. every vertex v in F is marked "full", if there is $i \in \{1, 2\}$ and a "full" vertex w in F_i such that $f_i(w) = v$.
4. every vertex v in F is marked "semifull", if there is $i \in \{1, 2\}$ and a "semifull" vertex w in F_i such that $f_i(w) = v$ and $f_i(u) \neq v$ for every $i \in \{1, 2\}$ and every "full" vertex u.
5. every vertex v in F is marked "partial" if $f_i(u) \neq v$ for every $i \in \{1, 2\}$ and every "full" or "semifull" vertex u.

We define $S(F_1, F_2)$ to be the set of all possible (not necessarily disjoint) unions of F_1, F_2. We can now define the set $\mathcal{F}$ to be

$$\mathcal{F} := \bigcup_{F_1 \in \mathcal{F}_1, F_2 \in \mathcal{F}_2} S(F_1, F_2).$$

Let $\mathcal{P}$ be the property of all $\mathcal{F}$-free graphs. Now we prove $\mathcal{P} \subseteq \mathcal{P}_1 \cup \mathcal{P}_2$. Towards a contradiction assume G is $\mathcal{F}$-free but G is in neither $\mathcal{P}_1$ nor in $\mathcal{P}_2$. Then for every $i \in \{1, 2\}$ there is a graph $F_i \in \mathcal{F}_i$ such that G is not F_i-free. It is easy to see that there is a union $F_\cup$ of F_1 and F_2 such that G is not $F_\cup$-free, which contradicts that G is $\mathcal{F}$-free.

Conversely, in order to prove $\mathcal{P}_1 \cup \mathcal{P}_2 \subseteq \mathcal{P}$, if G is $\mathcal{F}_i$ free for some $i \in \{1, 2\}$ then G must be $\mathcal{F}$-free by construction of $\mathcal{F}$. ◁

Combining the two claims above proves the Theorem 11. ◀

Further discussion of the relation between FO and GSF-locality

First let us remark that it is neither true that every FO definable property is GSF-local, nor that every GSF-local property is FO definable.

▶ **Example 14.** The property of bounded-degree graphs containing a triangle is FO definable but not GSF-local.

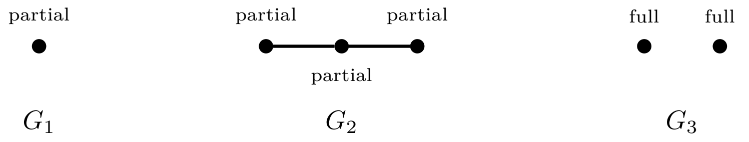


Figure 1 Marked graphs for Example 16.

Indeed, the existence of a fixed number of vertices of certain neighbourhood types can be expressed in FO, while in general, this cannot be expressed by forbidding generalised subgraphs. If a formula has a 0-profile (and hence does not require the existence of any types) then the property defined by that formula is GSF-local, as shown in Theorem 11.

▶ **Example 15.** The class of all bounded-degree graphs with an even number of vertices is GSF-local but not FO definable.

Let us remark that Theorem 11 combined with Lemma 9 proves that every finite union of properties definable by 0-profiles is both FO definable and GSF-local. Hence it is natural to ask whether the intersection of FO definable properties and GSF-local properties is precisely the set of finite unions of properties definable by 0-profiles. However, this is not the case. The following example shows that there are properties which are both FO definable and GSF-local but cannot be expressed by 0-profiles.

▶ **Example 16.** We let $d \geq 2$ and let $B_1 := (\{v\}, \{\})$, $B_2 = (\{v, w\}, \{\{v, w\}\})$ be two graphs. We further let τ_1, τ_2 be the 1-types of degree d such that $(B_1, v) \in \tau_1$ and $(B_2, v) \in \tau_2$. Consider the property $\mathcal{P}$ defined by the following FO formula

$$\varphi := \neg\exists x(x = x) \vee \exists^{=1} x\big(\varphi_{\tau_1}(x) \wedge \forall y(x \neq y \to \varphi_{\tau_2}(y))\big).$$

$\mathcal{P}$ contains, besides the empty graph, unions of an arbitrary amount of disjoint edges and one isolated vertex. To define a sequence of forbidden subgraphs we let G_1, G_2, G_3 be the marked graphs in Figure 1. Let $\mathcal{F}_{\text{even}} := \{G_1\}$ and $\mathcal{F}_{\text{odd}} := \{G_2, G_3\}$ and let $\overline{\mathcal{F}} = (\mathcal{F}_n)_{n\in\mathbb{N}}$ where $\mathcal{F}_i = \mathcal{F}_{\text{even}}$ if i is even and $\mathcal{F}_i = \mathcal{F}_{\text{odd}}$ if i is odd. Note that every graph on more than one vertex with an odd number of vertices which is $\mathcal{F}_{\text{odd}}$-free must contain a vertex of neighbourhood type τ_1, and that the set of $\mathcal{F}_{\text{even}}$-free graphs contains only the empty graph. Hence $\mathcal{P}$ is $\overline{\mathcal{F}}$-local. Now assume towards a contradiction that $\mathcal{P} = \cup_{1\leq i\leq k}\mathcal{P}_{\rho_i}$ for 0-profiles ρ_i. Let G_m be the graph consisting of m disjoint edges and one isolated vertex and H_m the graph consisting of m disjoint edges. Since $G_m \in \mathcal{P}$ there is $i \in \{1, \ldots, k\}$ such that $G_m \sim \rho_i$. By choice of G_m and H_m we have $0 \leq (\overline{v}_{d,r}(H_m))_j \leq (\overline{v}_{d,r}(G_m))_j \in \rho_i(j)$ for every $j \in \{1, \ldots, n_{d,r}\}$. Since additionally $\rho_i(j) \in \mathfrak{I}_0$ this implies that $(\overline{v}_{d,r}(H_m))_j \in \rho_i(j)$. But then $H_m \sim \rho_i$ which yields a contradiction as $H_m \notin \mathcal{P}$. Hence $\mathcal{P}$ can not be defined as a finite union of 0-profiles.

Figure 2 gives a schematic overview of all classes of properties discussed here and their relationship.

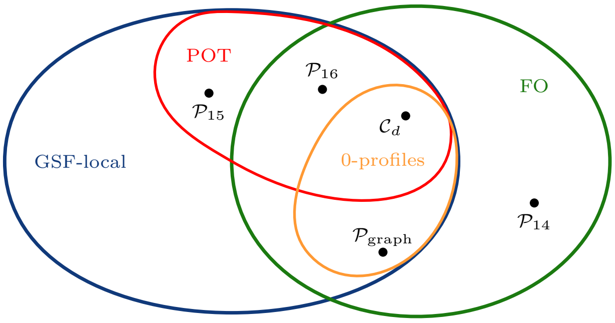


Figure 2 Overview of the classes of properties, here $\mathcal{P}_i$ refers to the property from Example i, $\mathcal{C}_d$ refers to the property of all graphs of bounded degree d and $\mathcal{P}_{\text{graph}}$ is the property defined in Section 4.2.

4 Proof of the main theorem

In this section we prove Theorem 1. We start by describing a property of relational structures, similar to a property in [1], which is not testable. We then show that the property can be expressed by a union of 0-profiles, and hence by Theorem 11 it is GSF-local.

Let σ be the signature, $d \in \mathbb{N}$ and $\mathcal{P}_{\textcircled{z}}$ be the property of d σ-structures of bounded-degree from [1].

Brief Description of the property $\mathcal{P}_{\textcircled{z}}$

$\mathcal{P}_{\textcircled{z}}$ is the property of all bounded-degree d σ-structures, which satisfy some first-order logic formula $\varphi_{\textcircled{z}}$. On a high level, each structure A in the property $\mathcal{P}_{\textcircled{z}}$ is a hybridization of a sequence of expander graphs and a tree structure, where the expander graphs are constructed by the zig-zag product that was introduced in [23]. Slightly more precisely, each model of $\varphi_{\textcircled{z}}$ is a rooted k-ary complete tree for some constant k, where the vertices on each level form an expander. In terms of logic language, for some constant $D > 1$, we considered

$$\sigma := \{\{E_{i,j}\}_{i,j\in[D]^2}, \{F_k\}_{k\in([D]^2)^2}, R, \{L_k\}_{k\in([D]^2)^2}\},$$

where $E_{i,j}$, F_k, R and L_k are binary relation symbols for $i, j \in [D]^2$ and $k \in ([D]^2)^2$. We further use F and E as an abbreviation to denote $\bigcup_{i,j\in[D]^2} E_{i,j}$ and $\bigcup_{k\in([D]^2)^2} F_k$. We defined an FO formula $\varphi_{\textcircled{z}}$ such that

$$\varphi_{\textcircled{z}} := \varphi_{\text{tree}} \wedge \varphi_{\text{rotationMap}} \wedge \varphi_{\text{base}} \wedge \varphi_{\text{recursion}}, \text{ and } \mathcal{P}_{\textcircled{z}} := \{\mathcal{A} \in \mathcal{C}_{\sigma,d} \mid \mathcal{A} \models \varphi_{\textcircled{z}}\},$$

where $\varphi_{\text{tree}}, \varphi_{\text{rotationMap}}, \varphi_{\text{base}}, \varphi_{\text{recursion}}$ are FO formulas which encode the tree structure (and degree regularity), rotation maps, base graph (with constant size) and recursive construction of expander graphs (via the zig-zag product). Note that for the construction we use some base graph H which is given by its rotation map $\text{ROT}_H : ([D]^2)^2 \times [D] \to ([D]^2)^2 \times [D]$, which is a special type of an encoding of a graph.

The precise formula is given in Appendix B. We will restate parts of the formula, whenever they are relevant in the proofs below.

4.1 Characterisation by neighbourhood profiles

Our aim in this section is to prove that a minor variation of property $\mathcal{P}_{\textcircled{Z}}$ of relational structures can be written as a finite union of properties defined by 0-profiles of radius 2. As the existence of a certain vertex cannot be expressed with a 0-profile (see Observation 10) and $\varphi_{\textcircled{Z}}$ demands the existence of a certain vertex (the root vertex), the property $\mathcal{P}_{\textcircled{Z}}$ cannot be expressed in terms of 0-profiles. However we define a slight variation of the formula $\varphi_{\textcircled{Z}}$ which, as we will see later, can be expressed by 0-profiles. Let

$$\varphi'_{\textcircled{Z}} := \varphi'_{\text{tree}} \wedge \varphi_{\text{rotationMap}} \wedge \varphi_{\text{base}} \wedge \varphi_{\text{recursion}},$$

where we obtain φ'_{tree} from φ_{tree} by replacing the subformula $\exists^{=1}x\varphi_{\text{root}}(x)$ by $\exists^{\leq 1}x\varphi_{\text{root}}(x)$, where $\varphi_{\text{root}}(x) := \forall y \neg F(y,x)$. We define the property

$$\mathcal{P}'_{\textcircled{Z}} := \{A \in \mathcal{C}_{\sigma,d} \mid A \models \varphi'_{\textcircled{Z}}\}.$$

We denote the empty structure by $A_\emptyset$ (i. e. $U(A_\emptyset) = \emptyset$).

▶ **Lemma 17.** *The properties $\mathcal{P}'_{\textcircled{Z}}$ and $\mathcal{P}_{\textcircled{Z}} \cup \{A_\emptyset\}$ are equal.*

To prove this we use the following lemma [1, Lemma 3.5].

▶ **Lemma 18** ([1])**.** *For $A \in \mathcal{C}_{\sigma,d}$ let G_F^A be the graph with vertex set $U(A)$ and edge set $\{\{a,b\} \mid (a,b) \in F(A)\}$. If $A \models \varphi_{\textcircled{Z}}$ then G_F^A is connected.*

Proof of Lemma 17. We fist prove that $\mathcal{P}'_{\textcircled{Z}} \subseteq \mathcal{P}_{\textcircled{Z}} \cup \{A_\emptyset\}$. Consider the formula $\tilde{\varphi}_{\textcircled{Z}}$ which is obtained from $\varphi_{\textcircled{Z}}$ by removing the subformula $\exists^{=1}x\varphi_{\text{root}}(x)$. We use the following simple observation, which we will prove in Appendix D.

▷ Claim 19. Satisfying $\tilde{\varphi}_{\textcircled{Z}}$ is closed under disjoint unions on $\mathcal{C}_{\sigma,d}$.

Since $A_\emptyset \in \mathcal{P}_{\textcircled{Z}} \cup \{A_\emptyset\}$ it is sufficient to consider only non-empty structures in the following. Therefore assume that there exists $A \in \mathcal{C}_{\sigma,d}$ with $U(A) \neq \emptyset$ such that $A \models \varphi'_{\textcircled{Z}}$ and A contains no element a for which $A \models \varphi_{\text{root}}(a)$. Let $A' \in \mathcal{C}_{\sigma,d}$ be any model of $\varphi_{\textcircled{Z}}$ with $U(A) \cap U(A') = \emptyset$. Then $A \cup A' \models \tilde{\varphi}_{\textcircled{Z}}$ by Claim 19. Furthermore, $A \cup A' \models \exists^{=1}x\varphi_{\text{root}}(x)$, which implies $A \cup A' \models \varphi_{\textcircled{Z}}$. By construction $G_F^{A\cup A'}$ has more than one connected component as both $U(A) \neq \emptyset$ and $U(A') \neq \emptyset$ and $A \cup A'$ is a disjoint union of A and A'. Hence we obtain a contradiction to Lemma 18. Therefore every non-empty structure satisfying $\varphi'_{\textcircled{Z}}$ must satisfy $\exists^{=1}x\varphi_{\text{root}}(x)$, and hence also $\varphi_{\textcircled{Z}}$.

Conversely, if $A \in \mathcal{C}_{\sigma,d}$ is a model of $\varphi_{\textcircled{Z}}$ then $A \models \exists^{=1}x\varphi_{\text{root}}(x)$. This implies directly that $A \models \exists^{\leq 1}x\varphi_{\text{root}}(x)$ and hence $A \models \varphi'_{\textcircled{Z}}$. Furthermore, $A_\emptyset \in \mathcal{P}'_{\textcircled{Z}}$ as $A \models \exists^{\leq 1}x\varphi_{\text{root}}(x)$ and $A \models \tilde{\varphi}_{\textcircled{Z}}$ as $\tilde{\varphi}_{\textcircled{Z}}$ is a conjunction of universally quantified formulas. Hence $\mathcal{P}_{\textcircled{Z}} \cup \{A_\emptyset\} \subseteq \mathcal{P}'_{\textcircled{Z}}$. ◀

We now define the 0-profiles which express the property $\mathcal{P}'_{\textcircled{Z}}$. For all σ-structures in $\mathcal{P}_{\textcircled{Z}}$ (all σ-structure in $\mathcal{P}'_{\textcircled{Z}}$ but $A_\emptyset$) it is crucial that they are allowed to contain precisely one root element. Hence the neighbourhood profile describing $\mathcal{P}'_{\textcircled{Z}}$ must restrict the number of occurrences of the 2-type of the root element. But since in $\mathcal{P}_{\textcircled{Z}}$, the root elements in different structures may have different 2-types, we partition $\mathcal{P}_{\textcircled{Z}}$ into parts $\mathcal{P}_1, \ldots, \mathcal{P}_m$ by the 2-type of the root element. Note that the number m of parts is constant as there are at most $n_{d,2,\sigma}$ 2-types in total. For each of these parts we then define a neighbourhood profile ρ_k such that $\mathcal{P}_k \cup \{A_\emptyset\} = \mathcal{P}_{\rho_k}$. We would like to remark here that the roots of all but one structure in $\mathcal{P}_{\textcircled{Z}}$ actually have the same 2-types. However, proving this requires a detailed insight into the construction of $\mathcal{P}_{\textcircled{Z}}$, so we avoid this here and use the partition into finitely many parts instead. We now define the parts and corresponding profiles formally.

Assume without loss of generality that the 2-types $\tau^1_{d,2,\sigma}, \dots, \tau^{n_{d,2,\sigma}}_{d,2,\sigma}$ of degree d are ordered in such a way that for $(B, b) \in \tau^k_{d,2,\sigma}$, it holds that $B \models \varphi_{\text{root}}(b)$ if and only if $k \in \{1, \dots, m\}$ for some $m \leq n_{d,2,\sigma}$. For $k \in \{1, \dots, m\}$, let

$$\mathcal{P}_k := \{A \in \mathcal{P}_{\textcircled{Z}} \mid \text{ there is } a \in U(A) \text{ such that } (\mathcal{N}^A_2(a), a) \in \tau^k_{d,2,\sigma}\}.$$

Since every $A \in \mathcal{P}_{\textcircled{Z}}$ satisfies $\exists^{=1} x \varphi_{\text{root}}(x)$ we get that

$$\mathcal{P}'_{\textcircled{Z}} = \bigcup_{1 \leq k \leq m} \mathcal{P}_k \cup \{A_\emptyset\}$$

and this union is disjoint. Furthermore, for $k \in \{1, \dots, m\}$, let $I_k \subseteq \{1, \dots, n_{d,2,\sigma}\}$ be the set of indices j such that there is a structure $A \in \mathcal{P}_k$ and $a \in U(A)$ with $(\mathcal{N}^A_2(a), a) \in \tau^j_{d,2,\sigma}$. For every $k \in \{1, \dots, m\}$ we define the 2-neighbourhood profile $\rho_k : \{1, \dots, n_{d,2,\sigma}\} \to \mathfrak{I}_0$ by

$$\rho_k(i) := \begin{cases} [0,1] & \text{if } i = k, \\ [0,\infty) & \text{if } i \in I_k \setminus \{k\}, \\ [0,0] & \text{otherwise.} \end{cases}$$

To prove that these 0-profiles of radius 2 define the property $\mathcal{P}'_{\textcircled{Z}}$, the crucial observation is that for every element a of some structure in $\mathcal{C}_{\sigma,d}$, the FO-formula $\varphi'_{\textcircled{Z}}$ only talks about elements of distance at most 2 to a (i. e. $\varphi'_{\textcircled{Z}}$ is 2-local). Hence the 2-histogram vector of a structure already captures whether the structure satisfies $\varphi'_{\textcircled{Z}}$. We will now formally prove this.

▶ **Lemma 20.** *It holds that $\mathcal{P}'_{\textcircled{Z}} = \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k}$.*

Proof. We first prove that $\mathcal{P}'_{\textcircled{Z}} \subseteq \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k}$. First note that trivially $A_\emptyset \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k}$. Now assume $A \in \mathcal{P}_{\textcircled{Z}}$. This implies that there is $k \in \{1, \dots, m\}$ such that $A \in \mathcal{P}_k$. By construction we have that for every $a \in A$, there is $i \in I_k$ such that $(\mathcal{N}^A_2(a), a) \in \tau^i_{d,2,\sigma}$. Furthermore, since $A \models \varphi_{\textcircled{Z}}$, we have that $A \models \exists^{=1} x \varphi_{\text{root}}(x)$, and that there can be at most one $a \in U(A)$ such that $(\mathcal{N}^A_2(a), a) \in \tau^k_{d,2,\sigma}$. Therefore $A \in \mathcal{P}_{\rho_k}$.

To prove $\bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \subseteq \mathcal{P}'_{\textcircled{Z}}$, we prove that every structure in $\bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k}$ must satisfy $\varphi'_{\textcircled{Z}}$. We will prove that every $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k}$ satisfies $\varphi_{\text{recursion}}$, and refer for the proof that A satisfies $\varphi'_{\text{tree}} \wedge \varphi_{\text{rotationMap}} \wedge \varphi_{\text{base}}$ to Claim 30, Claim 31 and Claim 32 in Appendix D. Note that $A_\emptyset \models \varphi'_{\textcircled{Z}}$ by Lemma 17 and hence we exclude $A_\emptyset$ in the following.

▷ Claim 21. Every structure $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$ satisfies $\varphi_{\text{recursion}}$.

Proof. Let $A \in \bigcup_{1 \le k \le m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$. Then there is a $k \in \{1, \dots, m\}$ such that $A \in \mathcal{P}_{\rho_k}$.

By definition, $\varphi_{\text{recursion}} := \forall x \forall z \big(\varphi(x,z) \vee \psi(x,z)\big)$ (see Appendix B), where

$$\varphi(x,z) := \neg \exists y F(x,y) \wedge \neg \exists y F(z,y) \text{ and}$$

$$\psi(x,z) := \bigwedge_{\substack{k'_1, k'_2 \in [D]^2 \\ \ell'_1, \ell'_2 \in [D]^2}} \Bigg(\exists y \big[E_{k'_1,\ell'_1}(x,y) \wedge E_{k'_2,\ell'_2}(y,z) \big] \rightarrow \bigwedge_{\substack{i,j,i',j' \in [D], k,\ell \in ([D]^2)^2 \\ \mathrm{ROT}_H(k,i) = ((k'_1,k'_2),i') \\ \mathrm{ROT}_H((\ell'_2,\ell'_1),j) = (\ell,j')}} \exists x' \exists z' \big[F_k(x,x') \wedge F_\ell(z,z') \wedge E_{(i,j),(j',i')}(x',z') \big] \Bigg).$$

Let $a, c \in U(A)$. Assume first that there is $b \in U(A)$ with $(a,b) \in F(A)$. Hence $A \not\models \varphi(a,c)$. Since $\varphi_{\text{recursion}} := \forall x \forall z \big(\varphi(x,z) \vee \psi(x,z)\big)$ we aim to prove $A \models \psi(a,c)$. By construction of ρ_k, there is an $i \in I_k$ such that $(\mathcal{N}_2^A(a), a) \in \tau^i_{d,2,\sigma}$. Therefore there is a structure $\tilde{A} \models \varphi_{\textcircled{\tiny Z}}$ and $\tilde{a} \in U(\tilde{A})$ such that $(\mathcal{N}_2^A(a), a) \cong (\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Let f be an isomorphism from $(\mathcal{N}_2^A(a), a)$ to $(\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Since $b \in N_2^A(a)$, we get that $f(b)$ is defined. Since f is an isomorphism mapping a onto $\tilde{a}$, we have that $(a,b) \in F(A)$ implies that $(\tilde{a}, f(b)) \in F(\tilde{A})$. Hence $\tilde{A} \not\models \varphi(\tilde{a}, \tilde{c})$, for every $\tilde{c} \in U(\tilde{A})$. But since $\tilde{A} \models \varphi_{\text{recursion}}$, as $\tilde{A} \models \varphi_{\textcircled{\tiny Z}}$, this shows that $\tilde{A} \models \psi(\tilde{a}, \tilde{c})$ for every $\tilde{c} \in U(\tilde{A})$.

Let $k'_1, k'_2 \in [D]^2$ and $\ell'_1, \ell'_2 \in [D]^2$ be indices such that there is $b' \in U(A)$ with $(a,b') \in E_{k'_1,\ell'_1}(A)$ and $(b',c) \in E_{k'_2,\ell'_2}(A)$. Since $b', c \in N_2^A(a)$, by assumption we get that $f(b')$ and $f(c)$ are defined. Furthermore, $(a,b') \in E_{k'_1,\ell'_1}(A)$ and $(b',c) \in E_{k'_2,\ell'_2}(A)$ imply that $(\tilde{a}, f(b')) \in E_{k'_1,\ell'_1}(\tilde{A})$ and $(f(b'), f(c)) \in E_{k'_2,\ell'_2}(\tilde{A})$, since f is an isomorphism mapping a onto $\tilde{a}$. We proved in the previous paragraph that $\tilde{A} \models \psi(\tilde{a}, f(c))$. Hence we can conclude that for all indices $i, j, i', j' \in [D]$, $k, \ell \in ([D]^2)^2$ for which $\mathrm{ROT}_H(k,i) = ((k'_1,k'_2),i')$ and $\mathrm{ROT}_H((\ell'_2,\ell'_1),j) = (\ell,j')$, there are elements $\tilde{a}', \tilde{c}' \in U(\tilde{A})$ such that $(\tilde{a}, \tilde{a}') \in F_k(\tilde{A})$, $(f(c), \tilde{c}') \in F_\ell(\tilde{A})$, and $(\tilde{a}', \tilde{c}') \in E_{(i,j),(j',i')}(\tilde{A})$. Since $\tilde{a}', \tilde{c}' \in N_2^{\tilde{A}}(\tilde{a})$, we get that $a' := f^{-1}(\tilde{a}')$ and $c' := f^{-1}(\tilde{c}')$ are defined. Furthermore, we get that $(a,a') \in F_k(A)$, $(c,c') \in F_\ell(A)$ and $(a',c') \in E_{(i,j),(j',i')}(A)$. This proves that $A \models \psi(a,c)$.

In the case that there is $b \in U(A)$ with $(c,b) \in F(A)$, we can prove similarly that $A \models \psi(a,c)$, by considering that there exist $\tilde{A} \models \varphi_{\textcircled{\tiny Z}}$ and $\tilde{c} \in U(\tilde{A})$ such that $(\mathcal{N}_2^A(a), c) \cong (\mathcal{N}_2^{\tilde{A}}(\tilde{c}), \tilde{c})$ by construction of ρ_k. Finally if there is no $b \in U(A)$ such that $(a,b) \in F(A)$ or $(c,b) \in F(A)$ then $A \models \varphi(a,c)$. Since this covers every case we get that $A \models \varphi_{\text{recursion}}$. ◁

Assume $A \in \bigcup_{1 \le k \le m} \mathcal{P}_{\rho_k}$. As proved in Claims 30, 31, 32 and 21 this implies that $A \models \varphi'_{\text{tree}}$, $A \models \varphi_{\text{rotationMap}}$, $A \models \varphi_{\text{base}}$ and $A \models \varphi_{\text{recursion}}$. Since $\varphi'_{\textcircled{\tiny Z}}$ is a conjunction of these formulas, we get $A \models \varphi'_{\textcircled{\tiny Z}}$ and hence $A \in \mathcal{P}'_{\textcircled{\tiny Z}}$. ◀

4.2 A local reduction from relational structures to graphs

In this section we will define our graph property $\mathcal{P}_{\text{graph}}$ by giving a reduction from the property $\mathcal{P}'_{\textcircled{\tiny Z}}$ and argue that $\mathcal{P}_{\text{graph}}$ is GSF-local while not testable. To do so, we show that this reduction is "local" which preserves the testability of these two properties.

Local reduction

We first introduce the following notion of local reduction between two property testing models. In the following, when the context is clear, we will use $\mathcal{C}$ to denote both a class of structure and the corresponding property testing model, which can be either the bounded-degree model for graphs or bounded-degree model for relational structures.

▶ **Definition 22** (Local reduction). *Let $\mathcal{C}, \mathcal{C}'$ be two property testing models and let $\mathcal{P} \subseteq \mathcal{C}$, $\mathcal{P}' \subseteq \mathcal{C}'$ be two properties. We say that a function $f : \mathcal{C} \to \mathcal{C}'$ is a local reduction from $\mathcal{P}$ to $\mathcal{P}'$ if there are constants $c_1, c_2 \in \mathbb{N}_{\geq 1}$ such that for every $X \in \mathcal{C}$ the following properties hold.*

1. *If $X \in \mathcal{P}$ then $f(X) \in \mathcal{P}'$.*
2. *If X is ϵ-far from $\mathcal{P}$ then $f(X)$ is (ϵ/c_1)-far from $\mathcal{P}'$.*
3. *For every query to $f(X)$ we can adaptively[2] compute c_2 queries such that the answer to the query to $f(X)$ can be computed from the answers to the c_2 queries to X.*

The following lemma is known.

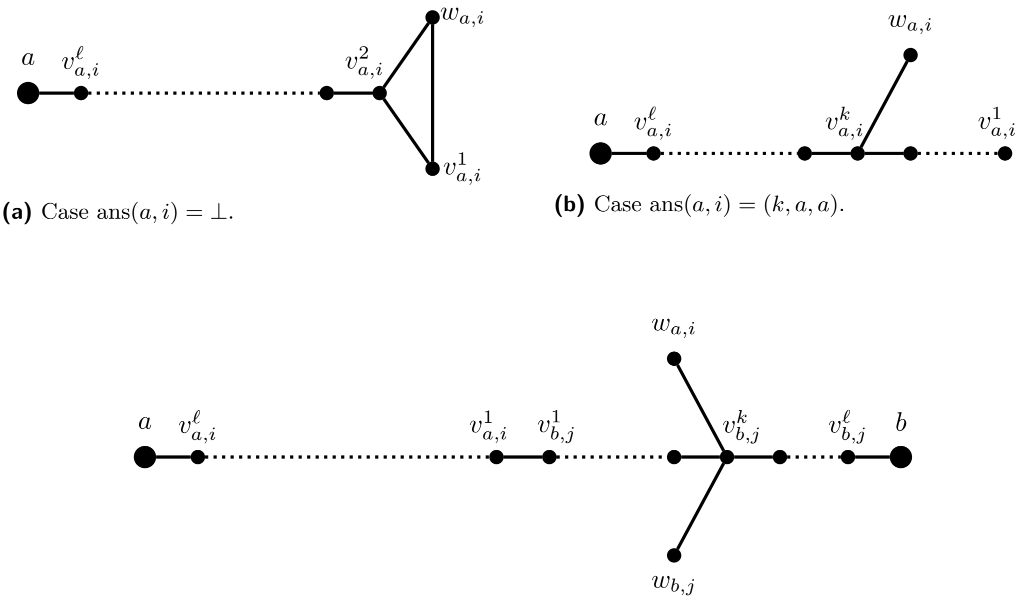
▶ **Lemma 23** (Theorem 7.14 in [9]). *Let $\mathcal{C}, \mathcal{C}'$ be two property testing models, $\mathcal{P} \subseteq \mathcal{C}$, $\mathcal{P}' \subseteq \mathcal{C}'$ be two properties and f a local reduction from $\mathcal{P}$ to $\mathcal{P}'$. If $\mathcal{P}'$ is testable then so is $\mathcal{P}$.*

Construction of the graph property

Now we construct a property $\mathcal{P}_{\text{graph}}$ from the property $\mathcal{P}'_{\textcircled{Z}}$. We obtain this graph property as $f(\mathcal{P}'_{\textcircled{Z}})$ by defining a map $f : \mathcal{C}_{\sigma,d} \to \mathcal{C}_d$. To define f we introduce a distinct arrow-graph gadget for every relation in σ (i. e. for every edge colour). The map f then replaces every tuple in a certain relation (every coloured edge) by the respective arrow-graph gadget. We further prove that this replacement operation defines a local reduction f from $\mathcal{P}'_{\textcircled{Z}}$ to $\mathcal{P}_{\text{graph}}$. Recall that a local reduction is a function maintaining distance that can be simulated locally by queries. Since by Lemma 23 local reductions preserve testability, we use the local reduction from $\mathcal{P}'_{\textcircled{Z}}$ to $\mathcal{P}_{\text{graph}}$ to obtain non-testability of the property $\mathcal{P}_{\text{graph}}$ from the non-testability of $\mathcal{P}'_{\textcircled{Z}}$. We will now define f formally.

Let ℓ be the number of relations (the number of edge colours) in σ. We first introduce the different types of arrow-graph gadgets we need to define the local reduction. For $1 \leq k \leq \ell$, we let H_k be the graph with vertex set $V(H_k) := \{a_1, \dots, a_{2\ell+2}, b_1, b_2\}$ and edge set $E(H_k) := \{\{a_i, a_{i+1}\} \mid 1 \leq i \leq 2\ell + 1\} \cup \{\{a_{\ell+1+k}, b_j\} \mid j \in \{1, 2\}\}$. We call H_k a *k-arrow.* For any graph G and vertices $v, w \in V(G)$, we say that there is a k-arrow from v to w, denoted $v \xrightarrow{k} w$, if there are $2\ell + 2$ vertices $v_2, \dots, v_{2\ell+1}, w_1, w_2 \in V(G)$ and an isomorphism $g : H_k \to \mathcal{N}_1^G(v_2, \dots, v_{2\ell+1}, w_1, w_2)$ such that $g(a_1) = v$ and $g(a_{2\ell+2}) = w$. We now define a second arrow gadget. For $1 \leq k \leq \ell$, we let L_k be the graph with vertex set $V(L_k) := \{a_1, \dots, a_{\ell+1}, b\}$ and edge set $E(L_k) := \{\{a_i, a_{i+1}\} \mid 1 \leq i \leq \ell\} \cup \{\{a_k, b\}\}$. We call L_k a *k-loop.* For any graph G and vertex $v \in V(G)$, we say that there is a k-loop at v, denoted $v \xrightarrow{k} v$, if there are $\ell + 1$ vertices $v_1, \dots, v_\ell, w \in V(G)$ and an isomorphism $g : L_k \to \mathcal{N}_1^G(v_1, \dots, v_\ell, w)$ such that $g(a_{\ell+1}) = v$. Finally we let $H_\perp$ be the graph with vertex set $V(H_\perp) := \{a_1, \dots, a_{\ell+1}, b\}$ and edge set $E(H_\perp) := \{\{a_i, a_{i+1}\} \mid 1 \leq i \leq \ell\} \cup \{\{a_i, b\} \mid i \in \{1, 2\}\}$. We call $H_\perp$ a *non-arrow.* For any graph G and vertex $v \in V(G)$, we say that there is a non-arrow at v, denoted $v \not\to$, if there are $\ell + 1$ vertices $v_1, \dots, v_\ell, w \in V(G)$ and an isomorphism $g : H_\perp \to N_1^G(v_1, \dots, v_\ell, w)$ such that $g(a_{\ell+1}) = v$.

[2] By adaptively computing queries we mean that the selection of the next query may depend on the answer to the previous query.



(a) Case $\text{ans}(a,i) = \bot$.

(b) Case $\text{ans}(a,i) = (k,a,a)$.

(c) Case $\text{ans}(a,i) = \text{ans}(b,j) = (k,a,b)$.

Figure 3 Different types of arrows in G_A.

We now define a function $f : \mathcal{C}_{\sigma,d} \to \mathcal{C}_d$ by $f(A) := G_A$, where G_A is the graph on vertex set $V(G_A) := U(A) \cup \{v^k_{a,i}, w_{a,i} \mid 1 \le i \le d, a \in U(A), 1 \le k \le \ell\}$ and edge set

$$\begin{aligned} E(G_A) :=& \Big\{\{a, v^{\ell}_{a,i}\} \mid a \in U(A), 1 \le i \le d\Big\} \\ &\cup \Big\{\{v^k_{a,i}, v^{k+1}_{a,i}\} \mid 1 \le k \le \ell - 1, a \in U(A), 1 \le i \le d\Big\} \\ &\cup \Big\{\{v^k_{b,j}, w_{b,j}\}, \{v^k_{b,j}, w_{a,i}\}, \{v^{\ell}_{a,i}, v^{\ell}_{b,j}\} \mid a \neq b, \text{ans}(a,i) = \text{ans}(b,j) = (k,a,b)\Big\} \\ &\cup \Big\{\{v^k_{a,i}, w_{a,i}\} \mid \text{ans}(a,i) = (k,a,a)\Big\} \cup \Big\{\{v^1_{a,i}, w_{a,i}\}, \{v^2_{a,i}, w_{a,i}\} \mid \text{ans}(a,i) = \bot\Big\}, \end{aligned}$$

where $\text{ans}(a,i) = (k,a,b)$ denotes that the i-th tuple of a is (a,b) and is in the k-th relation. Hence G_A is defined in such a way that if (a,b) is a tuple in the k-th relation of σ in A, then $a \xrightarrow{k} b$ in G_A, and a has a non-arrow for every i satisfying that $\text{ans}(a,i) = \bot$ for every k. For illustration see Figure 3.

Now we define property $\mathcal{P}_{\text{graph}} := \{f(A) \mid A \in \mathcal{P}'_{\textcircled{z}}\} \subseteq \mathcal{C}_d$.

▶ **Lemma 24.** *The map f is a local reduction from $\mathcal{P}'_{\textcircled{z}}$ to $\mathcal{P}_{\text{graph}}$.*

Proof. First note that for any $A \in \mathcal{P}'_{\textcircled{z}}$, we have that $f(A) \in \mathcal{P}_{\text{graph}}$ by definition.

Now let $c_1 = 2d + 2d^2\ell$. We prove that if $A \in \mathcal{C}_{\sigma,d}$ is ϵ-far from $\mathcal{P}'_{\textcircled{z}}$ then $f(A)$ is ϵ/c_1-far from $\mathcal{P}_{\text{graph}}$ by contraposition. Therefore assume that $f(A) =: G_A$ is not ϵ/c_1-far from $\mathcal{P}_{\text{graph}}$ for some $A \in \mathcal{C}_{\sigma,d}$. Then there is a set $E \subseteq \{e \subseteq V(G_A) \mid |e| = 2\}$ of size at most $\epsilon d|V(G_A)|/c_1$, and a graph $G \in \mathcal{P}_{\text{graph}}$ such that G is obtained from G_A by modifying the tuples in E. By definition of $\mathcal{P}_{\text{graph}}$, there is a structure $A_G \in \mathcal{P}'_{\textcircled{z}}$ such that $f(A_G) = G$. First note that $|U(A_G)| = |U(A)|$, as $(1 + d\ell)|U(A)| = |V(G_A)| = |V(G)| = (1 + d\ell)|U(A_G)|$. Hence there must be a set R of tuples that need to be modified to make A isomorphic to A_G. First note that R cannot contain a tuple (a,b) where $\{a, v^k_{a,i}, w_{a,i}, b, v^k_{b,i}, w_{b,i} \mid 1 \le i \le d, 1 \le$

$k \leq \ell\} \cap e = \emptyset$ for every $e \in E$. This is because if (a, b) is a tuple in A, then $a \xrightarrow{k} b$ for some k in G_A. But since $\{a, v^k_{a,i}, w_{a,i}, b, v^k_{b,i}, w_{b,i} \mid 1 \leq i \leq d, 1 \leq k \leq \ell\} \cap e = \emptyset$ for every $e \in E$, we have that $a \xrightarrow{k} b$ in G. But then (a, b) must be a tuple in A_G, and hence (a, b) cannot be in R. The same argument works when assuming that (a, b) is a tuple in A_G. Since for every $e \in E$, there are at most $2d$ tuples (a, b) such that $\{a, v^k_{a,i}, w_{a,i}, b, v^k_{b,i}, w_{b,i} \mid 1 \leq i \leq d, 1 \leq k \leq \ell\} \cap e \neq \emptyset$, we get that

$$|R| \leq 2d\epsilon d|V(G_A)|/c_1 = 2d(1 + d\ell)\epsilon d|U(A)|/c_1 = \epsilon d|U(A)|.$$

Hence A is not ϵ-far to being in $\mathcal{P}'_{\textcircled{Z}}$.

Let $c_2 := d+1$. Let $A \in \mathcal{C}_{\sigma,d}$ and $G_A := f(A)$. Note that any $a \in U(A)$ is adjacent in G_A to $v^\ell_{a,i}$, for every $1 \leq i \leq d$. Hence any neighbour query in G_A to a can be answered without querying A. Assume $v \in \{v^k_{a,i}, w_{a,i} \mid 1 \leq k \leq \ell\}$ for some $a \in U(A)$ and some $1 \leq i \leq d$. Then we can determine all neighbours of v by querying (a, i) and further if $\text{ans}(a, i) \neq \perp$ and $\text{ans}(a, i) = (k, a, b)$, then we need to query (b, j) for every $1 \leq j \leq d$. Hence we can determine the answer to any query to G_A by making c_2 queries to A. This proves that f is a local reduction from $\mathcal{P}'_{\textcircled{Z}}$ to $\mathcal{P}_{\text{graph}}$. ◀

We remark that $\mathcal{P}_{\text{graph}}$ is a simpler version of the simple graph property defined in [1] where extra care had to be taken to define degree-regular graphs.

4.3 The graph property is GSF-local

Let $\mathcal{P}_{\text{graph}}$ be the graph property as defined in Section 4.2. We now show that $\mathcal{P}_{\text{graph}}$ is GSF-local.

▶ **Lemma 25.** *The graph property $\mathcal{P}_{\text{graph}}$ is GSF-local.*

Proof. For this we will prove that $\mathcal{P}_{\text{graph}}$ is equal to a finite union of properties defined by 0-profiles, and then use Theorem 11 to prove that $\mathcal{P}_{\text{graph}}$ is GSF-local. We define the 0-profiles for $\mathcal{P}_{\text{graph}}$ in a very similar way to the relational structure case, and then use the description of $\mathcal{P}'_{\textcircled{Z}}$ by 0-profiles shown in Lemma 20. To this end, assume that the $4\ell+2$-types $\tau^1_{d,4\ell+2}, \ldots, \tau^{n_{d,4\ell+2}}_{d,4\ell+2}$ are ordered in such a way that $(\mathcal{N}^{f(B)}_{4\ell+2}(b), b) \in \tau^k_{d,4\ell+2}$, for every $k \in \{1, \ldots, m\}$ and $(B, b) \in \tau^k_{d,2,\sigma}$, where m is the number of parts of the partition of $\mathcal{P}_{\textcircled{Z}}$ defined in Subsection 4.1. For $k \in \{1, \ldots, m\}$, let $\hat{I}_k$ be the set of indices i such that there is $A \in \mathcal{P}_k$, and $v \in V(f(A))$ for which $(\mathcal{N}^{f(A)}_{4\ell+2}(v), v) \in \tau^i_{d,4\ell+2}$. Let $\hat{\rho_k} : \{1, \ldots, n_{d,4\ell+2}\} \to \mathfrak{I}_0$ be defined by

$$\hat{\rho_k}(i) := \begin{cases} [0,1] & \text{if } i = k, \\ [0,\infty) & \text{if } i \in \hat{I}_k \setminus \{k\}, \\ [0,0] & \text{otherwise.} \end{cases}$$

▷ Claim 26. It holds that $\mathcal{P}_{\text{graph}} = \bigcup_{1 \leq k \leq m} \mathcal{P}_{\hat{\rho}_k}$.

Proof. First we prove $\mathcal{P}_{\text{graph}} \subseteq \bigcup_{1 \leq k \leq m} \mathcal{P}_{\hat{\rho}_k}$. Assume $G \in \mathcal{P}_{\text{graph}}$ and let $A \in \mathcal{P}'_{\textcircled{Z}}$ be a structure such that $G = f(A)$. If $A = A_\emptyset$ then clearly $G \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\hat{\rho}_k}$. Hence assume $A \neq A_\emptyset$. Then $A \in \mathcal{P}_k$ for some $k \in \{1, \ldots, m\}$. By the construction of $\hat{I}_k$ we know that for every $v \in V(G)$ we have $(\mathcal{N}^G_{4\ell+2}(v), v) \in \tau^i_{d,4\ell+2}$ for some $i \in \hat{I}_k$. Furthermore, since $A \in \mathcal{P}_k$ there is at most one $a \in U(A)$ with $(\mathcal{N}^A_2(a), a) \in \tau^k_{d,2,\sigma}$. This implies directly that there can be at most one vertex $v \in V(G)$ with $(\mathcal{N}^G_{4\ell+2}(v), v) \in \tau^k_{d,4\ell+2}$ and hence $G \in \mathcal{P}_{\hat{\rho}}$.

Now we prove that $\bigcup_{1\le k\le m} \mathcal{P}_{\hat{\rho}_k} \subseteq \mathcal{P}_{\text{graph}}$. Let $G \in \bigcup_{1\le k\le m} \mathcal{P}_{\hat{\rho}_k}$ and let $k \in \{1, \dots, m\}$ be an index such that $G \in \mathcal{P}_{\hat{\rho}_k}$.

First note that every model of $\varphi_{\textcircled{Z}}$ is d regular for some large d. Then for any $A \models \varphi_{\textcircled{Z}}$, every vertex in $f(A)$ has either degree ≤ 4 or degree d . Since every structure in $\mathcal{P}'_{\textcircled{Z}}$ apart from the empty structure $A_\emptyset$ is a model of $\varphi_{\textcircled{Z}}$, this implies that every vertex in any graph $G' \in \mathcal{P}_{\text{graph}}$ has degree ≤ 4 or degree d. Since for every i for which $\hat{\rho}(i) \ne [0, 0]$, there is a graph $G' \in \mathcal{P}_{\text{graph}}$ and $v \in V(G')$ such that $(\mathcal{N}^{G'}_{4\ell+2}(v), v) \in \tau^i_{d,4\ell+2}$, we get that every vertex in G has to have degree ≤ 4 or degree d. Using this argument further, we get that every vertex $v \in V(G)$ of degree ≤ 4 has to be contained in the $(\ell + 1)$-neighbourhood of a vertex of degree d, and that the $(2\ell + 1)$-neighbourhood of every vertex $v \in V(G)$ of degree d is the union of k-arrows, k-loops and non-arrows which are disjoint apart from their endpoints. Hence there is a σ-structure A such that $f(A) \cong G$. Let g be an isomorphism from $f(A)$ to G.

Now we argue that $A \in \mathcal{P}_{\rho_k}$. First assume that there are two elements a, b with $(\mathcal{N}^A_2(a), a) \in \tau^k_{d,2,\sigma}$ and $(\mathcal{N}^A_2(b), b) \in \tau^k_{d,2,\sigma}$. By definition, we get that $(\mathcal{N}^{f(A)}_{4\ell+2}(a), a) \in \tau^k_{d,4\ell+2}$ and $(\mathcal{N}^{f(A)}_{4\ell+2}(b), b) \in \tau^k_{d,4\ell+2}$. Since g is an isomorphism, the restriction of g to $N^{f(A)}_{4\ell+2}(a)$ must be an isomorphism from $\mathcal{N}^{f(A)}_{4\ell+2}(a)$ to $\mathcal{N}^G_{4\ell+2}(g(a))$, and hence $(\mathcal{N}^G_{4\ell+2}(g(a)), g(a)) \cong (\mathcal{N}^{f(A)}_{4\ell+2}(a), a) \in \tau^k_{d,4\ell+2}$. But the same holds for the $(4\ell + 2)$-ball of $g(b)$, and hence we contradict the assumption that $G \in \mathcal{P}_{\hat{\rho}_k}$ since $\hat{\rho}_k(k) = [0, 1]$. Let us further assume that there is an $a \in U(A)$ such that $(\mathcal{N}^A_2(a), a) \in \tau^i_{d,2,\sigma}$ for some $i \notin I_k$. Let j be the index such that $(\mathcal{N}^{f(A)}_{4\ell+2}(a), a) \in \tau^j_{d,4\ell+2}$. Additionally note that a must have degree d in $f(A)$ by construction of f. As g is an isomorphism, we get that $(\mathcal{N}^G_{4\ell+2}(g(a)), g(a)) \in \tau^j_{d,4\ell+2}$, and $g(a)$ has degree d. But then by construction of $\hat{\rho}_k$, there must be $G' \in \mathcal{P}_{\text{graph}}$, and a vertex $v \in V(G')$ of degree d such that $(\mathcal{N}^{G'}_{4\ell+2}(v), v) \in \tau^j_{d,4\ell+2}$. By construction of $\mathcal{P}_{\text{graph}}$, there is a structure $A \in \mathcal{P}'_{\textcircled{Z}}$ such that $f(A') = G'$. Since v has degree d, it must be an element in A'. Furthermore $(\mathcal{N}^{A'}_2(v), v) \in \tau^i_{d,2,\sigma}$ by choice of i and j. Hence $A' \notin \mathcal{P}_{\rho_k}$. But this contradicts Lemma 20.

Hence we have shown that $A \in P_{\rho_k}$. Then by Lemma 20 $A \in \mathcal{P}'_{\textcircled{Z}}$, and by construction $G \in \mathcal{P}_{\text{graph}}$. ◁

Since by Claim 26 we can express $\mathcal{P}_{\text{graph}}$ as a finite union of properties each defined by a 0-profile, Theorem 11 implies that $\mathcal{P}_{\text{graph}}$ is GSF-local. ◀

4.4 Putting everything together

Now we prove our main theorem.

Proof of Theorem 1. Let the property $\mathcal{P}'_{\textcircled{Z}}$ of relational structures be as defined above. Note that $\mathcal{P}'_{\textcircled{Z}}$ is not testable, as $\mathcal{P}_{\textcircled{Z}}$ is not testable [1, Theorem 4.4] and $\mathcal{P}'_{\textcircled{Z}}$ only differs from $\mathcal{P}_{\textcircled{Z}}$ by the empty structure. By Lemma 24 and Lemma 23, the graph property $\mathcal{P}_{\text{graph}}$ that is locally reduced from $\mathcal{P}'_{\textcircled{Z}}$ is not testable. Lemma 25 shows that $\mathcal{P}_{\text{graph}}$ is also a GSF-local property. Hence there exists a GSF-local property of bounded-degree graphs which is not testable. Furthermore, since having a POT implies being testable, this proves that there is a GSF-local property which has no POT. By Theorem 6 this implies that not all GSF-local properties are non-propagating. ◀

References

1 Isolde Adler, Noleen Köhler, and Pan Peng. On testability of first-order properties in bounded-degree graphs. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1578–1597. SIAM, 2021.

2 Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing*, 39(1):143–167, 2009.

3 Itai Benjamini, Oded Schramm, and Asaf Shapira. Every minor-closed property of sparse graphs is testable. *Advances in mathematics*, 223(6):2200–2218, 2010.

4 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of computer and system sciences*, 47(3):549–595, 1993.

5 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory.* Perspectives in Mathematical Logic. Springer, 1995.

6 Hendrik Fichtenberger, Pan Peng, and Christian Sohler. Every testable (infinite) property of bounded-degree graphs contains an infinite hyperfinite subproperty. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 714–726. Society for Industrial and Applied Mathematics, 2019.

7 Sebastian Forster, Danupon Nanongkai, Thatchaphol Saranurak, Liu Yang, and Sorrachai Yingchareonthawornchai. Computing and testing small connectivity in near-linear time and queries via fast local cut algorithms. *SODA*, 2020.

8 Haim Gaifman. On local and non-local properties. In *Studies in Logic and the Foundations of Mathematics*, volume 107, pages 105–135. Elsevier, 1982.

9 Oded Goldreich. *Introduction to property testing.* Cambridge University Press, 2017.

10 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

11 Oded Goldreich and Tali Kaufman. Proximity oblivious testing and the role of invariances. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 173–190. Springer, 2011.

12 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. doi:10.1007/s00453-001-0078-7.

13 Oded Goldreich and Dana Ron. On proximity-oblivious testing. *SIAM Journal on Computing*, 40(2):534–566, 2011. Preliminary version appeared at *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC 2009)*.

14 Oded Goldreich and Igor Shinkar. Two-sided error proximity oblivious testing. *Random Structures & Algorithms*, 48(2):341–383, 2016.

15 William Hanf. *The Theory of Models*, chapter Model-theoretic methods in the study of elementary logic, pages 132–145. North Holland, 1965.

16 Avinatan Hassidim, Jonathan A Kelner, Huy N Nguyen, and Krzysztof Onak. Local graph partitions for approximation and testing. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31. IEEE, 2009.

17 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *BULL. AMER. MATH. SOC.*, 43(4):439–561, 2006.

18 Hiro Ito, Areej Khoury, and Ilan Newman. On the characterization of 1-sided error strongly testable graph properties for bounded-degree graphs. *Computational Complexity*, 29(1):1–45, 2020.

19 Ken-ichi Kawarabayashi and Yuichi Yoshida. Testing subdivision-freeness: property testing meets structural graph theory. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 437–446. ACM, 2013.

20 Akash Kumar, C Seshadhri, and Andrew Stolman. Random walks and forbidden minors ii: a poly (d ε-1)-query tester for minor-closed properties of bounded degree graphs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 559–567, 2019.

21 László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012.

22 Ilan Newman and Christian Sohler. Every property of hyperfinite graphs is testable. *SIAM Journal on Computing*, 42(3):1095–1112, 2013.

23 Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of mathematics*, pages 157–187, 2002.

24 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

25 Yuichi Yoshida and Hiro Ito. Property testing on k-vertex-connectivity of graphs. *Algorithmica*, 62(3-4):701–712, 2012.

A Formal definitions of property testers and POTs

Now we give the formal definitions of standard property testing and proximity-oblivious testing.

▶ **Definition 27** ((Standard) property testing)**.** *Let $\mathcal{P} = \cup_{n \in \mathbb{N}} \mathcal{P}_n$ be a property. An ϵ-*tester *for $\mathcal{P}_n$ is a probabilistic algorithm which, given query access to a structure $A \in \mathcal{C}$ with n vertices/elements,*

- *accepts A with probability $2/3$ if $A \in \mathcal{P}_n$.*
- *rejects A with probability $2/3$ if A is ϵ-far from $\mathcal{P}_n$.*

We say that a property $\mathcal{P}$ is testable *if for every $n \in \mathbb{N}$ and $\epsilon \in (0,1)$, there exists an ϵ-tester for $\mathcal{P}_n$ that makes at most $q = q(\epsilon, d)$ queries. We say the property $\mathcal{P}$ is testable with* one-sided error *if the ϵ-tester always accepts A if $A \in \mathcal{P}$.*

We introduce below the formal definition of proximity-oblivious testers.

▶ **Definition 28** ((One-sided error) proximity-oblivious testing)**.** *Let $\mathcal{P} = \cup_{n \in \mathbb{N}} \mathcal{P}_n$ be a property. Let $\eta : (0,1] \to (0,1]$ be a monotone function. A* proximity-oblivious tester (POT) *with detection probability η for $\mathcal{P}_n$ is a probabilistic algorithm which, given query access to a structure $A \in \mathcal{C}$ with n vertices/elements,*

- *accepts A with probability 1 if $A \in \mathcal{P}_n$.*
- *rejects A with probability at least $\eta(\mathrm{dist}(A, \mathcal{P}_n))$ if $A \notin \mathcal{P}_n$, where $\mathrm{dist}(A, \mathcal{P}_n)$ is the minimum fraction of different edges between A and any other $A' \in \mathcal{P}_n$.*

We say that a property $\mathcal{P}$ is proximity-oblivious testable *if for every $n \in \mathbb{N}$, there exists a POT for $\mathcal{P}_n$ of constant query complexity with detection probability η.*

B The FO formula

For the construction of the formula $\varphi_{\textcircled{Z}}$ we use a recursively defined sequence $(G_m)_{m \in \mathbb{N}_{>0}}$ of edge expanders [17, Proposition 9.2]. Using this sequence we define the formula $\varphi_{\textcircled{Z}}$ in such a way that any model restricted to relation F forms a rooted complete D^4-ary tree. Furthermore, the formula enforces that restricted to the vertices of level i of the tree the relation E encodes the rotation map of the expander G_i. The formula $\varphi_{\textcircled{Z}}$ is the conjunction of the following formulas. For a more detailed explanation and a proof of the precise form of the models of $\varphi_{\textcircled{Z}}$ see [1].

We use the following formula

$$\varphi_{\text{root}}(x) := \forall y \neg F(y, x),$$

which expresses that vertex x is a root vertex, i. e. has no incoming F-edges. We then define the formula φ_{tree} which expresses that the structure restricted to the relation F locally looks like a tree. More precisely, the formula expresses that there is precisely one root vertex, that every other vertex has one incoming F-edge and every vertex either has no F-children or has precisely D^4 F-children. We furthermore attach a R-self-loop to the root and D^4 L-self-loops to the leaves. This was important in [1] to make structures degree regular, but is of no relevance to this proof.

$$\begin{aligned}
&\varphi_{\text{tree}} := \exists^{=1} x \varphi_{\text{root}}(x) \wedge \\
&\quad \forall x \Big(\big(\varphi_{\text{root}}(x) \wedge R(x,x)\big) \vee \big(\exists^{=1} y F(y,x) \wedge \neg\exists y R(x,y) \wedge \neg\exists y R(y,x)\big) \Big) \wedge \\
&\quad \forall x \Bigg(\Big[\neg\exists y F(x,y) \wedge \bigwedge_{k \in ([D]^2)^2} L_k(x,x) \wedge \forall y \big(y \neq x \rightarrow \bigwedge_{k \in ([D]^2)^2} \neg L_k(x,y) \wedge \\
&\quad \bigwedge_{k \in ([D]^2)^2} \neg L_k(y,x) \big) \Big] \vee \Big[\neg\exists y \bigvee_{k \in ([D]^2)^2} \big(L_k(x,y) \vee L_k(y,x) \big) \wedge \\
&\quad \bigwedge_{k \in ([D]^2)^2} \exists y_k \Big(x \neq y_k \wedge F_k(x,y_k) \wedge \big(\bigwedge_{\substack{k' \in ([D]^2)^2 \\ k' \neq k}} \neg F_{k'}(x,y_k)\big) \wedge \forall y (y \neq y_k \rightarrow \neg F_k(x,y)) \Big) \Big] \Bigg).
\end{aligned}$$

We define formula $\varphi_{\text{rotationMap}}$ which expresses that the edge relations restricted to the relations E encode a rotation map.

$$\begin{aligned}
&\varphi_{\text{rotationMap}} := \forall x \forall y \Big(\bigwedge_{i,j \in [D]^2} (E_{i,j}(x,y) \rightarrow E_{j,i}(y,x)) \Big) \wedge \\
&\forall x \Big(\bigwedge_{i \in [D]^2} \Big(\bigvee_{j \in [D]^2} (\exists^{=1} y E_{i,j}(x,y) \wedge \bigwedge_{\substack{j' \in [D]^2 \\ j' \neq j}} \neg\exists y E_{i,j'}(x,y)) \Big) \Big)
\end{aligned}$$

The formula φ_{base} expresses that the children of the root vertex form the basis of the recursive construction of expanders. The basis of the recursive construction is the square of some D regular graph H on D^4 vertices with edge expansion ratio $1/4$. Explicit constructions of graphs with such properties are given in [23]. We assume that this graph is given by a rotation map ROT_H, which is an encoding of H.

$$\begin{aligned}
\varphi_{\text{base}} :=& \forall x \Big(\varphi_{\text{root}}(x) \rightarrow \Big[\bigwedge_{i,j \in [D]^2} \Big(E_{i,j}(x,x) \wedge \forall y \Big(x \neq y \rightarrow (\neg E_{i,j}(x,y) \wedge \neg E_{i,j}(y,x)) \Big) \Big) \wedge \\
& \bigwedge_{\substack{\text{ROT}_{H^2}(k,i) = (k',i') \\ k,k' \in ([D]^2)^2 \\ i,i' \in [D]^2}} \exists y \exists y' \big(F_k(x,y) \wedge F_{k'}(x,y') \wedge E_{i,i'}(y,y') \big) \Big] \Big)
\end{aligned}$$

We define the formula $\varphi_{\text{recursion}}$ which expresses the recursive construction of the sequence $(G_m)_{m \in \mathbb{N}_{>0}}$. This formula also depends on the base graph H.

$$\begin{aligned}
\varphi_{\text{recursion}} :=& \forall x \forall z \Big[\Big(\neg\exists y F(x,y) \wedge \neg\exists y F(z,y) \Big) \vee \\
& \bigwedge_{\substack{k'_1, k'_2 \in [D]^2 \\ \ell'_1, \ell'_2 \in [D]^2}} \Big(\exists y \big[E_{k'_1,\ell'_1}(x,y) \wedge E_{k'_2,\ell'_2}(y,z) \big] \rightarrow \\
& \bigwedge_{\substack{i,j,i',j' \in [D], k,\ell \in ([D]^2)^2 \\ \text{ROT}_H(k,i) = ((k'_1,k'_2),i') \\ \text{ROT}_H((\ell'_2,\ell'_1),j) = (\ell,j')}} \exists x' \exists z' \big[F_k(x,x') \wedge F_\ell(z,z') \wedge E_{(i,j),(j',i')}(x',z') \big] \Big) \Big]
\end{aligned}$$

C Deferred proofs from Section 3

Proof of Lemma 9. For the first direction assume φ is an FO-sentence. Then by Hanf's Theorem (Theorem 2) there is a sentence ψ in Hanf normal form such that $\mathcal{P}_\varphi = \mathcal{P}_\psi$.

We will first convert ψ into a sentence in Hanf normal form where every Hanf sentence appearing has the same locality radius. Let $r \in \mathbb{N}$ be the maximum locality radius appearing in ψ, and let $\varphi_\tau^{\geq m} := \exists^{\geq m} x \varphi_\tau(x)$ be a Hanf sentence, where τ is an r'-type for some $r' \leq r$. Let $\tau_1, \dots, \tau_k$ be a list of all r-types of bounded degree d for which $(\mathcal{N}_{r'}^B(b), b) \in \tau$ for $(B, b) \in \tau_i$, for every $1 \leq i \leq k$. Let Π be the set of all partitions of m into k parts. Let

$$\tilde{\varphi}_\tau^{\geq m} := \bigvee_{(m_1,\dots,m_k)\in\Pi} \ \bigwedge_{i=1}^{k} \exists^{\geq m_i} x \varphi_{\tau_i}(x).$$

▷ Claim 29. $\varphi_\tau^{\geq m}$ is d-equivalent to $\tilde{\varphi}_\tau^{\geq m}$.

Proof. Assume that $A \in \mathcal{C}_d$ satisfies $\varphi_\tau^{\geq m}$, and assume that $a_1, \dots, a_m$ are m distinct elements with $(\mathcal{N}_{r'}^A(a_j), a_j) \in \tau$, for every $1 \leq j \leq m$. Let $\tilde{\tau}_j$ be the r-type for which $(\mathcal{N}_r^A(a_j), a_j) \in \tilde{\tau}_j$. By choice of $\tau_1, \dots, \tau_k$, we get that there are indices $i_1, \dots, i_m$ such that $\tilde{\tau}_j = \tau_{i_j}$. For $i \in \{1, \dots, k\}$ let $m_i = |\{j \in \{1, \dots, m\} \mid i_j = i\}|$. Hence $A \models \bigwedge_{i=1}^{k} \exists^{\geq m_i} x \varphi_{\tau_i}(x)$ and since additionally $(m_1, \dots, m_k) \in \Pi$ this implies $A \models \tilde{\varphi}_\tau^{\geq m}$.

On the other hand, let $A \in \mathcal{C}_d$ satisfy $\tilde{\varphi}_\tau^{\geq m}$, and let $(m_1, \dots, m_k) \in \Pi$ be a partition of m such that $A \models \bigwedge_{i=1}^{k} \exists^{\geq m_i} x \varphi_{\tau_i}(x)$. For every $1 \leq i \leq k$, let $a_1^i, \dots, a_{m_i}^i$ be m_i distinct elements such that $(\mathcal{N}_r^A(a_j^i), a_j^i) \in \tau_i$, for every $1 \leq j \leq m_i$. By choice of $\tau_1, \dots, \tau_k$, we get that $(\mathcal{N}_{r'}^A(a_j^i), a_j^i) \in \tau$, for every pair $1 \leq i \leq k$, $1 \leq j \leq m_i$. But since $m_1 + \dots + m_k = m$ this implies that $A \models \varphi_\tau^{\geq m}$. This proves that $\varphi_\tau^{\geq m}$ and $\tilde{\varphi}_\tau^{\geq m}$ are d-equivalent. ◁

Let ψ' be the formula in which every Hanf-sentence $\varphi_\tau^{\geq m}$ for which τ is an r'-type for some $r' < r$ gets replaced by $\tilde{\varphi}_\tau^{\geq m}$. By a simple inductive argument using Claim 29, we get that ψ is d-equivalent to ψ', and hence $\mathcal{P}_\varphi = \mathcal{P}_\psi = \mathcal{P}_{\psi'}$. Furthermore since $\tilde{\varphi}_\tau^{\geq m}$ is a Boolean combination of Hanf-sentences for every $\varphi_\tau^{\geq m}$, and any Boolean combination of Boolean combinations is a Boolean combination itself, ψ' is in Hanf normal form. Furthermore, every Hanf-sentence appearing in ψ' has locality radius r by construction.

Since any Boolean combination can be converted into disjunctive normal form, we can assume that ψ' is a disjunction of sentences ξ of the form

$$\xi = \bigwedge_{j=1}^{k} \exists^{\geq m_j} x \varphi_{\tau_j}(x) \wedge \bigwedge_{j=k+1}^{\ell} \neg\exists^{\geq m_j} x \varphi_{\tau_j}(x),$$

where $\ell \in \mathbb{N}_{\geq 1}$, $1 \leq k \leq \ell$, $m_i \in \mathbb{N}_{\geq 1}$ and τ_i is an r-type for every $1 \leq i \leq \ell$. We can further assume that every sentence in the disjunction ψ' is satisfiable by some $A \in \mathcal{C}_d$, as any sentence with no bounded degree d model can be removed from ψ'.

Let $\tilde{\tau}_1, \dots, \tilde{\tau}_t$ be a list of all r-types of bounded degree d in the order we fixed. Let $k_i := \max(\{m_j \mid 1 \leq j \leq k, \tau_j = \tilde{\tau}_i\} \cup \{0\})$ and $\ell_i := \min(\{m_j \mid k+1 \leq j \leq \ell, \tau_j = \tilde{\tau}_i\} \cup \{\infty\})$ for every $i \in \{1, \dots, t\}$. Since ξ has at least one bounded degree model $k_i \leq \ell_i$ for every $i \in \{1, \dots, t\}$. Let $\rho : \{1, \dots, t\} \to \mathfrak{I}$ be the neighbourhood profile defined by $\rho(i) := [k_i, \ell_i]$ if $\ell_i < \infty$ and $\rho(i) := [k_i, \ell_i)$ otherwise. Then by construction, we get that $\mathcal{P}_\rho = \mathcal{P}_\xi$. Since ψ' is a disjunction of formulas, each of which defines a property which can be defined by some neighbourhood profile, we get that $\mathcal{P}_{\psi'}$ must be a finite union of properties defined by some neighbourhood profile.

On the other hand, for every r-neighbourhood profile ρ of degree d, $\tau_1, \dots, \tau_t$ a list of all r-types of bounded degree d in the order fixed and the formula

$$\varphi_\rho := \bigwedge_{\substack{i\in\{1,\dots,t\},\\ \rho(i)=[k_i,\ell_i]}} \left(\exists^{\geq k_i} x\varphi_{\tau_i}(x) \wedge \neg\exists^{\geq \ell_i+1} x\varphi_{\tau_i}(x)\right) \wedge \bigwedge_{\substack{i\in\{1,\dots,t\},\\ \rho(i)=[k_i,\infty)}} \exists^{\geq k_i} x\varphi_{\tau_i}(x)$$

it clearly holds that $\mathcal{P}_\rho = \mathcal{P}_{\varphi_\rho}$. Hence every finite union of properties defined by neighbourhood profiles can be defined by the disjunction of the formulas φ_ρ of all ρ in the finite union. ◀

D Deferred proofs from Section 4

Proof of Claim 19. Let $A, A' \in \mathcal{C}_{\sigma,d}$ such that $A \models \tilde{\varphi}_{\textcircled{Z}}$ and $A' \models \tilde{\varphi}_{\textcircled{Z}}$, where $\tilde{\varphi}_{\textcircled{Z}}$ was the formula obtained from $\varphi_{\textcircled{Z}}$ by removing the subformula $\exists^{=1} x\varphi_{\text{root}}(x)$. Our aim is to prove that $A \cup A' \models \tilde{\varphi}_{\textcircled{Z}}$ where $A \cup A'$ denotes the disjoint union of A and A'. For this we essentially prove that for any two elements $a \in U(A)$ and $b \in U(A')$ the formula $\tilde{\varphi}_{\textcircled{Z}}$ does not require a tuple containing a and b.

Let us define formulas

$$\varphi := \forall x\Big(\big(\varphi_{\text{root}}(x) \wedge R(x,x)\big) \vee \big(\exists^{=1} yF(y,x) \wedge \neg\exists yR(x,y) \wedge \neg\exists yR(y,x)\big)\Big),$$

$$\psi(x) := \neg\exists yF(x,y) \wedge \bigwedge_{k\in([D]^2)^2} L_k(x,x) \wedge$$

$$\forall y\Big(y \neq x \to \bigwedge_{k\in([D]^2)^2} \neg L_k(x,y) \wedge \bigwedge_{k\in([D]^2)^2} \neg L_k(y,x)\Big) \text{ and}$$

$$\chi(x) := \neg\exists y \bigvee_{k\in([D]^2)^2} \big(L_k(x,y) \vee L_k(y,x)\big) \wedge$$

$$\bigwedge_{k\in([D]^2)^2} \exists y_k\Big(x \neq y_k \wedge F_k(x,y_k) \wedge (\bigwedge_{\substack{k'\in([D]^2)^2\\ k'\neq k}} \neg F_{k'}(x,y_k)) \wedge \forall y(y \neq y_k \to \neg F_k(x,y))\Big).$$

Then $\tilde{\varphi}_{\textcircled{Z}} := \varphi \wedge \forall x(\psi(x) \vee \chi(x)) \wedge \varphi_{\text{rotationMap}} \wedge \varphi_{\text{base}} \wedge \varphi_{\text{recursion}}$. Hence it is sufficient to prove that $A \cup A' \models \varphi$, $A \cup A' \models \forall x(\psi(x) \vee \chi(x))$, $A \cup A' \models \varphi_{\text{rotationMap}}$, $A \cup A' \models \varphi_{\text{base}}$ and $A \cup A' \models \varphi_{\text{recursion}}$.

We first argue that $A \cup A' \models \varphi$. Let $a \in U(A \cup A')$ be arbitrary and assume without loss of generality that $a \in U(A)$. Assume that $A \cup A' \not\models \varphi_{\text{root}}(a) \wedge R(a,a)$. Since $\varphi_{\text{root}}(x) := \forall y\neg F(y,x)$ this implies that $A \not\models \varphi_{\text{root}}(a) \wedge R(a,a)$. Since $A \models \varphi$ we get that $A \models \exists^{=1} yF(y,a) \wedge \neg\exists yR(a,y) \wedge \neg\exists yR(y,a)$. Hence there is an element $b \in U(A)$ such that $(b,a) \in F(A)$. Furthermore, for every $b' \in U(A)$, $b' \neq b$ we have $(b',a) \notin F(A)$, $(a,b') \notin R(A)$ and $(b',a) \notin R(A)$. But because a cannot be in a tuple with any element in $U(A')$ we get that $A \cup A' \models \exists^{=1} yF(y,a) \wedge \neg\exists yR(a,y) \wedge \neg\exists yR(y,a)$. Hence $A \cup A' \models \varphi$.

Next we prove that $A \cup A' \models \forall x(\psi(x) \vee \chi(x))$. Let $a \in U(A \cup A')$ be arbitrary and assume without loss of generality that $a \in U(A)$. First assume that $(a,b) \notin F(A \cup A')$ for every $b \in U(A \cup A')$. Since A is a substructure of $A \cup A'$ this means that $A \models \neg\exists yF(a,y)$. But then $A \not\models \bigwedge_{k\in([D]^2)^2} \exists y_k\Big(a \neq y_k \wedge F_k(a,y_k)\Big)$ which implies $A \not\models \chi(a)$. Since $A \models \forall x(\psi(x) \vee \chi(x))$ this implies that $A \models \psi(a)$. Hence for every $k \in ([D]^2)^2$ we have $(a,a) \in L_k(A)$ and for every $b \in U(A)$, $b \neq a$ we have $(a,b), (b,a) \notin F_k(A)$. Since there are no tuples containing both elements from A and A' this directly implies that $A \cup A' \models \psi(a)$.

On the other hand, assume that there is $b \in U(A \cup A')$ such that $(a,b) \in F(A \cup A')$. Since we are considering the disjoint union of A and A' this implies that b must be an element from A. Hence $A \not\models \psi(a)$. Since $A \models \forall x(\psi(x) \vee \chi(x))$ this implies that $A \models \chi(a)$. Then for every $k \in ([D]^2)^2$ there is an element $b \in U(A)$ such that $(a,b) \in F_k(A)$, $(a,b) \notin F_{k'}(A)$ for every $k' \in ([D]^2)^2$, $k' \neq k$ and $(a,b') \notin F_k(A)$ for every $b' \in U(A)$, $b' \neq b$. But since in $A \cup A'$ there are no tuples containing both elements from A and A' this implies that $A \cup A' \models \chi(a)$. In conclusion we proved that $A \cup A' \models \forall x(\psi(x) \vee \chi(x))$.

We now prove $A \cup A' \models \varphi_{\text{rotationMap}}$. Hence assume $a, b \in U(A \cup A')$ are arbitrary elements. First consider the case that a, b are either both from $U(A)$ or both from $U(A')$. Then if for some $i, j \in [D]^2$ we have that $(a,b) \in E_{i,j}(A \cup A')$ then $(b,a) \in E_{j,i}(A \cup A')$ because $A \models \varphi_{\text{rotationMap}}$ and $A' \models \varphi_{\text{rotationMap}}$. Now consider the case that $|\{a,b\} \cap U(A)| = 1$. Then $(a,b) \notin E_{i,j}(A \cup A')$ and $(b,a) \notin E_{j,i}(A \cup A')$ and hence $A \cup A' \models \bigwedge_{i,j \in [D]^2}(E_{i,j}(a,b) \to E_{j,i}(b,a))$. Therefore $A \cup A' \models \forall x \forall y \Big(\bigwedge_{i,j \in [D]^2}(E_{i,j}(x,y) \to E_{j,i}(y,x))\Big)$.

Now consider an arbitrary element $a \in U(A \cup A')$ and any $i \in [D]^2$. Without loss of generality assume $a \in U(A)$. Since $A \models \varphi_{\text{rotationMap}}$ there must be an index $j \in [D]^2$ and an element $b \in U(A)$ such that $(a,b) \in E_{i,j}(A)$. Furthermore, for every $b' \in U(A)$, $b' \neq b$ we have $(a,b') \notin E_{i,j}(A)$ and for every $j' \in [D]^2$, $j' \neq j$ and every $\tilde{b} \in U(A)$ we have $(a,\tilde{b}) \notin E_{i,j'}(A)$. But since $a \in U(A)$ it also holds that $(a,b') \notin E_{i,j'}(A)$ for every $b' \in U(A')$ and every $j' \in [D]^2$. Hence $A \cup A' \models \bigvee_{j \in [D]^2} \big(\exists^{=1} y E_{i,j}(a,y) \wedge \bigwedge_{\substack{j' \in [D]^2 \\ j' \neq j}} \neg \exists y E_{i,j'}(a,y)\big)$. This concludes the proof of $A \cup A' \models \varphi_{\text{rotationMap}}$.

We now prove $A \cup A' \models \varphi_{\text{base}}$. Assume $a \in U(A \cup A')$ is an arbitrary element such that $A \cup A' \models \varphi_{\text{root}}(a)$. Without loss of generality assume $a \in U(A)$. Since $\varphi_{\text{root}}(x) := \forall y \neg F(y,x)$ and $A \cup A' \models \varphi_{\text{root}}(a)$ we get that $A \models \varphi_{\text{root}}(a)$. Since $A \models \varphi_{\text{base}}$ this means that for every $i, j \in [D]^2$ we have $(a,a) \in E_{i,j}(A)$ and $(a,b), (b,a) \notin E_{i,j}(A)$ for every $b \in U(A)$, $b \neq a$. Since further $(a,b), (b,a) \notin E_{i,j}(A \cup A')$ for every $b \in U(A')$ this implies that $A \cup A' \models \bigwedge_{i,j \in [D]^2} \Big(E_{i,j}(a,a) \wedge \forall y \Big(a \neq y \to \big(\neg E_{i,j}(a,y) \wedge \neg E_{i,j}(y,a)\big)\Big)\Big)$. Furthermore, since $A \models \varphi_{\text{base}}$ and $A \models \varphi_{\text{root}}(a)$ for every $k, k' \in ([D]^2)^2, i, i' \in [D]^2$ for which $\text{ROT}_{H^2}(k,i) = (k', i')$ there are $b, b' \in U(A)$ such that $(a,b) \in F_k(A)$, $(a,b') \in F_{k'}(A)$ and $(b,b') \in E_{i,i'}(A)$. Since A is a substructure of $A \cup A'$ this proves that $A \cup A \models \varphi_{\text{base}}$.

Finally we prove $A \cup A' \models \varphi_{\text{recursion}}$. Hence assume $a, c \in U(A \cup A')$ are arbitrary elements. Assume $A \cup A' \not\models \neg \exists y F(a,y) \wedge \neg \exists y F(c,y)$ and assume without loss of generality that there is $\tilde{a} \in U(A \cup A')$ such that $(a, \tilde{a}) \in F(A \cup A')$. Since there are no tuples containing both elements from A and A' we get that $a, \tilde{a}$ are from the same structure. Without loss of generality assume $a, \tilde{a} \in U(A)$. Assume that for indices $k'_1, k'_2 \in [D]^2, \ell'_1, \ell'_2 \in [D]^2$ and some element $b \in U(A \cup A')$ we have $(a,b) \in E_{k'_1, \ell'_1}(A \cup A')$ and $(b,c) \in E_{k'_2, \ell'_2}(A \cup A')$. As b also has to be in $U(A)$ and $A \models \varphi_{\text{recursion}}$ this implies that for every $i, j, i', j' \in [D], k, \ell \in ([D]^2)^2$ for which $\text{ROT}_H(k,i) = ((k'_1, k'_2), i')$ and $\text{ROT}_H((\ell'_2, \ell'_1), j) = (\ell, j')$ there are elements $a', c' \in U(A \cup A')$ such that $(a,a') \in F_k(A \cup A')$, $(c,c') \in F_\ell(A \cup A')$ and $(a',c') \in E_{(i,j),(j',i')}(A \cup A')$. Hence $A \cup A' \models \varphi_{\text{recursion}}$. ◁

▷ Claim 30. Every structure $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$ satisfies φ'_{tree}.

Proof. Let $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$. Then there is $k \in \{1, \ldots, m\}$ such that $A \in \mathcal{P}_{\rho_k}$.

By definition, $\varphi'_{\text{tree}} := \exists^{\leq 1} x \varphi_{\text{root}}(x) \wedge \varphi \wedge \forall x(\psi(x) \vee \chi(x))$, where

$$\varphi := \forall x \Big(\big(\varphi_{\text{root}}(x) \wedge R(x,x)\big) \vee \big(\exists^{=1} y F(y,x) \wedge \neg \exists y R(x,y) \wedge \neg \exists y R(y,x)\big)\Big),$$

$$\begin{aligned}
&\psi(x) := \neg\exists y F(x,y) \wedge \bigwedge_{k\in([D]^2)^2} L_k(x,x)\wedge \\
&\qquad \forall y\Big(y \neq x \to \bigwedge_{k\in([D]^2)^2} \neg L_k(x,y) \wedge \bigwedge_{k\in([D]^2)^2} \neg L_k(y,x)\Big) \text{ and} \\
&\chi(x) := \neg\exists y \bigvee_{k\in([D]^2)^2} \big(L_k(x,y) \vee L_k(y,x)\big)\wedge \\
&\qquad \bigwedge_{k\in([D]^2)^2} \exists y_k\Big(x \neq y_k \wedge F_k(x,y_k) \wedge (\bigwedge_{\substack{k'\in([D]^2)^2\\ k'\neq k}} \neg F_{k'}(x,y_k)) \wedge \forall y(y \neq y_k \to \neg F_k(x,y))\Big).
\end{aligned}$$

Thus, it is sufficient to prove that $A \models \exists^{\leq 1} x\varphi_{\mathrm{root}}(x)$, $A \models \varphi$ and $A \models \forall x(\psi(x) \vee \chi(x))$.

To prove $A \models \exists^{\leq 1} x\varphi_{\mathrm{root}}(x)$ we note that by construction of ρ_k we have $A \not\models \varphi_{\mathrm{root}}(a)$ for any $a \in U(A)$ for which $(\mathcal{N}_2^A(a), a) \notin \tau^k_{d,2,\sigma}$. Since ρ_k restricts the number of occurrences of elements of neighbourhood type $\tau^k_{d,2,\sigma}$ to at most one, this proves that there is at most one $a \in U(A)$ with $A \models \varphi_{\mathrm{tree}}(a)$ and hence $A \models \exists^{\leq 1} x\varphi_{\mathrm{root}}(x)$.

To prove $A \models \varphi$, let $a \in U(A)$ be an arbitrary element. Since $A \in \mathcal{P}_{\rho_k}$, there is an $i \in I_k$ such that $(\mathcal{N}_2^A(a), a) \in \tau^i_{d,2,\sigma}$. But then by definition, there exist $\tilde{A} \models \varphi_{\textcircled{Z}}$ and $\tilde{a} \in U(\tilde{A})$ such that $(\mathcal{N}_2^A(a), a) \cong (\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Assume f is an isomorphism from $(\mathcal{N}_2^A(a), a)$ to $(\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. First consider the case that $A \models \varphi_{\mathrm{root}}(a) := \forall y \neg F(y,a)$. Assume there is $\tilde{b} \in U(\tilde{A})$ such that $(\tilde{b}, \tilde{a}) \in F(\tilde{A})$. Since $\tilde{b} \in N_2^{\tilde{A}}(\tilde{a})$, there must be an element $b \in N_2^A(a)$ such that $f(b) = \tilde{b}$. Since f is an isomorphism mapping a to $\tilde{a}$, this implies $(b,a) \in F(A)$, which contradicts $A \models \varphi_{\mathrm{root}}(a)$. Hence $\tilde{A} \models \varphi_{\mathrm{root}}(\tilde{a})$. Since $\tilde{A} \models \varphi'_{\mathrm{tree}}$, it holds that $\tilde{A} \models \varphi$, which means that $(\tilde{a}, \tilde{a}) \in R(\tilde{A})$. But since f is an isomorphism mapping a onto $\tilde{a}$, this implies $(a,a) \in R(A)$. Now consider the case that $A \not\models \varphi_{\mathrm{root}}(a)$. Then there is $b \in U(A)$ with $(b,a) \in F(A)$. Since f is an isomorphism, this implies $(f(b), \tilde{a}) \in F(\tilde{A})$. Hence $\tilde{A} \models \exists^{=1} y F(y, \tilde{a}) \wedge \neg\exists y R(\tilde{a}, y) \wedge \neg\exists y R(y, \tilde{a})$, as $\tilde{A} \models \varphi$. Now assume that there is $b' \neq b$ such that $(b', a) \in F(A)$. Then $f(b) \neq f(b')$ and $(f(b), \tilde{a}), (f(b'), \tilde{a}) \in F(\tilde{A})$. Since this contradicts $\tilde{A} \models \exists^{=1} y F(y, \tilde{a})$ we have $A \models \exists^{=1} y F(y, a)$. Furthermore, assume that there is $b' \in U(A)$ such that either $(a, b') \in R(A)$ or $(b', a) \in R(A)$. Then either $(\tilde{a}, f(b')) \in R(\tilde{A}')$ or $(f(b'), \tilde{a}) \in R(\tilde{A})$, which contradicts $\tilde{A} \models \neg\exists R(\tilde{a}, y) \wedge \neg\exists y R(y, \tilde{a})$. Therefore $A \models \neg\exists R(a,y) \wedge \neg\exists y R(y,a)$ which completes the proof of $A \models \varphi$.

We prove $A \models \forall x(\psi(x) \vee \chi(x))$ by considering the two cases $A \models \neg\exists y F(a,y)$ and $A \models \exists y F(a,y)$ for each element $a \in U(A)$. For this, let $a \in U(A)$ be any element. By the construction of ρ_k there is $\tilde{A} \models \varphi_{\textcircled{Z}}$ and $\tilde{a} \in U(\tilde{A})$ such that $(\mathcal{N}_2^A(a), a) \cong (\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Let f be an isomorphism from $(\mathcal{N}_2^A(a), a)$ to $(\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. First consider the case that $A \models \neg\exists y F(a,y)$. If there was $\tilde{b} \in U(\tilde{A})$ with $(\tilde{a}, \tilde{b}) \in F(\tilde{A})$ then $(a, f^{-1}(\tilde{b})) \in F(A)$ contradicting our assumption. Hence $\tilde{A} \models \neg\exists y F(\tilde{a}, y)$ which implies that $\tilde{A} \not\models \chi(\tilde{a})$. But since $\tilde{A} \models \varphi_{\textcircled{Z}}$, it holds that $\tilde{A} \models \forall x(\psi(x) \vee \chi(x))$, which implies that $\tilde{A} \models \psi(\tilde{a})$. Hence $(\tilde{a}, \tilde{a}) \in L_k(\tilde{A})$ for every $k \in ([D]^2)^2$. Since f is an isomorphism and $f(a) = \tilde{a}$, it holds that $(a,a) \in L_k(A)$ for every $k \in ([D]^2)^2$, and hence $A \models \bigwedge_{k\in([D]^2)^2} L_k(a,a)$. Furthermore, assume that there is $b \in U(A)$, $b \neq a$ and $k \in ([D]^2)^2$ such that either $(a,b) \in L_k(A)$ or $(b,a) \in L_k(A)$. Since f is an isomorphism this implies that either $(\tilde{a}, f(b)) \in L_k(\tilde{A})$ or $(f(b), \tilde{a}) \in L_k(\tilde{A})$ which contradicts $\tilde{A} \models \chi(\tilde{a})$. Hence $A \models \forall y\Big(y \neq a \to \bigwedge_{k\in([D]^2)^2} \neg L_k(a,y) \wedge \bigwedge_{k\in([D]^2)^2} \neg L_k(y,a)\Big)$ proving that $A \models \psi(a)$.

Now consider the case that there is an element $b \in U(A)$ such that $(a,b) \in F(A)$. Since this implies that $(\tilde{a}, f(b)) \in F(\tilde{A})$, we get that $\tilde{A} \not\models \psi(\tilde{a})$, and hence $\tilde{A} \models \chi(\tilde{a})$. Now assume that there is $b \in U(A)$ and $k \in ([D]^2)^2$ such that either $(a,b) \in L_k(A)$ or $(b,a) \in L_k(A)$. But then either $(\tilde{a}, f(b)) \in L_k(\tilde{A})$ or $(f(b), \tilde{a}) \in L_k(\tilde{A})$, which contradicts $\tilde{A} \models \chi(\tilde{a})$. Hence $A \models \neg\exists y \bigvee_{k \in ([D]^2)^2} \big(L_k(a,y) \vee L_k(y,a)\big)$. For each $k \in ([D]^2)^2$, let $\tilde{b}_k \in U(\tilde{A})$ be an element such that $\tilde{A} \models \tilde{a} \neq \tilde{b}_k \wedge F_k(\tilde{a}, \tilde{b}_k) \wedge (\bigwedge_{k' \in ([D]^2)^2, k' \neq k} \neg F_{k'}(\tilde{a}, \tilde{b}_k)) \wedge \forall y(y \neq \tilde{b}_k \to \neg F_k(\tilde{a}, y))$. Since f is an isomorphism, this implies that $a \neq b_k := f^{-1}(\tilde{b}_k)$, $(a, b_k) \in F_k(A)$ and $(a, b_k) \notin F_{k'}(A)$, for each $k' \in ([D]^2)^2, k' \neq k$. Furthermore, assume there is $b \in U(A)$, $b \neq b_k$ such that $(a,b) \in F_k(A)$. Since f is an isomorphism, this implies $f(b) \neq f(b_k) = \tilde{b}_k$ and $(\tilde{a}, \tilde{b}) \in F_k(\tilde{A})$, which contradicts $\tilde{A} \models \forall y(y \neq \tilde{b}_k \to \neg F_k(\tilde{a}, y))$. Hence $A \models \forall y(y \neq b_k \to \neg F_k(a,y))$ and therefore concluding that $A \models \chi(a)$. This proves that in either case $A \models \psi(a) \vee \chi(a)$ and therefore $A \models \forall x(\psi(x) \vee \chi(x))$. ◁

▷ Claim 31. Every structure $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$ satisfies $\varphi_{\text{rotationMap}}$.

Proof. Let $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$. Then there is a $k \in \{1, \dots, m\}$ such that $A \in \mathcal{P}_{\rho_k}$.

By definition, $\varphi_{\text{rotationMap}} = \varphi \wedge \psi$, where

$$\varphi := \forall x \forall y \Big(\bigwedge_{i,j \in [D]^2} (E_{i,j}(x,y) \to E_{j,i}(y,x)) \Big) \text{ and}$$

$$\psi := \forall x \Big(\bigwedge_{i \in [D]^2} \Big(\bigvee_{j \in [D]^2} (\exists^{=1} y E_{i,j}(x,y) \wedge \bigwedge_{\substack{j' \in [D]^2 \\ j' \neq j}} \neg \exists y E_{i,j'}(x,y)) \Big) \Big).$$

Thus, it is sufficient to prove that $A \models \varphi$ and $A \models \psi$.

To prove $A \models \varphi$, assume towards a contradiction that there are $a, b \in U(A)$ such that for some pair $i, j \in [D]^2$, we have that $(a,b) \in E_{i,j}(A)$, but $(b,a) \notin E_{j,i}(A)$. By construction of $\mathcal{P}_{\rho_k}$, there is a structure $\tilde{A} \models \varphi_{\textcircled{Z}}$ and $\tilde{a} \in U(\tilde{A})$ such that $(\mathcal{N}_2^A(a), a) \cong (\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Assume f is an isomorphism from $(\mathcal{N}_2^A(a), a)$ to $(\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Note that $f(b)$ is defined since b is in the 2-neighbourhood of a. Furthermore since f is an isomorphism, $(a,b) \in E_{i,j}(A)$ implies $(\tilde{a}, f(b)) \in E_{i,j}(\tilde{A})$, and $(b,a) \notin E_{j,i}(A)$ implies $(f(b), \tilde{a}) \notin E_{j,i}(\tilde{A})$. Hence $\tilde{A} \not\models \varphi$, which contradicts $\tilde{A} \models \varphi_{\text{rotationMap}}$.

To prove $A \models \psi$, assume towards a contradiction that there is an $a \in U(A)$ and $i \in [D]^2$ such that $A \not\models \exists^{=1} y E_{i,j}(a,y) \wedge \bigwedge_{\substack{j' \in [D]^2 \\ j' \neq j}} \neg \exists y E_{i,j'}(a,y)$ for every $j \in [D]^2$. We know that there is a structure $\tilde{A} \models \varphi_{\textcircled{Z}}$ and $\tilde{a} \in U(\tilde{A})$ such that $(\mathcal{N}_2^A(a), a) \cong (\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Let f be an isomorphism from $(\mathcal{N}_2^A(a), a)$ to $(\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Since $\tilde{A} \models \psi$, there must be $j \in [D]^2$ such that $\tilde{A} \models \exists^{=1} y E_{i,j}(\tilde{a}, y) \wedge \bigwedge_{\substack{j' \in [D]^2 \\ j' \neq j}} \neg \exists y E_{i,j'}(\tilde{a}, y)$. Hence there must be $\tilde{b} \in U(\tilde{A})$ such that $(\tilde{a}, \tilde{b}) \in E_{i,j}(\tilde{A})$, which implies that $(a, f^{-1}(\tilde{b})) \in E_{i,j}(A)$. Since we assumed that $A \not\models \exists^{=1} y E_{i,j}(a,y) \wedge \bigwedge_{\substack{j' \in [D]^2 \\ j' \neq j}} \neg \exists y E_{i,j'}(a,y)$, there must be either $b \neq f^{-1}(\tilde{b})$ with $(a,b) \in E_{i,j}(A)$, or there must be $j' \in [D]^2$, $j' \neq j$ and $b' \in U(A)$ such that $(a, b') \in E_{i,j'}(A)$. In the first case $(\tilde{a}, f(b)) \in E_{i,j}(\tilde{A})$, since f is an isomorphism. But then $\tilde{A} \not\models \exists^{=1} y E_{i,j}(\tilde{a}, y)$, which is a contradiction. In the second case, we get that $(\tilde{a}, f(b')) \in E_{i,j'}(\tilde{A})$. But then $\tilde{A} \not\models \bigwedge_{\substack{j' \in [D]^2 \\ j' \neq j}} \neg \exists y E_{i,j'}(\tilde{a}, y)$, which is a contradiction. Hence $A \models \varphi \wedge \psi$. ◁

▷ Claim 32. Every structure $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$ satisfies φ_{base}.

Proof. Let $A \in \bigcup_{1 \leq k \leq m} \mathcal{P}_{\rho_k} \setminus \{A_\emptyset\}$. Then there is a $k \in \{1, \dots, m\}$ such that $A \in \mathcal{P}_{\rho_k}$.

By definition, $\varphi_{\text{base}} := \forall x\big(\varphi_{\text{root}}(x) \to (\varphi(x) \wedge \psi(x))\big)$, where

$$\varphi(x) := \bigwedge_{i,j\in[D]^2} \Big(E_{i,j}(x,x) \wedge \forall y\Big(x \neq y \to (\neg E_{i,j}(x,y) \wedge \neg E_{i,j}(y,x))\Big)\Big) \text{ and}$$

$$\psi(x) := \bigwedge_{\substack{\mathrm{ROT}_{H^2}(k,i)=(k',i')\\ k,k'\in([D]^2)^2\\ i,i'\in[D]^2}} \exists y \exists y' \big(F_k(x,y) \wedge F_{k'}(x,y') \wedge E_{i,i'}(y,y')\big).$$

Thus, it is sufficient to prove that $A \models \varphi(a)$ and $A \models \psi(a)$ for every $a \in U(A)$ for which $A \models \varphi_{\text{root}}(a)$. Therefore assume $a \in U(A)$ is any element such that $A \models \varphi_{\text{root}}(a)$. Because $A \in \mathcal{P}_{\rho_k}$ there is an $i \in I_k$ such that $(\mathcal{N}_2^A(a), a) \in \tau^i_{d,2,\sigma}$. Then by definition there is a structure $\tilde{A} \models \varphi_{Ⓩ}$ and $\tilde{a} \in U(\tilde{A})$ such that $(\mathcal{N}_2^A(a), a) \cong (\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Let f be an isomorphism from $(\mathcal{N}_2^A(a), a)$ to $(\mathcal{N}_2^{\tilde{A}}(\tilde{a}), \tilde{a})$. Assume that there is an element $\tilde{b} \in U(\tilde{A})$ such that $(\tilde{b}, \tilde{a}) \in F(\tilde{A})$. Since f is an isomorphism and $\tilde{b} \in N_2^{\tilde{A}}(\tilde{a})$ we get that $(f^{-1}(\tilde{b}), a) \in F(A)$ which contradicts that $A \models \varphi_{\text{root}}(a)$ as $\varphi_{\text{root}}(x) := \forall y \neg F(y,x)$. Hence there is no element $\tilde{b} \in U(\tilde{A})$ such that $(\tilde{b}, \tilde{a}) \in F(\tilde{A})$ which implies that $\tilde{A} \models \varphi_{\text{root}}(\tilde{a})$. But since $\tilde{A} \models \varphi_{Ⓩ}$ we have that $\tilde{A} \models \varphi_{\text{base}}$ and hence $\tilde{A} \models \varphi(\tilde{a})$ and $\tilde{A} \models \psi(\tilde{a})$.

To prove $A \models \varphi(a)$ first observe that $(a,a) \in E_{i,j}(A)$ for every $i,j \in [D]^2$ since $\tilde{A} \models \varphi(\tilde{a})$ implies that $(\tilde{a}, \tilde{a}) \in E_{i,j}(\tilde{A})$ for every $i,j \in [D]^2$ and f is an isomorphism mapping a onto $\tilde{a}$. Assume that there is an element $b \in U(A)$, $b \neq a$ and indices $i,j \in [D]^2$ such that either $(a,b) \in E_{i,j}(A)$ or $(b,a) \in E_{i,j}(A)$. Since $b \in N_2^A(a)$ and f is an isomorphism we get that $f(b) \neq f(a) = \tilde{a}$ and either $(\tilde{a}, f(b)) \in E_{i,j}(\tilde{A})$ or $(f(b), \tilde{a}) \in E_{i,j}(\tilde{A})$. But this contradicts $\tilde{A} \models \varphi(\tilde{a})$ and hence $A \models \varphi(a)$.

We now prove $A \models \psi(a)$. Let $k,k' \in ([D]^2)^2$ and $i,i' \in [D]^2$ such that $\mathrm{ROT}_{H^2}(k,i) = (k',i')$. Since $\tilde{A} \models \psi(\tilde{a})$ there must be elements $\tilde{b}, \tilde{b}' \in U(\tilde{A})$ such that $(\tilde{a}, \tilde{b}) \in F_k(\tilde{A})$, $(\tilde{a}, \tilde{b}') \in F_{k'}(\tilde{A})$ and $(\tilde{b}, \tilde{b}') \in E_{i,i'}(\tilde{A})$. But since $\tilde{b}, \tilde{b}' \in N_2^{\tilde{A}}(\tilde{a})$ we get that $f^{-1}(\tilde{b})$ and $f^{-1}(\tilde{b}')$ are defined and since f is an isomorphism we get that $(a, f^{-1}(\tilde{b})) \in F_k(A)$, $(a, f^{-1}(\tilde{b}')) \in F_{k'}(A)$ and $(f^{-1}(\tilde{b}), f^{-1}(\tilde{b}')) \in E_{i,i'}(A)$. Hence $A \models \exists y \exists y'\big(F_k(a,y) \wedge F_{k'}(a,y') \wedge E_{i,i'}(y,y')$ for any $k,k' \in ([D]^2)^2$ and $i,i' \in [D]^2$ such that $\mathrm{ROT}_{H^2}(k,i) = (k',i')$ which implies that $A \models \psi(a)$. ◁

Hardness of KT Characterizes Parallel Cryptography

Hanlin Ren
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

Rahul Santhanam
University of Oxford, UK

Abstract

A recent breakthrough of Liu and Pass (FOCS'20) shows that one-way functions exist if and only if the (polynomial-)time-bounded Kolmogorov complexity, K^t, is bounded-error hard on average to compute. In this paper, we strengthen this result and extend it to other complexity measures:

- We show, perhaps surprisingly, that the KT complexity is bounded-error average-case hard if and only if there exist one-way functions in *constant parallel time* (i.e. NC^0). This result crucially relies on the idea of *randomized encodings*. Previously, a seminal work of Applebaum, Ishai, and Kushilevitz (FOCS'04; SICOMP'06) used the same idea to show that NC^0-computable one-way functions exist if and only if logspace-computable one-way functions exist.
- Inspired by the above result, we present randomized average-case reductions among the NC^1-versions and logspace-versions of K^t complexity, and the KT complexity. Our reductions preserve both bounded-error average-case hardness and zero-error average-case hardness. To the best of our knowledge, this is the first reduction between the KT complexity and a variant of K^t complexity.
- We prove tight connections between the hardness of K^t complexity and the hardness of (the hardest) one-way functions. In analogy with the Exponential-Time Hypothesis and its variants, we define and motivate the *Perebor Hypotheses* for complexity measures such as K^t and KT. We show that a Strong Perebor Hypothesis for K^t implies the existence of (weak) one-way functions of near-optimal hardness $2^{n-o(n)}$. To the best of our knowledge, this is the first construction of one-way functions of near-optimal hardness based on a natural complexity assumption about a search problem.
- We show that a Weak Perebor Hypothesis for MCSP implies the existence of one-way functions, and establish a partial converse. This is the first unconditional construction of one-way functions from the hardness of MCSP over a natural distribution.
- Finally, we study the average-case hardness of MKtP. We show that it characterizes cryptographic pseudorandomness in one natural regime of parameters, and complexity-theoretic pseudorandomness in another natural regime.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Theory of computation → Problems, reductions and completeness; Theory of computation → Circuit complexity

Keywords and phrases one-way function, meta-complexity, KT complexity, parallel cryptography, randomized encodings

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.35

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2021/057/

Acknowledgements The first author is grateful to Lijie Chen, Mahdi Cheraghchi, and Yanyi Liu for helpful discussions. The second author thanks Yuval Ishai for a useful e-mail discussion. We would like to thank anonymous CCC reviewers for helpful comments that improve the presentation of this paper.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 35; pp. 35:1–35:58
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Backgrounds and Motivation

1.1.1 Meta-Complexity

Let μ be a complexity measure, such as the circuit size of a Boolean function or the time-bounded Kolmogorov complexity of a string. Traditional complexity theory studies the complexity measure on fixed functions, e.g. the AC^0 complexity of the Parity function. In contrast, we study the *meta-complexity* problem associated with μ: given an input function, what is its μ value?

Meta-complexity problems are fundamental to theoretical computer science and have been studied since the very beginning of the discipline [81]. They have connections to several areas of theoretical computer science, including circuit lower bounds, learning, meta-mathematics, average-case complexity, and cryptography. However, our knowledge about them is still very limited compared to our knowledge of other fundamental problems such as the Satisfiability problem.

Some of the basic complexity questions about meta-complexity include:

- Is computing a given measure μ complete for some natural complexity class? For example, is the Minimum Circuit Size Problem (MCSP, [58]) NP-complete?
- Can we show unconditional circuit lower bounds for computing μ, at least for weak circuit classes? Can we distinguish truth tables with $2^{o(n)}$-size circuits from random truth tables by a small $\mathsf{AC}^0[2]$ circuit?
- Is deciding whether μ is at least some parameter k robust to the choice of the parameter k? Let MCSP$[s(n)]$ denote the problem of whether an input function (represented as a truth table) has circuit complexity at most $s(n)$; are MCSP$[2^{n/2}]$ and MCSP$[2^{n/3}]$ computationally equivalent?
- How do low-level definitional issues affect the complexity of μ? Does the complexity of the time-bounded version of Kolmogorov complexity ("K^t") depend on the universal Turing machine that defines it?
- For which pairs of measures μ and μ' can we show that the problem of computing μ reduces to the problem of computing μ'? Can we reduce computing the time-bounded version of Kolmogorov complexity to computing circuit complexity?

There has been much interest in recent years in these questions. While there has been some progress on answering these questions affirmatively for specific measures [4, 48, 3, 6, 72, 43, 34, 40, 46, 47], there are also barriers to understanding these questions better, such as our inability to prove circuit lower bounds [58, 66] and the magnification phenomenon [73, 71, 65, 22]. Many of the above questions such as the NP-completeness of MCSP remain wide open.

1.1.2 Cryptography

A fundamental question in cryptography is whether one-way functions exist. We have been quite successful at basing one-way functions on the hardness of specific problems, such as factoring [75], discrete logarithm [25], and some lattice problems [1]. One problem with this approach, however, is that we have little complexity-theoretic evidence for the hardness of these problems (for example, they are unlikely to be NP-hard). The most compelling evidence for their hardness so far is simply that *we have not been able to find efficient algorithms for them.*

Can we base the existence of one-way functions on firm complexity-theoretic assumptions? A "holy-grail" in this direction would be to construct one-way functions assuming (only) $\mathsf{NP} \not\subseteq \mathsf{BPP}$ [25]. This goal remains elusive, and there are several obstacles to its resolution:

- Unless PH collapses, non-adaptive "black-box" reductions cannot transform worst-case hardness of NP into average-case hardness of NP [17]. As the latter is necessary for one-way functions, this barrier result demonstrates limits of such "black-box" reductions on basing one-way function from worst-case assumptions such as $\mathsf{NP} \not\subseteq \mathsf{BPP}$. For the task of constructing one-way functions (instead of just a hard-on-average problem in NP), stronger barrier results are known [2, 67].
- Even the seemingly easier task of basing one-way functions from *average*-case hardness of NP remains elusive. Indeed, Impagliazzo [50] called a world "Pessiland" where NP is hard on average but one-way functions do not exist. It is not hard to construct a relativized Pessiland [87], therefore a relativization barrier exists even for this "easier" task.

1.1.3 The Liu-Pass Result

Very recently, in a breakthrough result, Liu and Pass [62] showed an *equivalence* between the existence of one-way functions and the *bounded-error average-case hardness* of computing the K^t complexity (the Kolmogorov complexity of a string with respect to a given polynomial time bound t) over the uniform distribution. This result is significant for several reasons.

- From the perspective of cryptography, it establishes the first equivalence between the existence of one-way functions and the average-case complexity of a natural problem over a natural distribution. Such an *equivalence* result bases cryptography on firmer complexity-theoretic foundations.
- From the perspective of meta-complexity, it enables *robustness* results for the complexity of K^t in the average-case setting. Indeed, [62] proved that *approximating* the K^t complexity of a string or *finding* an optimal description for a string are both equivalent to the problem of computing the K^t complexity.
- More generally, such connections suggest the possibility of new and non-trivial *average-case* reductions between natural problems on natural distributions, which is by itself an important goal in average-case complexity theory. Several of the most basic questions in this area remain open: Is random 3-SAT as hard as random 4-SAT (or vice versa)? Is the decision version of Planted Clique as hard as its search version?[1]

Given these motivations, it is natural to ask if the main result of [62] can be extended to other meta-complexity problems. For example, is the average-case hardness of MCSP also equivalent to the existence of one-way functions? There is a "Kolmogorov-version" of circuit complexity, named KT, which is more "fine-grained" than circuit complexity [4]. Maybe this problem is also closely related to the existence of one-way functions? What about Levin's Kt complexity [60]?[2]

[1] The decision version of Planted Clique is to distinguish Erdős-Rényi random graphs from graphs with a planted clique; the search version is to find the planted clique.

[2] See Definition 14 for the precise definitions of K^t, KT, and Kt.

1.2 Our Contributions

We give strong positive answers to the above questions. We show somewhat surprisingly that the average-case hardness of KT complexity is equivalent to the existence of one-way functions computable in *fast parallel time*.[3] For MCSP, we obtain weaker results: exponential hardness of computing circuit size over the uniform distribution implies the existence of one-way functions, and there is a partial converse. Bounded-error average-case complexity of Kt complexity turns out to be equivalent to the existence of one-way functions in one natural setting of parameters (despite the fact that computing Kt in the worst case is EXP-hard [4]), and equivalent to the existence of complexity-theoretic pseudorandom generators in another natural setting of parameters.

We also extend the connection between the hardness of K^t complexity and one-way functions to the high end of the parametric regime – this yields one-way functions of almost optimal hardness from plausible assumptions about the hardness of K^t complexity. We define and motivate the Perebor Hypotheses[4], which are average-case analogues of the Exponential-Time Hypothesis and its variants for meta-complexity problems, stating that there is no better way to solve meta-complexity problems than brute force search. This is a conceptual contribution of this work, and we expect these hypotheses to have further applications to cryptography, average-case complexity, and fine-grained complexity.

We now describe our results in more detail.

1.2.1 Connections between Meta-Complexity and One-Way Functions

Our main result is an equivalence between "parallel cryptography" and the average-case hardness of MKTP:

▶ **Theorem 1** (Main Result; Informal). *There is a one-way function computable in uniform* NC^1 *if and only if* KT *is bounded-error hard on average.*

The class "uniform NC^1" in the above theorem is somewhat arbitrary since [11] proved that the existence of one-way functions in $\oplus\mathsf{L}$ implies the existence of one-way functions in NC^0.[5]

For comparison, Liu and Pass [62] showed an equivalence between ("sequential") cryptography and the average-case hardness of time-bounded Kolmogorov complexity (K^t).

▶ **Theorem 2** (Main Result of [62]). *There is a one-way function if and only if for some polynomial t,* K^t *is bounded-error hard on average.*

Theorem 2 shows that the one-way function defined based on hardness of K^t is a natural *universal* one-way function.[6] Similarly, Theorem 1 shows that the one-way function we define based on the hardness of KT is a natural universal one-way function in NC^1.

[3] Due to a result in [11], the "fast parallel time" here can be interpreted as either NC^0 or NC^1. We also point the reader to Benny Applebaum's book *Cryptography in Constant Parallel Time* [8], which inspired the title of the current paper.

[4] Our terminology is inspired by Trakhtenbrot's survey [81] on work in the former Soviet Union aiming to show that various meta-complexity problems require brute force search to solve. "Perebor" roughly means "by exhaustive search" in Russian.

[5] $\oplus\mathsf{L}$ is the class of problems solvable by a *parity* Turing machine with $O(\log n)$ space. This class contains both NC^1 and L (log-space).

[6] An artificial universal one-way function can be defined by enumerating uniform algorithms and concatenating their outputs [61, 30].

As a corollary, the classical open question of whether polynomial-time computable one-way functions imply one-way functions in NC^0 is equivalent to the question of whether average-case hardness of K^t implies average-case hardness of KT.

Results for MCSP. The KT complexity was defined as a variant of Kolmogorov-complexity that resembles circuit complexity [4]. Therefore, it is natural to ask whether our equivalence also holds for circuit complexity.

It turns out that circuit complexity is less convenient to deal with. Nevertheless, we still proved non-trivial analogues of Theorem 1, as follows:

▶ **Theorem 3** (Informal). *The following are true:*

- *If* MCSP *is exponentially hard on average, then there is a (super-polynomially hard) one-way function.*
- *If there is an exponentially hard weak one-way function in* NC^0, *then* MCSP *is (exponentially) hard on average.*

For the technical difficulties of handling circuit complexity, the reader is referred to Section 6 (and in particular Remark 76).

Results for MKtP. We also observe that the existence of (polynomial-time computable) one-way functions can be characterized by the bounded-error average-case complexity of Kt.

▶ **Theorem 4.** *There is a one-way function if and only if* Kt *is bounded-error hard on average.*

This result may seem surprising as computing Kt is EXP-hard under polynomial-size reductions [4]. This is true even for any oracle that is a *zero-error* heuristic for computing Kt. In contrast, we show that the *bounded-error* average-case complexity of Kt is captured by one-way functions, a notion that seems much "easier" than EXP.

The harder direction in Theorem 4 is to construct a one-way function from hardness of Kt. How could we construct a one-way function from merely a hard problem *in exponential time*? The crucial insight is as follows: For *most* strings $x \in \{0,1\}^n$ whose optimal Kt complexity is witnessed by a machine d and a time bound t where $\mathrm{Kt}(x) = |d| + \log t$, we have $t \le \mathrm{poly}(n)$. We refer the reader to Section 2.1.2 and Section 7 for more details.

Note that Theorem 4 can also be seen as a characterization of cryptographic pseudorandomness, by the known equivalence between one-way functions and cryptographic pseudorandomness [38]. In a different regime of parameters, average-case hardness of Kt turns out to capture the existence of *complexity-theoretic* pseudorandom generators, which are pseudorandom generators with non-trivial seed length computable in *exponential* time. Thus the average-case complexity of a single problem (Kt) can be used to capture both cryptographic pseudorandomness and complexity-theoretic pseudorandomness!

▶ **Theorem 5** (Informal). *For each $\epsilon > 0$, there is a pseudo-random generator from n^ϵ bits to n bits computable in time $2^{n^\epsilon}\mathrm{poly}(n)$ secure against $\mathrm{poly}(n)$ size circuits iff for each $c > 1/2$ there are no polynomial-size circuits solving* Kt *on more than $1 - 2^{-cn}$ fraction of inputs of length n.*

1.2.2 Application in Meta-Complexity: Robustness Theorems

We exploit the connection between MKTP and parallel cryptography to establish more robustness results for meta-complexity. It is known that parallel cryptography is *extremely* robust: L-computable one-way functions exist, if and only if NC^1-computable one-way functions exist, if and only if NC^0-computable one-way functions exist [11]. We define L- and NC^1-variants of K^t complexity, and translate the result in [11] to the following robustness theorem:

▶ **Theorem 6** (Bounded-Error Robustness of Meta-Complexity; Informal). *The following statements are equivalent:*

- KT *is bounded-error hard on average.*
- *For $t_1(n) := n^{10}$, the search version of NC^1-K^{t_1} is bounded-error hard on average.*
- *For $t_2(n) := 5n$, L-K^{t_2} is bounded-error hard on average to approximate, within an additive error of $100 \log n$.*

It is natural to ask whether the above theorem can be interpreted as a *reduction.* Somewhat surprisingly, we show the answer is *yes*! We discover an average-case *reduction* from L-K^t to MKTP, as follows:

▶ **Theorem 7** (Informal). *Let n, t be parameters, $m := \mathrm{poly}(n, t)$. There is a randomized reduction $\mathsf{Red}(x)$ that maps a length-n input to a length-m input, and satisfies the following property:*

- *Given a uniform random input x of length n, $\mathsf{Red}(x)$ produces a uniform random string of length m.*
- *Given a string x such that L-$\mathrm{K}^t(x)$ is small, for every possible randomness used in Red, the* KT *complexity of $\mathsf{Red}(x)$ is also small.*

To the best of our knowledge, this is the first reduction from a variant of K^t complexity to a variant of KT complexity. The only special property of L that we use is that L-computable functions have *perfect randomized encodings* [11]. If polynomial-time computable functions have such perfect randomized encodings, then our techniques imply an average-case reduction from the (standard) K^t complexity to the KT complexity.

We have focused on the *bounded-error* average-case complexity of meta-complexity problems so far. However, Theorem 7 also implies robustness in the *zero-error* regime. Here, let $\mathrm{MKTP}[s]$ be the problem of determining whether the input x satisfies $\mathrm{KT}(x) \le s(|x|)$, and let $\mathrm{MINK}^t[s]$ be the problem of determining whether the input x satisfies $\mathrm{K}^t(x) \le s(|x|)$.

▶ **Theorem 8** (Zero-Error Robustness of Meta-Complexity; Informal). *Among the following items, we have (1) $\iff$ (2), and both items are implied by (3).*

1. *There is a constant $c > 0$ such that NC^1-$\mathrm{MINK}^{t_1}[n - c \log n]$ is zero-error easy on average.*
2. *There is a constant $c > 0$ such that L-$\mathrm{MINK}^{t_2}[n - c \log n]$ is zero-error easy on average.*
3. *There is a constant $c > 0$ such that $\mathrm{MKTP}[n - c \log n]$ is zero-error easy on average.*

1.2.3 Application in Cryptography: Maximally Hard One-Way Functions

How hard can a one-way function be? The standard definition of one-way functions only requires that no polynomial-time adversary inverts a random output except with negligible probability. However, it is conceivable that some one-way function requires $2^n/\mathrm{poly}(n)$ time to invert (say, on a constant fraction of inputs)!

The results of [62] opens up the possibility to *characterize* the hardest one-way functions by the meta-complexity of Kolmogorov complexity. In particular, the existence of maximally hard one-way functions may be equivalent to the "Perebor" hypothesis, i.e. some meta-complexity problem requires brute force to solve.

In this work, we tighten the connection between *weak* one-way functions (for which it is hard to invert a random instance w.p. $1 - 1/\mathrm{poly}(n)$) and the hardness of K^t complexity. We managed to show a very tight result:

▶ **Theorem 9** (Informal). *For every constant $\alpha > 0$, there exists a weak one-way function with hardness $2^{(1-o(1))\alpha n}$ if and only if K^t complexity is hard on average for algorithms of size $2^{(1-o(1))\alpha n}$.*

Note that the two α's in the exponent $(1 - o(1))\alpha n$ are the *same*. That is, we essentially construct the best (weak) one-way functions from the hardness of K^t complexity.

We also attempted to strengthen the relationship between one-way functions in NC^0 and the hardness of KT complexity. Our result is that exponentially-hard weak one-way functions in NC^0 imply exponential hardness of KT.

▶ **Theorem 10** (Informal). *If there is a weak one-way function in NC^0 with hardness $2^{\Omega(n)}$, then* KT *requires $2^{\Omega(n)}$ size to solve on average.*

Finally, we put forward a few Perebor Hypotheses. These hypotheses assert brute-force search is unavoidable for solving meta-complexity problems such as K^t and KT, and are closely related to the maximum hardness of (weak) one-way functions. See Section 5.5 for more details.

1.3 Related Work

There have been several previous works connecting meta-complexity to cryptography. Impagliazzo and Levin [51] show that the existence of one-way functions is equivalent to the hardness of a certain learning task related to time-bounded Kolmogorov complexity. Oliveira and Santhanam [72] show a dichotomy between learnability and cryptographic pseudorandomness in the non-uniform setting: there is a non-trivial non-uniform learner for polynomial-size Boolean circuits iff there is no exponentially secure distribution on functions computable by polynomial-size circuits. Santhanam [76] proves an equivalence between the existence of one-way functions and the non-existence of natural proofs under a certain universality assumption about succinct pseudorandom distributions. We note here that the non-existence of natural proofs is equivalent to the zero-error average-case hardness of MCSP.

None of the above results gives an unconditional equivalence between the average-case hardness of a natural decision problem and the existence of one-way functions. This was finally achieved by Liu and Pass [62], who showed that the weak hardness of $\mathrm{K}^{\mathrm{poly}}$ over the uniform distribution is equivalent to the existence of one-way functions. [62] leaves open whether there are similar connections between one-way functions and the hardness of other meta-complexity problems such as KT and MCSP over the uniform distribution. In this work, we show such connections to *parallel* cryptography, i.e., to the existence of one-way functions in NC^1, which by [11] is equivalent to the existence of one-way functions in NC^0.

There is an extensive literature on parallel cryptography, beginning with the work of [11]. We refer to [8] and [9] for further information.

Our work also relates to average-case meta-complexity, which was first studied explicitly in [43]. [43] essentially observe that the identity reduction trivially reduces μ to μ' over the uniform distribution in a zero-error average-case sense, where μ and μ' are any two meta-complexity measures such that $\mu'(x) \leq \mu(x) \leq |x| + O(\log(|x|))$ for all x. In this work (particularly Sections 4.3 and 4.4), we give several *non-trivial* examples of zero-error and bounded-error average-case reductions between meta-complexity problems.

Concurrent works of [63] and [5]. We now discuss the relationship of our work with the concurrent works of [63] and [5], which overlap in some respects with ours.

Liu and Pass [63] show an equivalence between the bounded-error weak average-case hardness of Kt over the uniform distribution and the existence of one-way functions - this is essentially the same as our Theorem 4. They also show that the *zero-error* average-case hardness of Kt over the uniform distribution is equivalent to $\mathsf{EXP} \neq \mathsf{BPP}$. In contrast, our Theorem 5 gives an equivalence between the bounded-error average-case hardness of Kt over the uniform distribution in a different parametric regime and the worst-case hardness of EXP, where the hardness in each case is with respect to non-uniform adversaries. The somewhat surprising message of both sets of results is the same: a minor variation on an average-case complexity assumption that is equivalent to the worst-case hardness of EXP implies the existence of one-way functions.

[63] also give characterizations of parallel cryptography but they do this using space-bounded Kolmogorov complexity and the conditional version thereof. Their work does not contain any results relating to the hardness of KT or MCSP.

Allender, Cheraghchi, Myrisiotis, Tirumala, and Volkovich [5] relate the average-case hardness of the conditional version of KT complexity over the uniform distribution to the existence of one-way functions. They show that if the conditional version is hard on a polynomial fraction of instances, then one-way functions exist. They also give a weak converse: if one-way functions exist, then the conditional version of KT is hard on an exponential fraction of instances. In contrast, we *characterize* parallel cryptography by the average-case hardness of KT.

1.4 Organization

Section 2 presents some of our main ideas and techniques. Section 3 provides basic definitions and preliminaries.

The equivalence between the existence of NC^0-computable one-way functions and the hardness of KT complexity is proved in Section 4. We prove our robustness results in Section 4.3 and 4.4. In Section 5, we present the tight connection between the hardness of K^t complexity and maximally hard one-way functions. To motivate future study, we put forward a few Perebor Hypotheses in Section 5.5, which are closely related to the existence of maximally-hard one-way functions. The results related to MCSP are proved in Section 6, and the results related to MKtP are proved in Section 7. Finally, we leave a few open questions in Section 8.

2 Intuitions and Techniques

For strings $s_1, s_2, \dots, s_n$, we use $s_1 \circ s_2 \circ \cdots \circ s_n$ to denote their concatenation.

2.1 Parallel Cryptography and the Hardness of KT

Our proof of Theorem 1 builds on [62]. However, it turns out that we need new ideas for both directions of the equivalence.

2.1.1 Hardness of KT from One-Way Functions in NC^0

We first review how Liu and Pass [62] proved that one-way functions imply average-case hardness of K^t.

Any cryptographically-secure PRG G implies *zero-error* hardness of K^t [74, 58, 4]. Roughly speaking, the outputs of G have "non-trivial" K^t complexity, but random strings are likely to have "trivial" K^t complexity.[7] If there is a polynomial-time (zero-error) heuristic for K^t, this heuristic will recognize most random strings as "trivial", but recognize every output of G as "non-trivial". Thus, we can use it as a distinguisher for G, contradicting the security of G.

It is crucial in the above argument that our heuristic does not make mistakes. If the outputs of G are "sparse" and our heuristic has two-sided error, our heuristic could also recognize the outputs of G as "non-trivial". (Here, a PRG G with output length n is sparse if the number of possible outputs of G is significantly smaller than 2^n.) In this case, the heuristic may still be correct on most length-n strings, but fail to distinguish the outputs of G from true random strings.

Why not *make G dense*? This is the core idea of Liu and Pass. In particular, from an arbitrary one-way function f, they constructed a *dense* PRG G,[8] and used G to argue that K^t is bounded-error average-case hard. Roughly speaking, if the outputs of G occupy a $1/\mathrm{poly}(n)$ fraction of $\{0,1\}^n$, then any bounded-error heuristic for K^t with error probability $1/n^{\omega(1)}$ is a distinguisher for G. It follows from the security of G that K^t is bounded-error hard on average.

What about KT? Recall that the KT complexity of a string x is the minimum of $|d| + t$ over programs d and integers t such that x can be generated *implicitly* from d in at most t steps, i.e., the universal machine computes the i-th bit x_i of x correctly in at most t steps with oracle access to d. When we use the above framework to analyze the hardness of KT complexity, there is a problem: the outputs of G might have "trivial" KT complexity.

Let t be the running time of G (which is a large polynomial). Let $\mathit{out} := G(\mathit{seed})$ be any output of G, we can see that $\mathrm{K}^t(\mathit{out})$ is indeed non-trivial, as we can describe seed and the code of G with $|\mathit{seed}| + O(1) < n$ bits. Given this description, we can "decompress" out in $t(n)$ steps by computing G on seed. However, $\mathrm{KT}(\mathit{out})$ is the sum of the description length and the running time, which is $|\mathit{seed}| + O(\log n) + t(n) \gg n$. This is even worse than the trivial description for out whose complexity is $n + O(\log n)$.

One attempt is to pad both the seed and the output by a random string of length $\mathrm{poly}(t(n))$, so that G becomes *sublinear*-time computable. That is, $G'(\mathit{seed} \circ r) = \mathit{out} \circ r$ where r is a long string. Still, we only have $\mathrm{KT}(\mathit{out} \circ r) \le |\mathit{seed}| + |r| + t(n)$, but the trivial upper bound for $\mathrm{KT}(\mathit{out} \circ r)$ is only $|\mathit{out}| + |r|$. If $t(n)$ is larger than the stretch of G (i.e., $|\mathit{out}| - |\mathit{seed}|$), then we do not have non-trivial KT-complexity upper bounds on outputs of G.

This problem is inherent as we need G to be *dense*. Suppose that the number of possible outputs of G is $2^n/\mathrm{poly}(n)$, then there must be an output of G whose *Kolmogorov* complexity is at least $n - O(\log n)$. That is, the seed length of G has to be $n - O(\log n)$, even if we place no restrictions on the complexity of G! Now, if we want the outputs of G to have non-trivial KT complexity, we only have $O(\log n)$ time to compute each output bit of G. Therefore, *G is a PRG in constant parallel time.*[9]

[7] Here, the K^t (or KT) complexity of a length-n string is "non-trivial", if it is at most $n - \Omega(\log n)$. Most length-n strings have complexity at least $n - \Omega(\log n)$; every length-n string has complexity at most $n + O(\log n)$ (justifying the word "trivial").

[8] The input distribution of their PRG is not the uniform distribution, which is different from standard PRGs; see Definition 24. We ignore this difference in the informal exposition.

[9] Due to low-level issues in the computational models, the "constant time" in [8] actually corresponds to $O(\log n)$ time in this paper. See Section 3.1 for details.

We discovered that the (bounded-error) average-case complexity of KT is related to *cryptography in* NC^0. Now it is easy to see that NC^0-computable dense PRGs imply bounded-error hardness of KT complexity. We can construct such a PRG from NC^0-computable one-way functions, as follows.[10] We first use [62] to construct a dense PRG G. This PRG is not necessarily in NC^0, as [62] needs some more complex primitives (e.g. extractors). Nevertheless, we can apply the randomized encodings in [11] to compile G into a PRG in NC^0.

2.1.2 One-Way Functions in NC^0 from Hardness of KT

It is straightforward to construct a one-way function from hardness of KT, using techniques of [62, Section 4]. Roughly speaking, the one-way function f receives two inputs d, t, where d is the description of a machine, and t is a time bound. Let x be the string such that for each $i \in [n]$, x_i is equal to the output bit of $d(i)$ for t steps. We define $f(d,t) := (|d| + t, x)$. An inverter, on input (ℓ, x), is required to find a description of x with complexity at most ℓ, thus it needs to solve MKTP. (All one-way functions in this section are *weak*, meaning they cannot be inverted efficiently on a $1 - 1/\text{poly}(n)$ fraction of inputs.)

There is one problem: f is not in NC^0. By [11], it suffices to construct a one-way function in $\oplus\mathsf{L}$, but f is also not in $\oplus\mathsf{L}$ (unless $\oplus\mathsf{L} = \mathsf{P}$).

Our idea is to only consider *typical* inputs, and throw away the atypical ones. In particular, for most strings x, the values of t in the optimal description of $\text{KT}(x) = |d| + t$ are small. (We have $t = O(\log n)$ for every string x with Kolmogorov complexity at least $n - O(\log n)$.) We call an input *typical* if its value of t is at most $O(\log n)$. If KT is (bounded-error) hard on average, then it is also hard on average conditioned on the input being typical.

Therefore, we place the restriction that $t \le c \log n$ in our one-way function f, where c is a constant depending on the hardness of KT. We can still base the hardness of f on the hardness of KT. More importantly, f is computable in space complexity $O(c \log n)$, and we obtain a one-way function in NC^0 by [11].

2.2 Applebaum-Ishai-Kushilevitz as a Reduction

For any "reasonable" circuit class $\mathcal{C}$, we can use [62] to show that the existence of one-way functions computable in $\mathcal{C}$ is equivalent to the hardness of $\mathcal{C}$-K^t. (The precise definition of $\mathcal{C}$-K^t is beyond the scope of this paper, but NC^1-K^t and L-K^t are defined in Definition 15.) Now, let us review the main results of [11]: $\oplus\mathsf{L}$-computable one-way functions exist if and only if NC^0-computable one-way functions exist. In other words, $\oplus\mathsf{L}$-K^t is hard on average if and only if NC^0-K^t is hard on average![11]

It is natural to ask whether there is a reduction between $\oplus\mathsf{L}$-K^t and NC^0-K^t. It turns out that the answer is *yes*! In this section, we describe this reduction without using the language of one-way functions. This reduction is randomized, reduces any string with non-trivial $\oplus\mathsf{L}$-K^t complexity to a string with non-trivial NC^0-K^t complexity, and reduces a random string to a random string. Although it may not be a worst-case reduction, it establishes non-trivial equivalence results between average-case complexities of $\oplus\mathsf{L}$-K^t and NC^0-K^t.

[10] Note that this is different from [38, 37]. The PRG we construct is dense, but its input distribution is not uniform. In contrast, [38, 37] constructs a PRG (on uniformly random inputs) from an arbitrary one-way function, but the PRG is not necessarily dense.

[11] NC^0 may not be reasonable in the above sense, but the reduction we present in this section is correct.

The property that enables our reduction is *resamplability* [26]. For now, think of "easy" as being NC^0-computable and "hard" as otherwise. A hard function f is *resamplable*, if given an input x and random coins r, there is an easy procedure (the "resampler") that produces a uniform random input of f whose answer is the same as x.

▶ **Example 11.** The parity function $\mathrm{PARITY}(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ is hard (i.e., not computable in NC^0). Given n input bits $x_1, x_2, \dots, x_n$ and $n-1$ random bits $r_1, r_2, \dots, r_{n-1}$, we can produce a uniform random input whose answer is the same as x, as follows:

$$(x_1 \oplus r_1, x_2 \oplus r_1 \oplus r_2, x_3 \oplus r_2 \oplus r_3, x_4 \oplus r_3 \oplus r_4, \dots, x_{n-1} \oplus r_{n-2} \oplus r_{n-1}, x_n \oplus r_{n-1}).$$

Note that the resampler is easy (i.e., in NC^0), thus parity is resamplable.

The reduction. We will use a $\oplus\mathsf{L}$-complete problem named DCMD that is resamplable (see Section 3.7). Our reduction is very simple: given an input $x \in \{0,1\}^n$, we choose a large enough $N = \mathrm{poly}(n)$, and replace every bit x_i by a random length-N instance of DCMD whose answer is x_i. Our reduction outputs the concatenation of these n instances.

Since DCMD is balanced (i.e., the number of 0-instances and 1-instances are the same), our reduction maps a random instance to a random instance.

Now assume that $\oplus\mathsf{L}\text{-}\mathrm{K}^t(x) = n - \gamma$ is non-trivial, and d is a $\oplus\mathsf{L}$ machine of description length $n - \gamma$ that "computes" x. Since DCMD is $\oplus\mathsf{L}$-complete (under NC^0-reductions), for each i, the computation of x_i can be reduced to a DCMD-instance s_i of length N such that $\mathrm{DCMD}(s_i) = x_i$. Moreover, given the description d, we can produce $s_1 \circ s_2 \circ \cdots \circ s_n$ in NC^0.

We use the *resamplability* of DCMD. The resampler for DCMD only uses $N-1$ random bits (which is optimal). Consider the following NC^0 circuit. It receives d and $r_1, r_2, \dots, r_n$ as inputs, where each r_i is a random string of length $N-1$. It computes $s_1, s_2, \dots, s_n$ from d, and for each i, feeds s_i and r_i to the resampler to obtain a uniform random DCMD instance whose answer is the same as s_i. When r_i are random bits, the output distribution of this NC^0 circuit is identical to the distribution of $\mathsf{NC}^0\text{-}\mathrm{K}^t$ instances we reduced x to. Moreover, the $\mathsf{NC}^0\text{-}\mathrm{K}^t$ complexity of *every* string in this distribution is at most $(n-\gamma) + (N-1)n + O(\log n) = Nn - \gamma + O(\log n)$, which is non-trivial.[12]

As a consequence, we also obtain an (average-case) reduction from $\oplus\mathsf{L}\text{-}\mathrm{K}^t$ to KT.

2.3 Tighter Connections

To obtain a tight relationship between hardness of K^t and hardness of weak one-way functions, we optimize the construction from one-way functions to PRGs in [62]. Suppose that given a one-way function f with input length n, we could construct a PRG with output length m'. Then solving K^t on length m' is (roughly) as hard as inverting f on length n. Therefore, we need m' to be as close to n as possible. As the PRG is dense, its output length m' is close to its input length m, thus we only need m to be close to n.

It turns out that the input of the PRG consists of the input of f and the seeds of a few pseudorandom objects.

- One object is an *extractor* $\mathsf{Ext}(\mathcal{X}, r)$ [70, 68], which given a "somewhat random" distribution $\mathcal{X}$ and a truly random seed r, outputs a distribution that is statistically close to the uniform random distribution.

  We use the near-optimal explicit extractors with $O(\log^2 n)$ seed length [36].

[12] The additive factor here is $O(\log n)$ since in our computational model, each memory access requires $\Theta(\log n)$ time. See Section 3.1 for details.

- Another object is a *hardcore function* $\mathsf{HC}(x, r)$ [32]. Let f be a one-way function, x be a random input, and r be a random seed. Given $f(x) \circ r$, it should be infeasible to distinguish between $\mathsf{HC}(x, r)$ and a uniformly random string. Note that $\mathsf{HC}(x, r)$ needs to have multiple output bits; in contrast, a *hardcore predicate* (also defined in [32]) only has one output bit.

  We use the observation, implicit in [82, 79], that any seed-extending "black-box" pseudorandom generator is a good hardcore function. We use the *direct product* generator [42, 41] as our hardcore function, which has $O(\log^2 n)$ seed length, and very small "advice complexity." The advice complexity turns out to be related to the overhead of our reduction.

There is another problem: [62] needs a *strong* one-way function to start with, but we only have a *weak* one-way function. (A strong one-way function is infeasible to invert on *almost every* input, but a weak one-way function is only infeasible to invert on *a non-trivial fraction* of inputs.) Yao [92] showed how to "amplify" a weak one-way function to a strong one-way function, but the overhead of this procedure is too large. In particular, Yao's hardness amplification does not preserve *exponential* hardness, and it is open whether exponentially-hard weak one-way functions imply exponentially-hard strong one-way functions.

Our idea is to use *Impagliazzo's hardcore lemma* [49] instead. The hardcore lemma states that for any weak one-way function f, there is a "hardcore" distribution on which f becomes a strong one-way function. We (and [62]; see Footnote 8) allow the input distribution of our PRG to be *arbitrary*, as long as the output distribution is pseudorandom. Such "PRGs" still imply hardness of K^t. The hardcore lemma has small complexity overhead, which allows us to prove tight results.

Now, from a weak one-way function of input length n, we can construct a PRG with output length $n+O(\log^2 n)$. This construction allows us to transform the hardness of one-way function to the hardness of K^t at almost no cost.

Tighter connections between MKTP and one-way functions in NC^0. Here, the problem becomes to construct NC^0-computable PRGs from NC^0-computable one-way functions. We use a construction of universal hash functions in NC^0 with linear seed length by Applebaum [10]. Such hash functions are both good extractors (by the leftover hash lemma) and good hardcore functions (proved in [15, 44]). As the hash functions require linear seed length, from an NC^0-computable one-way function with input length n, we obtain an NC^0-computable PRG with output length $O(n)$. It follows that if the one-way function is hard against $2^{\Omega(n)}$-size adversaries, then MKTP is also hard against $2^{\Omega(n)}$-size algorithms.

2.4 MCSP-Related Results

One-way functions from hardness of MCSP. We use the straightforward construction: our one-way function receives a circuit C, and outputs $|C|$ and $tt(C)$, where $|C|$ is the size of C and $tt(C)$ is the truth table of C. The inverter, on input (s, tt), is required to find a size-s circuit whose truth table is tt, thus needs to solve MCSP.

One problem with this construction is that if we sample a uniform circuit (according to some distribution), the induced distribution over truth tables may not be uniform. In the case of K^t (and KT), we can show that for every string of length n, its optimal description is sampled (in the one-way function experiment) w.p. at least $2^{-n}/\mathrm{poly}(n)$, therefore we can "transfer" the hardness of K^t over a random truth table to the hardness of inverting the one-way function over a random description.

Using the best bounds on the maximum circuit complexity of n-bit Boolean functions [28], we can still prove that for every truth table of length N, its optimal circuit is sampled w.p. at least $2^{-N}/2^{\eta}$, where $\eta < o(N)$. This means that starting from *exponential* hardness of MCSP, we can still obtain non-trivial one-way functions.

We conjecture that hardness of MCSP actually implies one-way functions in NC^0; see Remark 76 for details.

Hardness of MCSP from one-way functions in NC^0. To argue about the hardness of MCSP, we need a PRG whose outputs have non-trivial circuit complexity. As before, we use the hash functions in [10] to construct an exponentially-hard PRG. We would like to argue that all outputs of the PRG have non-trivial circuit complexity. In order to do this, we use the mass production theorem of Uhlig [83, 84] to generate a circuit of size $(1+o(1))2^n/n$ that evaluates a given function on *multiple* inputs. (If our PRG has locality d, i.e., each output bit depends on d input bits, then we need a size-$(1+o(1))2^n/n$ circuit that evaluates d inputs in parallel.) However, Uhlig's theorem only gives us non-trivial circuit size if our PRG has *linear* stretch, i.e., stretch ϵn for some constant $\epsilon > 0$. This is why we need the hardness of the one-way function in our assumption to be at least $\mathrm{poly}(2^{\epsilon n})$.

2.5 Using Hardness of Kt to Capture Cryptographic and Complexity-Theoretic Pseudorandomness

To show Theorem 4, we use ideas similar to those in Section 2.1.2. Suppose we try to define a one-way function by computing the string corresponding to an optimal description with respect to Kt complexity. An obvious issue is that such strings might require exponential time to compute, while the one-way function needs to be evaluated efficiently. However, we observe that *typical* inputs only require polynomial time to generate from their optimal descriptions. Here, the typical inputs are those with Kolmogorov complexity $n - O(\log n)$. In their optimal descriptions $\mathrm{Kt}(x) = |d| + \log t$, we have $t \le \mathrm{poly}(n)$. Our one-way function receives two inputs d, t, where d is the description of a machine, and $t \le \mathrm{poly}(n)$ is a time bound. We simply simulate the machine d for t steps and output what it outputs. The proof that this gives a one-way function is closely analogous to the proof of the reverse implication in Theorem 1. The proof that one-way functions imply the average-case hardness of Kt complexity mimics the proof of the corresponding implication in Theorem 2, since the outputs of a cryptographic PRG with stretch $\lambda \log n$ have non-trivial Kt complexity when λ is large enough compared to the time required to compute the PRG.

To show Theorem 5, we use the Nisan-Wigderson generator [69] in a way similar to how it is used by [4] to show that Kt is complete for exponential time under polynomial-size reductions. The interesting direction is to show that the Nisan-Wigderson generator implies the average-case hardness of Kt for the range of parameters in the statement of Theorem 5. We use the fact that the Nisan-Wigderson generator can be made seed-extending without loss of generality. We truncate the output of the generator so that the stretch is $(1+\epsilon)n$ for some small $\epsilon > 0$ – this implies that the outputs of the generator on all seeds have non-trivial Kt complexity. Since the generator is seed-extending, the output has high entropy, hence a strong enough average-case algorithm for Kt can distinguish random strings (which have trivial Kt complexity) from the outputs of the PRG. Here we take advantage of the stretch being *small* rather than *large*: this gives us better parameters for our average-case hardness result.

3 Preliminaries

We use $\mathcal{U}_n$ to denote the uniform distribution over length-n binary strings. For a distribution $\mathcal{D}$, we use $\mathbf{x} \leftarrow \mathcal{D}$ to denote that $\mathbf{x}$ is a random variable drawn from $\mathcal{D}$. A function $\mathrm{negl} : \mathbb{N} \to [0, 1]$ is *negligible* if for every constant c, $\mathrm{negl}(n) \leq 1/n^c$ for large enough integers n.

Let $D : \{0, 1\}^n \to \{0, 1\}$ be a function, X and Y be two random variables over $\{0, 1\}^n$. For $\epsilon > 0$, we say D *ϵ-distinguishes* X from Y if

$$|\Pr[D(X) = 1] - \Pr[D(Y) = 1]| \geq \epsilon.$$

Otherwise we say X and Y are *ϵ-indistinguishable* by D.

We often consider ensemble of functions in this paper. For example, a function $f : \{0, 1\}^\star$ to $\{0, 1\}^\star$ can be interpreted as an ensemble $f = \{f_n : \{0, 1\}^n \to \{0, 1\}^\star\}$, and each f_n is the *n-th slice* of f. Similarly, we also consider ensemble of distributions $\mathcal{D} = \{\mathcal{D}_n\}$ as input distributions for a function f, where each $\mathcal{D}_n$ is a distribution over $\{0, 1\}^n$.

3.1 Computational Model and Uniformity

We need a computational model with random access to inputs. We consider a Turing machine that accesses the length-n input x via an "address" tape and a length-$O(1)$ "answer" tape. Whenever the machine enters a particular "address" state, let i be the binary number written in the address tape. After one step, the content of the answer tape becomes x_i, and the address tape is cleared. (In other words, the Turing machine treats x as the truth table of an *oracle*.)

We also assume that the address tape has length $\lceil \log n \rceil$. In particular, there are two special markers at the address tape, and there are $\lceil \log n \rceil$ cells strictly between them. The machine can only modify this portion of $\lceil \log n \rceil$ cells; the rest of the address tape is read-only. For sub-linear time Turing machines, this can be viewed as a mechanism to provide information about n (i.e., the length of x; up to a factor of 2). We also require that whenever the machine enters the "address" state, all the $\lceil \log n \rceil$ cells between the two markers are non-empty, so we can interpret the concatenation of these cells as a (binary) address.

Every bit operation takes one step. Therefore, it takes $\Theta(\log n)$ time to write down an address. Note that we clear the address tape after each access, which means when we access another input bit, we have to spend another $\Theta(\log n)$ time to write down the address from scratch. This definition ensures that in $O(\log n)$ time we can only access a constant number of input bits, so $\mathsf{DLOGTIME}$ becomes a natural uniform analogue of NC^0.

In addition to the address tape and the answer tape, we also have a constant number of work tapes. In the case that our Turing machine computes a multi-output function f, we also provide an input tape that contains an index i (note that our *real* input is the "oracle" x), which means our Turing machine outputs the i-th bit of $f(x)$. We use $M^x(i)$ to denote the output of the machine M on input i, given oracle access to the string x. To measure the space complexity of our Turing machine, we assume the input tape is read-only and we only count the total length of work tapes.

▶ **Definition 12.** *Let $c > 0$ be a constant, $p(\cdot)$ be a polynomial, and $F = \{F_n : \{0, 1\}^n \to \{0, 1\}^{p(n)}\}$ be an ensemble of functions. We say $F \in \mathsf{TIME}[c \log n]$ if there is a Turing machine M with running time $c \log n$ such that, for every $x \in \{0, 1\}^n$ and $1 \leq i \leq p(n)$, $M^x(n, i)$ outputs the i-th bit of $F_n(x)$.*

Let $\mathsf{DLOGTIME} = \bigcup_{c \geq 1} \mathsf{TIME}[c \log n]$.

▶ **Definition 13.** *Let $c > 0$ be a constant, $p(\cdot)$ be a polynomial, and $F = \{F_n : \{0,1\}^n \to \{0,1\}^{p(n)}\}$ be an ensemble of functions.*

We say F is in $\mathsf{ATIME}[c \log n]$, if there is an alternating *Turing machine M of $O(\log n)$ running time such that, for every $x \in \{0,1\}^n$, $1 \le i \le p(n)$, and $b \in \{0,1,\star\}$, $M^x(n,i,b) = 1$ if the i-th bit of $F_n(x)$ is b.*

We say F is in $\mathsf{SPACE}[c \log n]$, if there is a Turing machine M of space complexity $c \log n$ that satisfies the above requirement. We say F is in uniform $\oplus\mathsf{SPACE}[c \log n]$, if there is a parity *Turing machine M of space complexity $c \log n$ that satisfies the above requirement.*

Let $\mathsf{ALOGTIME} = \mathsf{NC}^1 = \bigcup_{c \ge 1} \mathsf{ATIME}[c \log n]$. (That is, in this paper, we use $\mathsf{ALOGTIME}$ and NC^1 interchangeably.) We also define $\mathsf{L} = \bigcup_{c \ge 1} \mathsf{SPACE}[c \log n]$, and $\oplus\mathsf{L} = \bigcup_{c \ge 1} \oplus\mathsf{SPACE}[c \log n]$.

For the readers not familiar with parity Turing machines, we provide an alternative definition of $\oplus\mathsf{L}$ by its complete problems; see Section 3.7.

3.2 Resource-Bounded Kolmogorov Complexity

We define some variants of resource-bounded Kolmogorov complexity. In particular, we define the plain Kolmogorov complexity K, the KT complexity [4], the time-bounded Kolmogorov complexity K^t [59], and Levin's Kt complexity [60]. Then we define the NC^1- and L-versions of K^t.

▶ **Definition 14.** *Let U be a Turing machine, x be a string. Artificially let $x_{|x|+1} = \star$.*

- $\mathrm{K}_U(x)$ *is the minimum $|d|$ over the description $d \in \{0,1\}^\star$, such that for every $1 \le i \le |x|+1$ and $b \in \{0,1,\star\}$, $U^d(i,b)$ accepts if and only if $x_i = b$.*
- $\mathrm{KT}_U(x)$ *is the minimum value of $|d| + t$ over the pairs (d,t), such that for every $1 \le i \le |x|+1$ and $b \in \{0,1,\star\}$, $U^d(i,b)$ accepts in t steps if and only if $x_i = b$.*
- $\mathrm{Kt}_U(x)$ *is the minimum value of $|d| + \log t$ over the pairs (d,t), such that for every $1 \le i \le |x|+1$ and $b \in \{0,1,\star\}$, $U^d(i,b)$ accepts in t steps if and only if $x_i = b$.*
- *Let $t : \mathbb{N} \to \mathbb{N}$ be a resource bound. $\mathrm{K}^t_U(x)$ is the minimum value of $|d|$ such that for every $1 \le i \le |x|+1$ and $b \in \{0,1,\star\}$, $U^d(i,b)$ accepts in $t(|x|)$ steps if and only if $x_i = b$.*

▶ **Definition 15.** *Let U_a be an alternating Turing machine, and U_s be a (space-bounded) Turing machine. Let x be a string and artificially let $x_{|x|+1} = \star$.*

- *Let $t : \mathbb{N} \to \mathbb{N}$ be a resource bound. $\mathsf{NC}^1\text{-}\mathrm{K}^t_{U_a}(x)$ is the minimum value of $|d|$ such that for every $1 \le i \le |x|+1$ and $b \in \{0,1,\star\}$, $U_a^d(i,b)$ accepts in alternating time $\log t(|x|)$ if and only if $x_i = b$.*
- *Let $t : \mathbb{N} \to \mathbb{N}$ be a resource bound. $\mathsf{L}\text{-}\mathrm{K}^t_{U_s}(x)$ is the minimum value of $|d|$ such that for every $1 \le i \le |x|+1$ and $b \in \{0,1,\star\}$, $U_s^d(i,b)$ accepts in space $\log t(|x|)$ if and only if $x_i = b$.*

Our results hold for every efficient enough universal Turing machine U. Therefore, in this paper, we drop the subscript U and simply write KT, K^t, etc.

We also define the *circuit complexity* of a truth table:

▶ **Definition 16.** *Let $N = 2^n$, $tt \in \{0,1\}^N$ be a truth table that corresponds to a function $f : \{0,1\}^n \to \{0,1\}$. We define $\mathsf{Size}(tt)$ as the size (number of gates) of the smallest circuit that computes f.*

Given a complexity measure μ, the Minimum μ Problem is the language $\{(x, 1^k) : \mu(x) \le k\}$. In particular:

▶ **Definition 17.** *We define the following problems:*

- *(Minimum* KT *Problem)* $\mathrm{MKTP} := \{(x, 1^k) : \mathrm{KT}(x) \le k\}$*.*
- *(Minimum Time-Bounded Kolmogorov Complexity Problem)* $\mathrm{MINKT} := \{(x, 1^t, 1^s) : \mathrm{K}^t(x) \le s\}$*.*
- *(Minimum Circuit Size Problem)* $\mathrm{MCSP} := \{(tt, 1^s) : \mathsf{Size}(tt) \le s\}$*.*

There are natural search versions for the problems above. The search version for MKTP is to find an optimal description d for x, such that x can be generated from d implicitly in time at most $k - |d|$. The search version for MINKT is to find an optimal description d of size at most s for x such that x can be generated from d in time at most t. The search version for MCSP is to find a circuit of size at most s for the Boolean function whose truth table is tt.

We need a "trivial" upper bound on these complexity measures. We only state the upper bound for KT complexity.

▶ **Fact 18** ([4, Proposition 13])**.** *There is an absolute constant* $c' > 0$ *such that* $\mathrm{KT}(x) \le |x| + c' \log |x|$ *for every string* x*.*

We need the fact that most strings have large Kolmogorov complexity.

▶ **Fact 19.** *Let* n *be an integer,* $s \le n - 1$*, then*

$$\Pr_{\mathbf{x} \leftarrow \mathcal{U}_n}[\mathrm{K}(\mathbf{x}) \le s] \le 2^{-(n-s-1)}.$$

Proof Sketch. The number of strings x such that $\mathrm{K}(x) \le s$ is at most $\sum_{i=0}^{s} 2^i = 2^{s+1} - 1$. ◀

3.3 Basic Information Theory

We also need some basic concepts in information theory. The *Shannon entropy* of a random variable X, denoted as $\mathrm{H}(X)$, is defined as

$$\mathrm{H}(X) := \mathop{\mathbb{E}}_{\mathbf{x} \leftarrow X}[-\log \Pr[X = \mathbf{x}]].$$

The *min-entropy* of a random variable X, denoted as $\mathrm{H}_\infty(X)$, is the largest real number k such that for every x in the support of X,

$$\Pr[X = x] \le 2^{-k}.$$

Let X, Y be two random variables defined over a set $\mathcal{S}$. The *statistical distance* between X and Y, denoted as $\mathsf{SD}(X, Y)$, is defined as

$$\mathsf{SD}(X, Y) := \frac{1}{2} \sum_{s \in \mathcal{S}} |\Pr[X = s] - \Pr[Y = s]|.$$

An equivalent definition is as follows: $\mathsf{SD}(X, Y)$ is the maximum value of ϵ such that there is a (possibly unbounded) distinguisher $\mathcal{D}$ that ϵ-distinguishes X from Y:

$$\mathsf{SD}(X, Y) := \max_{\mathcal{D}: \mathcal{S} \to \{0,1\}} |\Pr[\mathcal{D}(X) = 1] - \Pr[\mathcal{D}(Y) = 1]|.$$

3.4 Bounded-Error Average-Case Hardness

We define the (bounded-error) average-case hardness of a function f. (Think of $f = \mathrm{KT}$ or K^t.) In the cryptographic setting, we require that any algorithm with an *arbitrary polynomial* run time fails to solve a *fixed-polynomial* fraction of inputs.

▶ **Definition 20.** *Let $f : \{0,1\}^\star \to \mathbb{N}$ be a function.*

- *We say that f is* (bounded-error) hard on average *if the following is true. There is a constant $c > 0$ such that for every PPT*[13] *machine $\mathcal{A}$ and every large enough input length n,*

$$\Pr_{\mathbf{x}\leftarrow\mathcal{U}_n}[\mathcal{A}(\mathbf{x}) = f(\mathbf{x})] \le 1 - \frac{1}{n^c}.$$

- *Let d be a constant. We say that f is (bounded-error) hard on average to $(d\log n)$-*approximate *if the following is true. There is a constant $c > 0$ such that for every PPT machine $\mathcal{A}$ and every large enough input length n,*

$$\Pr_{\mathbf{x}\leftarrow\mathcal{U}_n}[f(\mathbf{x}) \le \mathcal{A}(\mathbf{x}) \le f(\mathbf{x}) + d\log n] \le 1 - \frac{1}{n^c}.$$

3.5 One-Way Functions

We recall the standard definition of one-way functions and weak one-way functions.

▶ **Definition 21** (One-Way Functions). *Let $f : \{0,1\}^\star \to \{0,1\}^\star$ be a polynomial-time computable function. We say f is a* one-way function *if for every PPT adversary $\mathcal{A}$, it inverts a random output of f with negligible probability. That is, for every $n \in \mathbb{N}$,*

$$\Pr_{\mathbf{x}\leftarrow\mathcal{U}_n}[\mathcal{A}(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))] \le \mathrm{negl}(n).$$

One-way functions are also called *strong* one-way functions, as no PPT adversary could invert it non-trivially. We also consider *weak* one-way functions, where no PPT adversary could invert it on a $1 - \mathrm{negl}(n)$ fraction of inputs.

▶ **Definition 22** (Weak One-Way Functions). *Let $f : \{0,1\}^\star \to \{0,1\}^\star$ be a polynomial-time computable function. We say f is a* weak one-way function *if there is a polynomial $p(\cdot)$ such that the following holds. For every PPT adversary $\mathcal{A}$, it inverts a random output of f with probability at most $1 - 1/p(n)$. That is, for every $n \in \mathbb{N}$,*

$$\Pr_{\mathbf{x}\leftarrow\mathcal{U}_n}[\mathcal{A}(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))] \le 1 - \frac{1}{p(n)}.$$

By a standard padding trick (see e.g., [30]), we can assume that (weak or strong) one-way functions are *length-preserving*, i.e. for every input $x \in \{0,1\}^\star$, $|f(x)| = |x|$. In this paper, we will implicitly assume that *every one-way function is length-preserving*.

Yao showed that every weak one-way function can be *amplified* into a strong one-way function.

▶ **Theorem 23** ([92, 30]). *If there exists a weak one-way function, then there exists a strong one-way function.*

In particular, let f be a weak one-way function. Then there is a polynomial $k(\cdot)$, such that the following function f^k is a strong one-way function.

$$f^k(x_1, x_2, \ldots, x_{k(n)}) = f(x_1) \circ f(x_2) \circ \cdots \circ f(x_{k(n)}),$$

where $x_1, x_2, \ldots, x_{k(n)}$ are length-n inputs.

[13] PPT stands for probabilistic polynomial-time.

3.6 Conditionally Secure Entropy-Preserving PRGs

Here we define conditionally secure entropy-preserving PRGs (condEP-PRGs), introduced in [62].

A *pseudorandom generator*, according to the standard definition, is a polynomial-time computable function $G : \{0,1\}^n \to \{0,1\}^m$ (where $m > n$), such that $G(\mathcal{U}_n)$ and $\mathcal{U}_m$ are computationally indistinguishable. Compared with standard PRGs, a *condEP-PRG* $G : \{0,1\}^n \to \{0,1\}^m$ has three differences:

- The input distribution of G is not $\mathcal{U}_n$. Instead, it is the uniform distribution over a subset of inputs $\mathcal{E}_n$, called the *condition*. (We will use $\mathcal{E}_n$ to denote both the subset and the uniform distribution over this subset.)
- G is *entropy-preserving*, meaning that $G(\mathcal{E}_n)$ has large (information-theoretic) entropy. (Note that $\log|\mathcal{E}_n| \le n \le m$. As a consequence, $\log|\mathcal{E}_n|$ cannot be too small compared to m.)
- Finally, G only $(1/p(n))$-fools PPT adversaries for a fixed polynomial $p(\cdot)$. For comparison, a standard PRG is required to $(1/p(n))$-fool PPT adversaries *for every polynomial* $p(\cdot)$. This difference is mostly technical.

▶ **Definition 24** (Conditionally Secure Entropy-Preserving PRG, abbr. condEP-PRG, [62]). *Let $\gamma > 0$ be a constant, and $p(\cdot)$ be a polynomial. Consider a polynomial-time computable ensemble of functions $G = \{G_n : \{0,1\}^n \to \{0,1\}^{n+\gamma\log n}\}$. We say G is a* condEP-PRG*, if there is a family of subsets $\mathcal{E} = \{\mathcal{E}_n \subseteq \{0,1\}^n\}$ (called the "events" or "conditions"), such that the following are true.*

1. *(Pseudorandomness) $G_n(\mathcal{E}_n)$ is $(1/p(n))$-indistinguishable from $\mathcal{U}_{n+\gamma\log n}$ by PPT adversaries. That is, for every PPT $\mathcal{A}$ and every integer n,*

$$\left|\Pr_{\mathbf{x}\leftarrow\mathcal{U}_{n+\gamma\log n}}[\mathcal{A}(\mathbf{x})=1] - \Pr_{\mathbf{x}\leftarrow G_n(\mathcal{E}_n)}[\mathcal{A}(\mathbf{x})=1]\right| < 1/p(n).$$

2. *(Entropy-Preservation) There is a constant d such that for every large enough n, $\mathrm{H}(G_n(\mathcal{E}_n)) \ge n - d\log n$.*

We say the stretch *of G is $\gamma\log n$, and the* security *of G is $1/p(n)$.*

▶ **Theorem 25** ([62]). *There is a function* EP-PRG *computable in* ALOGTIME*, such that the following holds. For any one-way function $f : \{0,1\}^\star \to \{0,1\}^\star$ and any constant $\gamma > 0$, let $G(x,z) = \mathsf{EP\text{-}PRG}(\gamma, x, f(x), z)$, then G is a condEP-PRG with stretch $\gamma\log n$ and security $1/n^\gamma$.*

▶ Remark 26. It is important that the machine EP-PRG is fixed and does not depend on the constant γ. Suppose there is an absolute constant $c > 0$ such that for every $\gamma > 0$, there is a PRG G_γ that runs in $\mathsf{TIME}[c\log n]$ and stretches n bits into $n + \gamma\log n$ bits. The outputs of G_γ will always have KT complexity at most $n + c\log n + O(1) < n + \gamma\log n$, hence a heuristic for MKTP can always distinguish the outputs of G_γ from truly random strings. It follows that we can use such G_γ to argue about the hardness of MKTP. On the other hand, if the time complexity of G_γ depends on γ, it does not necessarily imply any hardness of MKTP.

3.7 Complete Problems for ⊕L

We introduce the ⊕L-complete problems, called Connected Matrix Determinant (CMD) and Decomposed Connected Matrix Determinant (DCMD), that will be crucial to us. Originally motivated by secure multi-party computation [54, 55], these problems have found surprisingly many applications in cryptography and complexity theory [11, 33, 26, 43, 23].

Let n be any integer, define $\ell_{\mathrm{CMD}}(n) := n(n+1)/2$ and $\ell_{\mathrm{DCMD}}(n) := n^3(n+1)/2$.

▶ **Definition 27** (See e.g., [23]). *An instance of* CMD *is an $n \times n$ matrix over* GF(2) *where the main diagonal and above may contain either* 0 *or* 1*, the second diagonal (i.e., the one below the main diagonal) contains* 1*, and other entries are* 0*. In other words, the matrix is of the following form (where $*$ represents any element in* GF(2)*):*

$$\begin{pmatrix} * & * & * & \cdots & * & * \\ 1 & * & * & \cdots & * & * \\ 0 & 1 & * & \cdots & * & * \\ 0 & 0 & 1 & \cdots & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & * \end{pmatrix}.$$

The instance is an $(n(n+1)/2)$-bit string specifying elements on and above the main diagonal. We define $x \in \mathrm{CMD}$ if and only if the determinant (over GF(2)*) of the matrix corresponding to x is* 1*.*

An instance of DCMD *is a string of length $n^3(n+1)/2$. For an input x, $\mathrm{DCMD}(x)$ is computed as follows: we partition x into blocks of length n^2, let $y_i (1 \le i \le n(n+1)/2)$ be the parity of the i-th block, and define $\mathrm{DCMD}(x) := \mathrm{CMD}(y_1 \circ y_2 \circ \cdots \circ y_{n(n+1)/2})$.*

The precise definitions of CMD and DCMD are not important here, but we need the following important facts about them.

▶ **Theorem 28** ([11]). *Let n be an integer. There is a function $P_{\mathrm{CMD}} : \{0,1\}^{\ell_{\mathrm{CMD}}(n)} \times \{0,1\}^{\ell_{\mathrm{DCMD}}(n)-1} \to \{0,1\}^{\ell_{\mathrm{DCMD}}(n)}$, computable in* DLOGTIME*, such that the following hold. For any input $x \in \{0,1\}^{\ell_{\mathrm{CMD}}(n)}$, the distribution of $P_{\mathrm{CMD}}(x, \mathcal{U}_{\ell_{\mathrm{DCMD}}(n)-1})$ is equal to the uniform distribution over $\{y : y \in \{0,1\}^{\ell_{\mathrm{DCMD}}(n)} : \mathrm{DCMD}(y) = \mathrm{CMD}(x)\}$.*

Note that P_{CMD} only uses $\ell_{\mathrm{DCMD}}(n) - 1$ random bits, which is optimal. It also implies:

▶ **Corollary 29.** DCMD *is* balanced*. In other words, for every integer n, the number of Yes instances and No instances of* DCMD *on input length $\ell_{\mathrm{DCMD}}(n)$ are the same.*

Proof. Fix any Yes instance $x \in \{0,1\}^n$ of CMD, then $\{P_{\mathrm{CMD}}(x, r) : r \in \{0,1\}^{\ell_{\mathrm{DCMD}}(n)-1}\}$ contains every Yes instance of DCMD. It follows that there are at most $2^{\ell_{\mathrm{DCMD}}(n)-1}$ Yes instances of DCMD on input length $\ell_{\mathrm{DCMD}}(n)$. The same upper bound can also be obtained for No instances. Since there are $2^{\ell_{\mathrm{DCMD}}(n)}$ strings of length $\ell_{\mathrm{DCMD}}(n)$, there must be exactly $2^{\ell_{\mathrm{DCMD}}(n)-1}$ Yes instances and exactly $2^{\ell_{\mathrm{DCMD}}(n)-1}$ No instances of length $\ell_{\mathrm{DCMD}}(n)$. ◀

▶ **Theorem 30** ([54, 55]). CMD *is $\oplus$L-complete under projections.*[14]

In other words, a language L is in $\oplus$L if and only if there is a polynomial $t(\cdot)$ and a DLOGTIME*-computable projection $p : \{0,1\}^n \to \{0,1\}^{\ell_{\mathrm{CMD}}(t(n))}$, such that for every input $x \in \{0,1\}^n$, $x \in L$ if and only if $\mathrm{CMD}(p(x)) = 1$.*

▶ Remark 31. A proof of Theorem 30 can be found in [23, Section B.1]. However, the proof in [23] does not show that the projections are DLOGTIME-uniform. In particular, the reduction needs to calculate the topological order of the underlying (parity) branching program ($\sigma_1, \sigma_2, \ldots, \sigma_m$ in [23, Section B.1]), which may not be computable in DLOGTIME.

[14] A *projection* is a (multi-output) function where each output bit either is a constant, or only depends on one input bit.

We can fix this issue by adding a clock to the log-space Turing machine; a state of the Turing machine appears earlier in the topological order if its clock value is smaller. Equivalently, let $G = (V, E)$ be the old branching program. The new branching program G^{new} has a vertex (i, v) for every $0 \le i \le |V|$ and $v \in V$, and has edges from (i, u) to $(i+1, v)$ for every edge $(u, v) \in G$ and every $0 \le i < |V|$. Let $V = \{v_1, \dots, v_n\}$, then

$$(0, v_1), \dots, (0, v_n), (1, v_1), \dots, (1, v_n), \dots, (|V|, v_1), \dots, (|V|, v_n)$$

is a valid topological ordering of G^{new}. Now we can use [23, Section B.1] to reduce the computation of G^{new} to CMD by a DLOGTIME-uniform projection.

We would like to thank Yanyi Liu for pointing out this issue.

Theorem 28 and 30 implies the following beautiful result in [11].

▶ **Theorem 32** ([11]). *Suppose there is a one-way function computable in* $\oplus$L. *Then there is a one-way function computable in* DLOGTIME.

Proof Sketch. Let f be a one-way function in $\oplus$L. There is a DLOGTIME-computable function $p(\cdot, i)$ that maps n input bits to $\mathrm{poly}(n)$ output bits, such that for every integer i and every string x, the i-th output bit of $f(x)$ is $\mathrm{CMD}(p(x, i))$. Consider the following function:

$$g(x, y, i) := P_{\mathrm{CMD}}(p(x, i), y).$$

It turns out that the function $g(x, y) = g(x, y, 1) \circ \cdots \circ g(x, y, n)$ is still one-way. ◀

4 KT Complexity and Parallel Cryptography

In this section, we characterize the existence of one-way functions in DLOGTIME by the average-case hardness of MKTP. Recall that the seminal work of [11] showed that the existence of one-way functions in DLOGTIME is also equivalent to the existence of one-way functions in uniform NC^1, L, or $\oplus$L.

▶ **Theorem 1** (Main Result; Informal). *There is a one-way function computable in uniform* NC^1 *if and only if* KT *is bounded-error hard on average.*

4.1 One-Way Functions in NC^0 from Hardness of MKTP

▶ **Theorem 33.** *Suppose that the search version of* KT *is bounded-error hard on average. Then there is a one-way function computable in* DLOGTIME.

Proof. We show that there is a weak one-way function computable in logarithmic space. Then by Theorem 23, there is a one-way function in logarithmic space, and by Theorem 32, there is a one-way function in DLOGTIME.

Suppose KT is bounded-error hard on average. By Definition 20, there is a constant $c > 0$ such that for every PPT algorithm $\mathcal{A}$ and every large enough n, the probability that $\mathcal{A}$ solves the search version of KT on a random length-n input is at most $1 - 1/n^c$.

For a string x, we define $t(x)$ to be the parameter t in the definition of $\mathrm{KT}(x)$ (Definition 14). Formally, $t(x)$ is the smallest integer t such that there is a description d of length $\mathrm{KT}(x) - t$, such that for every $1 \le i \le |x|$ and $b \in \{0, 1, \star\}$, $U^d(i, b)$ accepts in t steps if and only if $x_i = b$. We can see that most strings have small $t(x)$. In what follows, let c_1 be the absolute constant in Fact 18, such that for every $x \in \{0,1\}^n$, $\mathrm{KT}(x) \le |x| + c_1 \log |x|$.

▷ Claim 34. For all but an $1/n^{c+1}$ fraction of strings $x \in \{0,1\}^n$, we have $t(x) \leq (c + c_1 + 2) \log n$.

Proof of Claim 34. By Fact 18, for every $x \in \{0,1\}^n$, $\mathrm{KT}(x) \leq n + c_1 \log n$. By Fact 19, all but a $1/n^{c+1}$ fraction of strings $x \in \{0,1\}^n$ satisfies that $\mathrm{K}(x) > n - (c+1)\log n - 1$. For such strings x, we have $t(x) \leq \mathrm{KT}(x) - \mathrm{K}(x) \leq (c + c_1 + 2)\log n$. ◁

For convenience, we say a pair (d,t) *outputs* the string x, if (d,t) is a valid "witness" for $\mathrm{KT}(x)$, i.e. for every $1 \leq i \leq |x| + 1$ and $b \in \{0,1,\star\}$, $U^d(i,b)$ accepts in time t if and only if $x_i = b$. Let $\mathsf{Output}(d,t)$ be the (unique) string that (d,t) outputs; if (d,t) does not output any (finite) string, let $\mathsf{Output}(d,t) = \perp$.

We define a weak one-way function f as follows.

Algorithm 1 Weak OWF in L from Average-Case Hardness of MKTP.

```
function $f(\ell, t, M)$
    The input consists of integers $\ell \in [n + c_1 \log n], t \in [(c + c_1 + 2)\log n]$, and a string
  $M \in \{0,1\}^{n + c_1 \log n}$.
    $M' \leftarrow$ the first $\ell$ bits of $M$
    $out \leftarrow \mathsf{Output}(M', t)$
    if $|out| = n$ then
        return the concatenation of $\ell$, $t$, and $out$
    else
        return $\perp$
```

Since $t \leq O(\log n)$, we can always compute $\mathsf{Output}(M', t)$ in logarithmic space. It follows that f is computable in logarithmic space.

Let $\mathcal{D}_{\mathsf{owf}}$ be the output distribution of f on uniform inputs. In other words, to sample from $\mathcal{D}_{\mathsf{owf}}$, we sample two integers $\ell \leftarrow [n + c_1 \log n]$, $t \leftarrow [(c + c_1 + 2)\log n]$ and a string M of length $n + c_1 \log n$, and output $f(\ell, t, M)$. We prove that $\mathcal{D}_{\mathsf{owf}}$ *almost dominates* the uniform distribution over $\{0,1\}^n$, in the following sense.

▷ Claim 35. Let n be a large enough integer. For every string x such that $t(x) \leq (c + c_1 + 2)\log n$, the probability that a random sample from $\mathcal{D}_{\mathsf{owf}}$ is equal to $(\mathrm{KT}(x) - t(x), t(x), x)$ is at least $\frac{1}{2^n n^{2+c_1}}$.

Proof of Claim 35. For a large enough n, with probability at least $\frac{1}{n^2}$, the sampler for $\mathcal{D}_{\mathsf{owf}}$ samples $t = t(x)$ and $\ell = \mathrm{KT}(x) - t(x)$. Then, with probability $\frac{1}{2^\ell} \geq \frac{1}{2^n n^{c_1}}$, the sampler samples a description M' such that $\mathsf{Output}(M', t) = x$. It follows that w.p. at least $\frac{1}{2^n n^{2+c_1}}$ the sampler outputs $(\mathrm{KT}(x) - t(x), t(x), x)$. ◁

Now, we can prove the security of the weak OWF f. Let $\mathcal{A}_{\mathsf{owf}}$ be a candidate PPT adversary trying to invert f. We construct a polynomial-time algorithm $\mathcal{A}_{\mathsf{KT}}$ that attempts to solve (the search version of) MKTP as in Algorithm 2.

Note that for a fixed input x, $\mathcal{A}_{\mathsf{KT}}(x)$ fails to output a valid witness for $\mathrm{KT}(x)$ only if $\mathcal{A}_{\mathsf{owf}}$ fails to invert the output $(\mathrm{KT}(x) - t(x), t(x), x)$. Let $p_{\mathsf{fail}}(x)$ be the probability (over the internal randomness of $\mathcal{A}_{\mathsf{owf}}$) that $\mathcal{A}_{\mathsf{owf}}$ fails to invert $(\mathrm{KT}(x) - t(x), t(x), x)$, then we have

$$\mathop{\mathbb{E}}_{\mathbf{x} \leftarrow \mathcal{U}_n}[p_{\mathsf{fail}}(\mathbf{x})] \geq \Pr_{\mathbf{x} \leftarrow \mathcal{U}_n}[\mathcal{A}_{\mathsf{KT}}(\mathbf{x}) \text{ fails on input } \mathbf{x}] \geq \frac{1}{n^c}. \tag{1}$$

Let p be the probability that $\mathcal{A}_{\mathsf{owf}}$ fails to invert a random input of f. Then

Algorithm 2 Bounded-Error Heuristic $\mathcal{A}_{\mathsf{KT}}$ for MKTP from Inverter $\mathcal{A}_{\mathsf{owf}}$ for f.

```
function $\mathcal{A}_{\mathsf{KT}}(x)$
    $n \leftarrow |x|$; $\textsc{Opt} \leftarrow +\infty$; $\textsc{Witness} \leftarrow \perp$
    for $\ell \in [n + c_1 \log n]$ and $t \in [(c + c_1) \log n]$ do
        $(\ell', t', M) \leftarrow \mathcal{A}_{\mathsf{owf}}(\ell, t, x)$
        $M' \leftarrow$ the first $\ell'$ bits of $M$
        if $\mathsf{Output}(M', t') = x$ and $\textsc{Opt} > |M'| + t'$ then
            $\textsc{Opt} \leftarrow |M'| + t'$
            $\textsc{Witness} \leftarrow (M', t')$
    return $(\textsc{Opt}, \textsc{Witness})$
```

$$
\begin{aligned}
p &\geq \sum_{\substack{x \in \{0,1\}^n \\ t(x) \leq (c+c_1+2)\log n}} \Pr_{\mathbf{y} \leftarrow \mathcal{D}_{\mathsf{owf}}}[\mathbf{y} = (\mathrm{KT}(x) - t(x), t(x), x)] \cdot p_{\mathsf{fail}}(x) \\
&\geq \sum_{\substack{x \in \{0,1\}^n \\ t(x) \leq (c+c_1+2)\log n}} \frac{1}{2^n n^{2+c_1}} \cdot p_{\mathsf{fail}}(x) && \text{(Claim 35)} \\
&\geq \frac{1}{2^n n^{2+c_1}} \left(\sum_{x \in \{0,1\}^n} p_{\mathsf{fail}}(x) - \frac{2^n}{n^{c+1}} \right) && \text{(Claim 34)} \\
&\geq \frac{1}{n^{2+c_1}} \left(\mathop{\mathbb{E}}_{\mathbf{x} \leftarrow \mathcal{U}_n} p_{\mathsf{fail}}(\mathbf{x}) \right) - \frac{1}{n^{c+c_1+3}} \\
&\geq \frac{1}{n^{2+c_1+c}} - \frac{1}{n^{c+c_1+3}}. && \text{By (1)}
\end{aligned}
$$

Let $c' = c + c_1 + 4$, then every PPT adversary $\mathcal{A}_{\mathsf{owf}}$ fails to invert a random input of f w.p. at least $\frac{1}{n^{c'}}$. It follows that f is a weak OWF. ◀

4.2 Hardness of MKTP from One-Way Functions in $\oplus\mathsf{L}$

In this section, we prove the following theorem.

▶ **Theorem 36.** *Suppose there is a one-way function computable in* $\oplus\mathsf{L}$*. Then for every constant* $\lambda > 0$*,* KT *is bounded-error hard on average to approximate within an additive factor of* $\lambda \log n$*.*

Let f be a one-way function in $\oplus\mathsf{L}$. The proof consists of three steps:

- First, we use f to build a condEP-PRG G. If f is computable in $\oplus\mathsf{L}$, then G is also computable in $\oplus\mathsf{L}$. This step is the same as in [62], and follows directly from Theorem 25.
- Second, we construct the randomized encoding $\tilde{G}$ of G. We argue that $\tilde{G}$ is also a condEP-PRG. Moreover, $\tilde{G}$ is computable in $\mathsf{DLOGTIME}$. This step is implemented in Lemma 37.
- Last, as every output of $\tilde{G}$ has small KT complexity, we use the security of $\tilde{G}$ to show that MKTP is bounded-error hard on average. This step is implemented in Lemma 39.

4.2.1 CondEP-PRG in DLOGTIME

In this section, we prove the following lemma that constructs a $\mathsf{DLOGTIME}$-computable condEP-PRG from a $\oplus\mathsf{L}$-computable condEP-PRG.

▶ **Lemma 37.** *Suppose there is a constant $c > 0$ such that for every constant $\lambda > 0$, there is a condEP-PRG G with stretch $\lambda \log n$ and security $1/n^{\lambda}$ that is computable in $\oplus\mathsf{SPACE}[c \log n]$.*

Then there is a constant $c' > 0$ such that for every constant $\lambda > 0$, there is a condEP-PRG $\tilde{G}$ with stretch $\lambda \log n$ and security $1/n^{\lambda}$ that is computable in $\mathsf{TIME}[c' \log n]$.

Proof. Fix an input length n. Let λ' be a constant that depends on λ, we will fix λ' later. Let G be a condEP-PRG with stretch $\lambda' \log n$ and security $1/n^{\lambda'}$ that is computable in $\oplus\mathsf{SPACE}[c \log n]$. We denote the n-th slice of G as $G_n : \{0,1\}^n \to \{0,1\}^{\ell}$, where $\ell := n + \lambda' \log n$.

Let $N := n^c$. Since $G \in \oplus\mathsf{SPACE}[\log N]$, there are projections

$$G_1^{\mathsf{proj}}, G_2^{\mathsf{proj}}, \dots, G_\ell^{\mathsf{proj}} : \{0,1\}^n \to \{0,1\}^{\ell_{\mathrm{CMD}}(N)},$$

such that the i-th output bit of $G(x)$ is equal to $\mathrm{CMD}(G_i^{\mathsf{proj}}(x))$. Let $r_1, r_2, \dots, r_\ell$ be random strings of length $\ell_{\mathrm{DCMD}}(N) - 1$. Let P_{CMD} be the $\mathsf{DLOGTIME}$-computable function defined in Theorem 28, then the i-th output bit of $G(x)$ is equal to $\mathrm{DCMD}(P_{\mathrm{CMD}}(G_i^{\mathsf{proj}}(x), r_i))$. We define

$$\tilde{G}(x, r_1, \dots, r_\ell) = P_{\mathrm{CMD}}(G_1^{\mathsf{proj}}(x), r_1) \circ P_{\mathrm{CMD}}(G_2^{\mathsf{proj}}(x), r_2) \circ \dots \circ P_{\mathrm{CMD}}(G_\ell^{\mathsf{proj}}(x), r_\ell).$$

($\tilde{G}$ is the "randomized encoding" of G in the sense of [11].)

It is easy to see that there is a constant c_g depending only on c, such that $\tilde{G} \in \mathsf{TIME}[c_g \log n]$. Note that the input length of $\tilde{G}$ is $n_{\mathsf{in}} := n + \ell \cdot (\ell_{\mathrm{DCMD}}(N) - 1)$, the output length of $\tilde{G}$ is $n_{\mathsf{out}} := \ell \cdot \ell_{\mathrm{DCMD}}(N)$, and $n_{\mathsf{out}} = n_{\mathsf{in}} + (\ell - n) = n_{\mathsf{in}} + \lambda' \log n$. Here, we fix λ' large enough such that $\lambda' \ge \lambda \frac{\log n_{\mathsf{in}}}{\log n}$.

▷ Claim 38. $\tilde{G}$ is a condEP-PRG with stretch $\lambda \log n_{\mathsf{in}}$ and security $1/(n_{\mathsf{in}})^{\lambda}$.

Proof. Clearly, the stretch of $\tilde{G}$ is $\lambda' \log n \ge \lambda \log n_{\mathsf{in}}$.

Suppose $\mathcal{E} = \{\mathcal{E}_n \subseteq \{0,1\}^n\}$ is a sequence of events such that G and $\mathcal{E}$ satisfy Definition 24. Let $\tilde{\mathcal{E}} := \{\tilde{\mathcal{E}}_{n_{\mathsf{in}}}\}$ where $\tilde{\mathcal{E}}_{n_{\mathsf{in}}} := \mathcal{E}_n \times \{0,1\}^{\ell \cdot (\ell_{\mathrm{DCMD}}(N) - 1)}$. We verify that $\tilde{G}$ and $\tilde{\mathcal{E}}$ satisfy Definition 24.

Pseudorandomness. Suppose, for the sake of contradiction, that there is a PPT adversary $\mathcal{A}'$ such that

$$\Pr[\mathcal{A}'(\tilde{G}(\tilde{\mathcal{E}}_{n_{\mathsf{in}}}))] - \Pr[\mathcal{A}'(\mathcal{U}_{n_{\mathsf{out}}})] \ge 1/(n_{\mathsf{in}})^{\lambda}.$$

Consider an adversary $\mathcal{A}$ that distinguishes $G(\mathcal{E}_n)$ from $\mathcal{U}_\ell$ as follows. On input y, for every $1 \le i \le \ell$, let $\mathbf{r}_i$ be a uniformly random length-$\ell_{\mathrm{DCMD}}(N)$ input of DCMD such that $\mathrm{DCMD}(\mathbf{r}_i) = y_i$. We concatenate them as $\mathbf{r} = \mathbf{r}_1 \circ \mathbf{r}_2 \circ \dots \circ \mathbf{r}_\ell$, and let $\mathcal{A}(y) = \mathcal{A}'(\mathbf{r})$.

Suppose $\mathbf{y} \leftarrow G(\mathcal{E}_n)$, then the distribution of $\mathbf{r}$ is exactly $\tilde{G}(\tilde{\mathcal{E}}_{n_{\mathsf{in}}})$. On the other hand, suppose $\mathbf{y} \sim \mathcal{U}_\ell$, then the distribution of $\mathbf{r}$ is exactly $\mathcal{U}_{n_{\mathsf{out}}}$. As $\mathcal{A}'$ distinguishes $\tilde{G}(\tilde{\mathcal{E}}_{n_{\mathsf{in}}})$ from $\mathcal{U}_{n_{\mathsf{out}}}$ with advantage $\ge 1/(n_{\mathsf{in}})^{\lambda}$, we can see that $\mathcal{A}$ also distinguishes $G(\mathcal{E}_n)$ from $\mathcal{U}_\ell$ with advantage $\ge 1/(n_{\mathsf{in}})^{\lambda} \ge 1/n^{\lambda'}$, contradicting the security of G.

Entropy-preservation. Consider the above experiment, where we first sample $\mathbf{y} \leftarrow G(\mathcal{E}_n)$, then sample a uniform string $\mathbf{r}_i$ of length $\ell_{\mathrm{DCMD}}(N)$ such that $\mathrm{DCMD}(\mathbf{r}_i) = \mathbf{y}_i$ for every $1 \le i \le \ell$, and finally concatenate them as $\mathbf{r} = \mathbf{r}_1 \circ \mathbf{r}_2 \circ \dots \circ \mathbf{r}_\ell$. The distribution of $\mathbf{r}$ is exactly $\tilde{G}(\tilde{\mathcal{E}}_{n_{\mathsf{in}}})$. Therefore,

$$\begin{aligned} \mathrm{H}(\tilde{G}(\tilde{\mathcal{E}}_{n_{\mathsf{in}}})) &= \mathrm{H}(G(\mathcal{E}_n)) + \ell \cdot (\ell_{\mathrm{DCMD}}(N) - 1) \\ &\ge n - \Omega(\log n) + \ell \cdot (\ell_{\mathrm{DCMD}}(N) - 1) \\ &\ge n_{\mathsf{in}} - \Omega(\log n_{\mathsf{in}}). \end{aligned}$$

◁

We have only defined $\tilde{G}$ and $\tilde{\mathcal{E}}$ on input lengths of the form

$$n_{\mathsf{in}}(n) = n + (n + \lambda' \log n)(\ell_{\mathrm{DCMD}}(n^c) + 1).$$

However, it is straightforward to define $\tilde{G}$ and $\tilde{\mathcal{E}}$ on every input length. Let m be an input length, $m' = n_{\mathsf{in}}(n)$ be the largest number of the form $n_{\mathsf{in}}(n)$ such that $m' \le m$. On input $x \in \{0,1\}^m$, let x_1 be the length-m' prefix of x and x_2 be the rest of x (i.e., $x = x_1 \circ x_2$), and we can define $\tilde{G}(x) = \tilde{G}(x_1) \circ x_2$. Similarly, we could define $\tilde{\mathcal{E}}_m = \tilde{\mathcal{E}}_{m'} \times \{0,1\}^{m-m'}$. ◀

4.2.2 Hardness of MKTP

▶ **Lemma 39.** *Suppose there is a constant $c > 0$ such that for every constant $\lambda > 0$, there is a condEP-PRG G with stretch $\lambda \log n$ and security $1/n^\lambda$ that is computable in $\mathsf{TIME}[c \log n]$.*

Then for every constant $\lambda > 0$, KT *is bounded-error hard on average to approximate within an additive error of $\lambda \log n$.*

Proof. Let $\lambda' := \lambda + c_1 + 2$ for a constant c_1 defined later, and G be a condEP-PRG with stretch $\lambda' \log n$ and security $1/n^{\lambda'}$ that is computable in $\mathsf{TIME}[c \log n]$. Fix an input length n, and let $\ell := n + \lambda' \log n$.

We note that the KT complexity of every output of G is nontrivial. Let c_1 be a large enough constant that only depends on c. Since $G \in \mathsf{TIME}[c \log n]$, there is a description d of constant length such that the following holds: For every input $x \in \{0,1\}^n$, every $1 \le i \le \ell+1$, and every $b \in \{0,1,\star\}$, $U^{d,x}(i,b)$ accepts in $(c_1 - 1)\log n$ time if and only if the i-th bit of $G(x)$ is equal to b. It follows that for every $x \in \{0,1\}^n$,

$$\mathrm{KT}(G(x)) \le n + (c_1 - 1)\log n + O(1) < \ell - (\lambda' - c_1)\log n.$$

Suppose, for the sake of contradiction, that KT is bounded-error easy on average to approximate, within an additive factor of $\lambda \log n$. For a large constant c_{kt} that we fix later, there is a PPT machine $\mathcal{A}$ such that

$$\Pr_{\mathbf{y} \leftarrow \mathcal{U}_\ell}[\mathrm{KT}(\mathbf{y}) \le \mathcal{A}(\mathbf{y}) \le \mathrm{KT}(\mathbf{y}) + \lambda \log n] \ge 1 - \frac{1}{n^{c_{\mathrm{kt}}}}. \tag{2}$$

It is natural to consider the following adversary $\mathcal{A}'$: On input $y \in \{0,1\}^\ell$, $\mathcal{A}'$ outputs 1 if $\mathcal{A}(y) \ge \ell - 2\log n$, and outputs 0 otherwise. We will prove the following two lemmas, showing that $\mathcal{A}'$ distinguishes $G(\mathcal{E}_n)$ from $\mathcal{U}_\ell$ with good advantage.

▶ **Lemma 40.** $\Pr_{\mathbf{y} \leftarrow \mathcal{U}_\ell}[\mathcal{A}'(\mathbf{y}) = 1] \ge 1 - \frac{1}{n^2} - \frac{1}{n^{c_{\mathrm{kt}}}}$.

Proof. By Fact 19, all but a $\frac{1}{n^2}$ fraction of strings $y \in \{0,1\}^\ell$ satisfies that $\mathrm{K}(y) \ge \ell - 2\log n$. Therefore, for all but a $\left(\frac{1}{n^2} + \frac{1}{n^{c_{\mathrm{kt}}}}\right)$ fraction of strings $y \in \{0,1\}^\ell$, we have $\mathcal{A}(y) \ge \mathrm{KT}(y) \ge \mathrm{K}(y) \ge \ell - 2\log n$. On these strings y we have $\mathcal{A}'(y) = 1$. ◁

▶ **Lemma 41.** $\Pr_{\mathbf{y} \leftarrow G(\mathcal{E}_n)}[\mathcal{A}'(\mathbf{y}) = 1] \le 1 - \frac{1}{n}$.

Proof. Let $H := \mathrm{H}(G(\mathcal{E}_n))$. Let d be the constant such that $\ell - H \le d \log n$. The constant d does not depend on c_{kt}, which means we can set $c_{\mathrm{kt}} := d + 15$.

Consider the set of outputs of G that is outputted with probability at most 2^{1-H}. We say these inputs are *good*. Let Good be the set of good inputs, i.e.,

$$\mathsf{Good} := \left\{ y \in \{0,1\}^\ell : 0 < \Pr[G(\mathcal{E}_n) = y] \le 2^{1-H} \right\}.$$

We can see that there are many good strings. Actually, let $p := \Pr_{\mathbf{y}\leftarrow G(\mathcal{E}_n)}[\mathbf{y} \in \mathsf{Good}]$, then

$$H = \mathrm{H}(G(\mathcal{E}_n)) \le p \cdot n + (1-p)\cdot(H-1),$$

which implies that $p \ge \frac{1}{n-H+1}$.

Let Err be the subset of Good on which $\mathcal{A}$ fails to produce a good approximation of KT. (In case that $\mathcal{A}$ is a randomized algorithm, it fails w.p. at least $1/n^4$.) That is,

$$\mathsf{Err} := \left\{y \in \mathsf{Good} : \Pr[\mathrm{KT}(y) \le \mathcal{A}(y) \le \mathrm{KT}(y) + \lambda \log n] \le 1 - 1/n^4\right\}.$$

By Equation (2), $|\mathsf{Err}| \le 2^\ell / n^{c_{\mathrm{kt}}-4}$. Therefore,

$$\Pr_{\mathbf{y}\leftarrow G(\mathcal{E}_n)}[\mathbf{y} \in \mathsf{Err}] \le (2^\ell / n^{c_{\mathrm{kt}}-4}) \cdot 2^{1-H} \le 2 \cdot n^{d+4-c_{\mathrm{kt}}} \le 1/n^4.$$

Note that for every y in the range of $G(\mathcal{E}_n)$, if $\mathcal{A}$ is correct on y, we have $\mathcal{A}(y) < \ell - (\lambda' - c_1)\log n + \lambda \log n = \ell - 2\log n$. Therefore for every $y \in \mathsf{Good} \setminus \mathsf{Err}$, we have $\mathcal{A}'(y) = 0$ w.p. at least $1 - 1/n^4$ over the internal randomness of $\mathcal{A}'$. It follows that

$$\begin{aligned}\Pr_{\mathbf{y}\leftarrow G(\mathcal{E}_n)}[\mathcal{A}'(\mathbf{y}) = 1] &\le (1-p) + \Pr_{\mathbf{y}\leftarrow G(\mathcal{E}_n)}[\mathbf{y} \in \mathsf{Err}] + \frac{1}{n^4} \\ &\le 1 - \frac{1}{n-H+1} + \frac{1}{n^4} + \frac{1}{n^4} \\ &\le 1 - \frac{1}{n}. \end{aligned}$$ ◁

From the pseudorandomness of the condEP-PRG G, we conclude that KT is hard on average to approximate within an additive error of $\lambda \log n$.

Note that we have only proved the hardness of MKTP on input lengths of the form $n + \lambda' \log n$, but it is straightforward to extend the argument to every input length m. Let m' be the largest number of the form $m' = n + \lambda' \log n$ such that $m' \le m$, then $m - m' \le O(1)$. For every $x \in \{0,1\}^m$, let x_1 be the length-m' prefix of x. There is an absolute constant d such that $\mathrm{KT}(x_1) - d \log m \le \mathrm{KT}(x) \le \mathrm{KT}(x_1) + d \log m$. It follows that if we can approximate MKTP on input length m', then we can also approximate MKTP on input length m. ◀

4.2.3 Proof of Theorem 36

▶ **Theorem 36.** *Suppose there is a one-way function computable in* $\oplus\mathsf{L}$*. Then for every constant* $\lambda > 0$*,* KT *is bounded-error hard on average to approximate within an additive factor of* $\lambda \log n$*.*

Proof. Let c be a constant such that there is a one-way function f computable in $\oplus\mathsf{SPACE}[c \log n]$. Let $\mathsf{EP\text{-}PRG}$ be the Turing machine guaranteed in Theorem 25. For every constant $\lambda > 0$, let $G(x,z) = \mathsf{EP\text{-}PRG}(\lambda, x, f(x), z)$. Then there is a constant c_1 only depending on c (not on λ) such that G is computable in $\oplus\mathsf{SPACE}[c_1 \log n]$. Moreover, G is a condEP-PRG with stretch $\lambda \log n$ and security $1/n^\lambda$.

By Lemma 37, there is a constant c_2 only depending on c such that for every constant $\lambda > 0$, there is a condEP-PRG with stretch $\lambda \log n$ and security $1/n^\lambda$ that is computable in $\mathsf{TIME}[c_2 \log n]$. By Lemma 39, for every constant $\lambda > 0$, KT is bounded-error hard on average to approximate within an additive error of $\lambda \log n$. ◀

4.3 Bounded-Error Average-Case Robustness of Meta-Complexity

Our techniques also show that the meta-complexity of (resource-bounded) Kolmogorov complexity is "robust", i.e. a slight change in the underlying computation model has little effect on their hardness. Actually, for many resource-bounded variants of Kolmogorov complexity, such as KT, NC^1-K^t, and L-K^t, either all of them admit bounded-error polynomial-time heuristics, or none of them do. (See Section 3.2 for their definition.)

▶ **Theorem 42.** *The following are equivalent:*

1. *There is a one-way function computable in* $\oplus\mathsf{L}$*.*
2. *There is a one-way function computable in* $\mathsf{DLOGTIME}$*.*
3. *The search version of* KT *is hard on average.*
4. *For every constant* $\lambda > 0$*,* KT *is hard on average to approximate within an additive error of* $\lambda \log n$*.*
5. *There is a polynomial* $t(\cdot)$ *such that the search version of* NC^1-K^t *is hard on average.*
6. *For every constant* $\lambda > 0$ *and polynomial* $t(\cdot)$ *such that* $t(n) > 2n$*,* NC^1-K^t *is hard on average to approximate within an additive error of* $\lambda \log n$*.*
7. *There is a polynomial* $t(\cdot)$ *such that the search version of* L-K^t *is hard on average.*
8. *For every constant* $\lambda > 0$ *and polynomial* $t(\cdot)$ *such that* $t(n) > 2n$*,* L-K^t *is hard on average to approximate within an additive error of* $\lambda \log n$*.*

Proof Sketch. (2) $\implies$ (1), (4) $\implies$ (3), (6) $\implies$ (5), and (8) $\implies$ (7) are trivial.

(3) $\implies$ (2): Directly from Theorem 33.

(5) $\implies$ (2) and (7) $\implies$ (2): The construction from [62, Section 4] gives a one-way function computable in $\mathsf{ALOGTIME}$ (i.e., uniform NC^1), based on the hardness of NC^1-K^t. By Theorem 32, there is a one-way function computable in $\mathsf{DLOGTIME}$. The same argument works for L-K^t.

(1) $\implies$ (4): Directly from Theorem 36.

(1) $\implies$ (6): Consider the condEP-PRG G computable in $\mathsf{TIME}[c \log n]$ that we constructed in the proof of Theorem 36, where c is some constant. Let $t'(n) := n^{O(c)}$, for every $x \in \{0,1\}^n$ that in the range of G, NC^1-$\mathrm{K}^{t'}(x) \le n - \Theta(\log n)$. It follows that there is a polynomial t' such that NC^1-$\mathrm{K}^{t'}$ is hard on average to approximate.

To prove that NC^1-K^t is hard on average to approximate for every polynomial t, we use a padding trick. (See also [62, Theorem 5.6].) Let $\epsilon > 0$ be a small enough constant, and $n_1 = n^\epsilon$. Consider the generator $G'(x, r) = G(x) \circ r$, where $|x| = n_1$ and $|r| = n - n_1$. It is easy to see that if G is a condEP-PRG, then G' is also a condEP-PRG. For every $x \in \{0,1\}^n$ that is in the range of G', if we take ϵ to be a small enough constant, we have NC^1-$\mathrm{K}^t(x) \le n - \Theta(\log n)$. Since G' is pseudorandom, NC^1-K^t is hard on average to approximate.

(1) $\implies$ (8): The same argument as in (1) $\implies$ (6) also works for L-K^t. ◀

4.4 Zero-Error Average-Case Reductions

Our techniques actually imply reductions among MKTP, NC^1-MINKT, and L-MINKT. A closer look at these reductions reveals that they are not only *two-sided error* average-case reductions, but also *zero-error* ones! This allows us to prove new relations between the *zero-error* average-case complexity of variants of MINKT and MKTP.

The standard definition of an average-case complexity class, such as AvgZPP, is a class of pairs $(L, \mathcal{D})$ where L is a language, and $\mathcal{D}$ is a distribution ensemble over inputs. (See, e.g., [12, Chapter 18].) In this section, we only deal with the uniform distribution as the input distribution. Therefore, for simplicity, we define AvgZPP as a class of languages rather than (language, distribution) pairs.

▶ **Definition 43.** *Let L be a language and $\delta > 0$ be a constant. We say $L \in \mathsf{Avg}_\delta\mathsf{ZPP}$ if there is a zero-error PPT heuristic $\mathcal{H}$, such that the following are true: (To emphasize that $\mathcal{H}$ is a randomized heuristic, we use $\mathcal{H}(x; r)$ to denote the output of $\mathcal{H}$ on input x and randomness r.)*

- *For every input $x \in \{0,1\}^\star$ and $r \in \{0,1\}^{\mathrm{poly}(|x|)}$, $\mathcal{H}(x;r) \in \{L(x), \bot\}$.*
- *For every integer n, $\Pr_{\mathbf{x}\leftarrow\mathcal{U}_n,\mathbf{r}\leftarrow\mathcal{U}_{\mathrm{poly}(n)}}[\mathcal{H}(\mathbf{x};\mathbf{r}) \neq \bot] \geq \delta$.*

Let $\mathsf{Avg}_{\Omega(1)}\mathsf{ZPP} := \bigcup_{\delta>0} \mathsf{Avg}_\delta\mathsf{ZPP}$.

We consider the parameterized versions of MKTP and MINKT in this section. Let $t(n) \leq \mathrm{poly}(n)$ be a time bound, and $s(n) \leq n$ be a size parameter. We define $\mathrm{MKTP}[s] = \{x : \mathrm{KT}(x) \leq s(|x|)\}$, and $\mathrm{MINK}^t[s] = \{x : \mathrm{K}^{t(|x|)}(x) \leq s(|x|)\}$. The problems NC^1-$\mathrm{MINK}^t[s]$ and L-$\mathrm{MINK}^t[s]$ are defined similarly.

A language L is *sparse* if for every integer n, $\Pr_{\mathbf{x}\leftarrow\mathcal{U}_n}[\mathbf{x} \in L] \leq o(1)$. From Fact 19, for every unbounded function $f(n) = \omega(1)$, $\mathrm{MKTP}[n - f(n)]$ and $\mathrm{MINK}^{\mathrm{poly}(n)}[n - f(n)]$ are sparse. In general, to solve a sparse problem L on average, it suffices to design a heuristic that distinguishes every instance in L from the random instances. Therefore, the following notion of reductions will be convenient for studying the zero-error average-case complexity of sparse problems:

▶ **Definition 44.** *Let L_1, L_2 be two problems. We say there is a* one-sided mapping reduction *from L_1 to L_2, if there are polynomials $p(\cdot)$, $m(\cdot)$, and a randomized polynomial-time mapping $\mathsf{Red} : \{0,1\}^n \times \{0,1\}^{p(n)} \to \{0,1\}^{m(n)}$, such that the following holds.*

- *For every $x \in L_1 \cap \{0,1\}^n$ and $r \in \{0,1\}^{p(n)}$, it holds that $\mathsf{Red}(x;r) \in L_2$.*
- *The distribution of $\mathsf{Red}(\mathcal{U}_n;\mathcal{U}_{p(n)})$ is equal to $\mathcal{U}_{m(n)}$.*

▶ Remark 45. Here we require that the reduction maps the uniform distribution to the uniform distribution exactly. In some cases, this requirement is too strong, and we only need that $\mathcal{U}_{m(n)}$ *dominates* $\mathsf{Red}(\mathcal{U}_n;\mathcal{U}_{p(n)})$. (See [12, Definition 18.6].) Nevertheless, thanks to the perfect randomized encodings [11], we are able to design reductions as strong as Definition 44.

In short, a one-sided mapping reduction (among sparse problems) maps a Yes instance to a Yes instance, and maps a random instance to a random instance. It is easy to see that such reductions preserve the property of being in $\mathsf{Avg}_{\Omega(1)}\mathsf{ZPP}$.

▶ **Fact 46.** *Let L_1, L_2 be two sparse problems. Suppose that there is a one-sided mapping reduction Red from L_1 to L_2. If there is a constant $\delta_2 > 0$ such that $L_2 \in \mathsf{Avg}_{\delta_2}\mathsf{ZPP}$, then there is a constant $\delta_1 > 0$ such that $L_1 \in \mathsf{Avg}_{\delta_1}\mathsf{ZPP}$.*

For every $s_1(n) \leq s_2(n)$, there is a one-sided mapping reduction from $\mathrm{MKTP}[s_1(n)]$ to $\mathrm{MKTP}[s_2(n)]$. (The identity mapping is a valid reduction [43].) Similarly, for every s_1, t_1, s_2, t_2 such that an alternating machine of description length s_1 and (alternating) time $\log t_1$ can be compiled into a deterministic machine of description length s_2 and space $\log t_2$, there is a one-sided mapping reduction from NC^1-$\mathrm{MINK}^{t_1}[s_1]$ to L-$\mathrm{MINK}^{t_2}[s_2]$. (Again, the identity mapping is a valid reduction.)

Now we present a one-sided mapping reduction from L-MINKT to MKTP. Actually, the reduction we present is from L-MINKT to $\mathrm{MINK}^{t'}$, where $t'(n) = \lambda \log n$ for some absolute constant $\lambda > 0$.

▶ **Theorem 47.** *For every polynomial $t(\cdot)$ and integer $c > 0$, there is a constant $c' > 0$ such that there is a one-sided mapping reduction Red from L-$\mathrm{MINK}^t[n - c'\log n]$ to $\mathrm{MINK}^{t'}[n - c\log n]$.*

Proof. For convenience, denote $s(n) := n - c' \log n$. Let $x \in \{0,1\}^n$ be an input to L-$\mathrm{MINK}^t[s]$.

The reduction is simple. It fixes $N := \mathrm{poly}(t(n))$, and reduces a length-n input to a length-$\tilde{N}$ input, where $\tilde{N} := n \cdot \ell_{\mathrm{DCMD}}(N)$. For every bit x_i ($1 \le i \le n$), it samples a uniformly random string $s_i \in \{0,1\}^{\ell_{\mathrm{DCMD}}(N)}$, conditioned on that $\mathrm{DCMD}(s_i) = x_i$. Finally, it outputs the concatenation of $s_1, s_2, \dots, s_n$.

Since DCMD is balanced (Corollary 29), the reduction maps a random instance to a random instance. Now it remains to show that it maps a Yes instance to a Yes instance.

Suppose x is a Yes instance. Denote $\mathsf{Red}(x;r) := s_1 \circ s_2 \circ \cdots \circ s_n$, where r is the random coins that our reduction uses. For $t'(n) = \lambda \log n$, we want to prove that $\mathrm{K}^{t'}(\mathsf{Red}(x;r)) \le \tilde{N} - c \log \tilde{N}$.

Let U be the universal Turing machine we consider, then there is a description d of length at most $s(n)$, such that for every $1 \le i \le n+1$ and every $b \in \{0,1,\star\}$, $U^d(i,b)$ accepts in space $\log t(n)$ if and only if $x_i = b$. Since CMD is L-hard under projections (Theorem 30), for $N = \mathrm{poly}(t(n))$, there is a $\mathsf{DLOGTIME}$-computable projection

$$p_x : \{0,1\}^{s(n)} \times [n+1] \times \{0,1,\star\} \to \{0,1\}^{\ell_{\mathrm{CMD}}(N)},$$

such that for every $1 \le i \le n+1$ and $b \in \{0,1,\star\}$, $x_i = b$ if and only if $\mathrm{CMD}(p_x(d,i,b)) = 1$.

The description of $\mathsf{Red}(x;r)$ contains the string d, and n strings $s'_1, s'_2, \dots, s'_n$ of length $\ell_{\mathrm{DCMD}}(N) - 1$ each. Let P_{CMD} be the $\mathsf{DLOGTIME}$-computable projection in Theorem 28. The string s'_i is chosen such that $P_{\mathrm{CMD}}(p_x(d,i,1), s'_i) = s_i$. (Note that $\mathrm{CMD}(p_x(d,i,1)) = \mathrm{DCMD}(s_i)$, so each s'_i exists and is unique.)

Let $1 \le i \le |\mathsf{Red}(x;r)|$. To compute the i-th bit of $\mathsf{Red}(x;r)$, we first "locate" i by computing $k := \lfloor \frac{i-1}{\ell_{\mathrm{DCMD}}(N)} \rfloor + 1$, and $j := i - \ell_{\mathrm{DCMD}}(N)(k-1)$. Now, the i-th bit of $\mathsf{Red}(x;r)$ is the j-th bit of s_k. We can simply calculate the j-th bit of $P_{\mathrm{CMD}}(p(d,i,1), s'_i)$, which takes $\lambda \log \tilde{N}$ time for some absolute constant $\lambda > 0$.

It follows that whenever L-$\mathrm{K}^t(x) \le n - c' \log n$, regardless of the random bits r we choose, there is a description that allows us to quickly retrieve each bit of $\mathsf{Red}(x;r)$. Moreover, the description has length $n - c' \log n + n(\ell_{\mathrm{DCMD}}(N) - 1) = \tilde{N} - c' \log n$. If the constant c' is big enough compared with c, then $\mathsf{Red}(x;r)$ is a Yes instance of $\mathrm{MINK}^{t'}[\tilde{N} - c \log \tilde{N}]$. ◀

Note that for $c > \lambda$, $\mathrm{MINK}^{t'}[n - c \log n]$ reduces to $\mathrm{MKTP}[n - (c - \lambda) \log n]$ via the identity mapping. (See the proof of Theorem 48.) Therefore, Theorem 47 shows a one-sided mapping reduction from some version of MINKT to some version of MKTP. To the best of our knowledge, this reduction is the first result of its kind.

Moreover, Theorem 47 demonstrates the *robustness* of meta-complexity w.r.t. the zero-error average-case complexity. In particular:

▶ **Theorem 48.** *Let $t(\cdot)$ be a fixed polynomial such that $t(n) > 2n$. The following are equivalent:*

1. *There is a constant $c > 0$ such that NC^1-$\mathrm{MINK}^t[n - c \log n] \in \mathsf{Avg}_{\Omega(1)}\mathsf{ZPP}$.*
2. *There is a constant $c > 0$ such that L-$\mathrm{MINK}^t[n - c \log n] \in \mathsf{Avg}_{\Omega(1)}\mathsf{ZPP}$.*
3. *There is a constant $c > 0$ such that $\mathrm{MINK}^{t'}[n - c \log n] \in \mathsf{Avg}_{\Omega(1)}\mathsf{ZPP}$, where $t'(n) = \lambda \log n$ is defined above.*

Moreover, the above items are implied by the following items:

4. *There is a constant $c > 0$ such that $\mathrm{MKTP}[n - c \log n] \in \mathsf{Avg}_{\Omega(1)}\mathsf{ZPP}$.*

Proof. (4) $\implies$ (3): It suffices to show that for every $c > 0$, the identity mapping reduces $\mathrm{MINK}^{t'}[n - c'\log n]$ to $\mathrm{MKTP}[n - c\log n]$, where $c' = c + \lambda$. Let $x \in \mathrm{MINK}^{t'}[n - c'\log n] \cap \{0,1\}^n$, and d be a description of length $n - c'\log n$ witnessing the fact that $\mathrm{K}^{t'}(x) \le n - c'\log n$. Since $(d, t'(n))$ is also a witness that $\mathrm{KT}(x) \le n - c\log n$. we have $x \in \mathsf{NC}^1\text{-MKTP}[n - c\log n]$.

(3) $\implies$ (2): By Theorem 47 and Fact 46.

(3) $\implies$ (1): Note that the only property of L used in the proof of Theorem 47 is that CMD is hard for L. (In other words, $\mathsf{L} \subseteq \oplus\mathsf{L}$.) As CMD is also hard for NC^1, the proof of Theorem 47 is also true for L-MINKT replaced by NC^1-MINKT.

(2) $\implies$ (3): Every machine that runs in $t'(n)$ time also runs in $t'(n)$ space. Therefore, for every $c > 0$, the identity mapping reduces $\mathrm{MINK}^{t'}[n - c'\log n]$ to $\mathsf{L}\text{-}\mathrm{MINK}^{t_1}[n - c\log n]$, where $t_1(n) = 2^{O(t'(n))}$. We can use a padding trick [62, Theorem 5.6] to reduce $\mathsf{L}\text{-}\mathrm{MINK}^{t_1}$ to $\mathsf{L}\text{-}\mathrm{MINK}^{t}$.

(1) $\implies$ (3): The same argument as (2) $\implies$ (3) also works for $\mathsf{NC}^1\text{-}\mathrm{K}^t$. ◀

5 Tighter Connections between Meta-Complexity and One-Way Functions

In this section, we present a tighter connection between the hardness of MINKT (or MKTP) and the maximum security of *weak* one-way functions. We first define the *security* of weak one-way functions.

▶ **Definition 49.** *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function. We say f is a weak one-way function* with security $S(n)$*, if there is a polynomial $p(\cdot)$ such that for every circuit C of size $S(n)$,*

$$\Pr_{\mathbf{x} \leftarrow \mathcal{U}_n}[C(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))] \le 1 - \frac{1}{p(n)}.$$

Our main results are as follows.

▶ **Theorem 50.** *Let $S(n)$ be any monotone function such that $S(n + O(\log^2 n)) \le S(n) \cdot n^{O(\log n)}$. The following are equivalent:*

(a) *There is a weak one-way function with security $S(n) \cdot n^{\Theta(\log n)}$.*

(b) *There are polynomials p, t such that the search version of K^t requires $S(n) \cdot n^{\Theta(\log n)}$ size to compute on a $1 - 1/p(n)$ fraction of inputs.*

(c) *For every constant $\lambda > 0$, there are polynomials p, t, such that K^t requires $S(n) \cdot n^{\Theta(\log n)}$ size to $(\lambda \log n)$-approximate on a $1 - 1/p(n)$ fraction of inputs.*

▶ **Theorem 51.** *Suppose there is a weak one-way function f with security $2^{\Omega(n)}$ computable in $\mathsf{DLOGTIME}$. Then there is a polynomial p such that KT requires $2^{\Omega(n)}$ size to compute on a $1 - 1/p(n)$ fraction of inputs.*

▶ Remark 52. A few remarks are in order.

- In this section, we only consider non-uniform adversaries. The reason is that we will use Impagliazzo's hardcore lemma (Lemma 84) in the proof of Theorem 57, which only works for non-uniform adversaries. We remark that there are hardcore lemmas that also work for uniform adversaries: if there is no time-t' algorithm that inverts a weak one-way function on a $1 - o(1)$ fraction of inputs, then there is no time-t algorithm that non-trivially inverts every hardcore of the same one-way function. However, we do not know whether the dependence of t' on t is tight. Theorem 4.5 of [86] achieves $t' = \mathrm{poly}(t)$, but we need $t' = t \cdot \mathrm{polylog}(t)$. We leave this issue for future work.

- Our equivalence only holds for *weak* one-way functions. Indeed, it is an open problem whether the existence of *exponentially*-hard weak one-way functions is equivalent to the existence of *exponentially*-hard strong one-way functions [31]. Yao's hardness amplification theorem (Theorem 23) blows up the input length by a polynomial factor, therefore given a $2^{\Omega(n)}$-hard weak one-way function, it only produces a $2^{n^{\Omega(1)}}$-hard strong one-way function.
- Our result for KT (Theorem 51) is weaker than our result for K^t. In particular, suppose the one-way function has security $2^{\alpha n}$, we can only show that KT requires $2^{\beta n}$ size on average, for some constant β that is much smaller than α.
- The best seed length of known explicit extractors that extract all min-entropy is $O(\log^2 n)$ [36]. This is why we see an $n^{\Theta(\log n)}$ factor in Theorem 50.

We rely on the construction of condEP-PRGs from weak one-way functions in [93, 62], thus we structure this section as follows. In Section 5.1, we define *extractors* and *hardcore functions*, which are technical building blocks of the construction. In Section 5.2, we describe the construction in [93, 62]. (The correctness of this construction is proved in Appendix A.) The proofs of Theorem 50 and 51 appear in Section 5.3 and 5.4 respectively.

5.1 Technical Building Blocks

5.1.1 Extractors

▶ **Definition 53.** *A function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ϵ)*-extractor if for every random variable* X *over* $\{0,1\}^n$ *such that* $\mathrm{H}_\infty(X) \geq k$*, the statistical distance between* $\mathsf{Ext}(X,\mathcal{U}_d)$ *and* $\mathcal{U}_m$ *is at most* ϵ*.*

Moreover, Ext *is a* strong (k,ϵ)-extractor *if for every random variable* X *as above, the statistical distance between* $\mathsf{Ext}(X,\mathcal{U}_d)$ *and* $\mathcal{U}_m$ *is at most* ϵ*, even* conditioned on the seed*. That is, the statistical distance between the following two distributions is at most* ϵ*:*

$$\mathcal{D}_1 := (\mathbf{r} \circ \mathsf{Ext}(\mathbf{x},\mathbf{r}) \mid \mathbf{r} \leftarrow \mathcal{U}_d, \mathbf{x} \leftarrow X), \text{ and } \mathcal{D}_2 := \mathcal{U}_{d+m}.$$

5.1.2 Hardcore Functions

▶ **Definition 54.** *Let* $\epsilon = \epsilon(n) > 0$*,* $L = L(n) \leq \mathrm{poly}(n)$*,* $\mathsf{HC} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a function, and* R *be a probabilistic oracle algorithm. We say* HC *is a* hardcore function *with reconstruction algorithm* R*, distinguishing probability* ϵ*, and list size* L*, if the following holds.*

- *On every oracle* $\mathcal{O}$*,* $R^{\mathcal{O}}$ *outputs a list of* L *strings of length* n*.*
- *For every string* x *and every oracle* $\mathcal{O}$ *that* ϵ*-distinguishes* $\mathcal{U}_d \circ \mathsf{HC}(x,\mathcal{U}_d)$ *from* $\mathcal{U}_{d+m}$*,* x *is in the list output by* $R^{\mathcal{O}}$ *w.p.* $\geq 1/2$*.*

Our definition of hardcore functions indeed implies the standard definition in [32]:

▶ **Fact 55.** *Let* $\mathsf{HC} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a hardcore function with a* $\mathrm{poly}(n)$*-time reconstruction algorithm, distinguishing probability* $\epsilon = 1/\mathrm{poly}(n)$*, and list size* $L \leq \mathrm{poly}(n)$*.*

Let f *be any one-way function,* $\mathbf{x} \leftarrow \mathcal{U}_n$*, and* $\mathbf{r} \leftarrow \mathcal{U}_d$*. No polynomial-size adversary can* 2ϵ*-distinguish the distribution* $f(\mathbf{x}) \circ \mathbf{r} \circ \mathsf{HC}(\mathbf{x},\mathbf{r})$ *from the distribution* $f(\mathbf{x}) \circ \mathbf{r} \circ \mathcal{U}_m$*.*

Proof. Let $\mathcal{A}$ be an adversary of size $\mathrm{poly}(n)$ that 2ϵ-distinguishes the distribution $f(\mathbf{x}) \circ \mathbf{r} \circ \mathsf{HC}(\mathbf{x},\mathbf{r})$ from $f(\mathbf{x}) \circ \mathbf{r} \circ \mathcal{U}_m$. Say $x \in \{0,1\}^n$ is *good* if $\mathcal{A}$ can ϵ-distinguish $f(x) \circ \mathbf{r} \circ \mathsf{HC}(x,\mathbf{r})$ from $f(x) \circ \mathbf{r} \circ \mathcal{U}_m$. Then by a Markov bound, at least an ϵ fraction of inputs x are good. We will use $\mathcal{A}$ to invert $f(x)$ on every good input x in probabilistic polynomial time. Our inversion algorithm will have success probability $1/2$ on a good x; as $(\epsilon/2) > 1/\mathrm{poly}(n)$, this contradicts the one-wayness of f.

On input $y = f(x)$, where x is good, define the oracle

$$\mathcal{O}(z) := \mathcal{A}(y, z).$$

Then $\mathcal{O}$ can ϵ-distinguish $\mathcal{U}_d \circ \mathsf{HC}(x, \mathcal{U}_d)$ from $\mathcal{U}_{d+m}$. The reconstruction algorithm $R^{\mathcal{O}}$ outputs a list of size $\mathrm{poly}(n)$ which contains x. We could easily find any element x' in this list such that $f(x') = y$, and output x'. With probability $1/2$ over the internal randomness of R, we invert y successfully. ◀

5.2 CondEP-PRGs from Weak One-Way Functions

In this section, we present the following construction from weak one-way functions to condEP-PRGs.

▶ **Construction 56** ([93, 62]). Let $0 < \epsilon < \frac{1}{10n^2}$ be the desired security parameter of the condEP-PRG (i.e., it should be $O(\epsilon)$-indistinguishable from uniformly random strings). Let $\delta > 0$, and f be a weak one-way function that is hard to invert on a $(1-\delta)$ fraction of inputs. Let $\alpha > 0$ be the desired stretch of our condEP-PRG. Suppose we have the following objects:

- For every k, a strong (k, ϵ)-extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with optimal output length, where $d := d_{\mathsf{Ext}}(n, \epsilon)$ and $m := k - 2\log(1/\epsilon) - O(1)$. We write the extractor as $\mathsf{Ext}^{(k)}$ if we need to emphasize the min-entropy parameter k.
- For $k_{\mathsf{HC}} := \alpha + \log(n/\delta) + 4\log(1/\epsilon) + O(1)$, a hardcore function $\mathsf{HC} : \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{k_{\mathsf{HC}}}$ with $\mathrm{poly}(n/\epsilon)$-time reconstruction algorithm R, distinguishing probability ϵ, and list size $L \le \mathrm{poly}(n/\epsilon)$, where $d' := d_{\mathsf{HC}}(n, k_{\mathsf{HC}}, \epsilon)$.

Let $G_{n,r} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^d \times \{0,1\}^{d'} \to \{0,1\}^{n+2d+d'+\alpha}$ be the following construction:

$$G_{n,r}(x, z_1, z_2, z_3) := z_1 \circ \mathsf{Ext}^{(r-1)}(x, z_1) \circ z_2 \circ \mathsf{Ext}^{(\lfloor n-r-\log(2n/\delta)\rfloor)}(f(x), z_2) \circ z_3 \circ \mathsf{HC}(x, z_3).$$

▶ **Theorem 57.** *Let $\epsilon, \delta, \alpha, f$ be defined as in Construction 56. If $\epsilon \ge 1/\mathrm{poly}(n)$ and $L \le \mathrm{poly}(n)$, then there is a function $r : \mathbb{N} \to \mathbb{N}$ such that $G = \{G_{n,r(n)}\}_{n\in\mathbb{N}}$ is a condEP-PRG with stretch α and security 4ϵ.*

More precisely, let $\tilde{n} = n + 2d + d'$. Suppose that for every subset $\mathcal{D} \subseteq \{0,1\}^{\tilde{n}}$ such that $\mathrm{H}(G(\mathcal{D})) \ge \tilde{n} - \Omega(\log(\frac{n}{\delta\epsilon}))$ and every k, there is an adversary of size s that 4ϵ-distinguishes $G_{n,k}(\mathcal{D})$ from the uniform random distribution. Then there is an adversary of size $s \cdot \mathrm{poly}(nL/\epsilon)$ that inverts f on a $1-\delta$ fraction of inputs.

The proof basically follows from [62], and we present a self-contained proof in Appendix A. However, there are two major differences between our proof and the proof in [62]:

- We replace the extractors and hardcore functions with better constructions. In particular, our extractors and hardcore functions in Section 5.3 requires only $O(\log^2 n)$ random bits.
- More importantly, in the very beginning, we need to transform the weak one-way function into a strong one. [62] uses hardness amplification (Theorem 23) to implement this step. However, Theorem 23 does not preserve *exponential* security, therefore we use *Impagliazzo's hardcore lemma* [49] instead. We only obtain a strong one-way function on a "hardcore" distribution of inputs (instead of the uniform distribution), but this already suffices for our purpose.

5.2.1 Warm-Up: Proof of Theorem 25

Theorem 57 immediately implies Theorem 25.

▶ **Theorem 25** ([62]). *There is a function* EP-PRG *computable in* ALOGTIME*, such that the following holds. For any one-way function $f : \{0,1\}^\star \to \{0,1\}^\star$ and any constant $\gamma > 0$, let $G(x,z) = \mathsf{EP\text{-}PRG}(\gamma, x, f(x), z)$, then G is a condEP-PRG with stretch $\gamma \log n$ and security $1/n^\gamma$.*

We first introduce the (very simple) extractors and hardcore functions used in [93, 62].

- The extractors are derived from the *leftover hash lemma* [38]. (See also [85, Theorem 6.18].) Let $h : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a pairwise independent family of hash functions, where $d = O(n+m)$, then for every k, ϵ such that $m = k - 2\log(1/\epsilon)$, h is also a strong (k,ϵ)-extractor.
  We instantiate the pairwise independent hash family by Toeplitz matrices.[15] More precisely, our keys will have length $d := n + m - 1$, and every $key \in \{0,1\}^{n+m-1}$ corresponds to a Toeplitz matrix. For every $1 \le i \le m$ and every input $x \in \{0,1\}^n$, the i-th output of $H(x, key)$ is the inner product of x and $key_{i\sim(i+n-1)}$ (the substring of key from the i-th bit to the $(i+n-1)$-th bit) in GF(2). In other words, $\mathsf{Ext}(x, key)$ is the concatenation of $\langle x, key_{i\sim(i+n-1)}\rangle$ for each i, where $\langle\cdot,\cdot\rangle$ denotes inner product.
- Let $\mathsf{GL} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^k$ be the Goldreich-Levin hardcore function.
  In [32], GL is defined in terms of Toeplitz matrices (again). Let $d := n + k - 1$. For every $x \in \{0,1\}^n$, $r \in \{0,1\}^d$ and $1 \le i \le k$, the i-th output bit of $\mathsf{GL}(x,r)$ is the inner product of x and $r_{i\sim(i+n-1)}$ in GF(2). Also, it is shown in [32] that for every $\epsilon > 0$, GL is a hardcore function with distinguishing probability ϵ and list size $\mathrm{poly}(n \cdot 2^k/\epsilon)$.

Proof Sketch of Theorem 25. We can plug the parameters $\epsilon := \frac{1}{4n^\gamma}$, $\alpha := \gamma \log n$, $\delta := 1/2$ into Theorem 57. The list size of GL is $L \le \mathrm{poly}(n)$. Theorem 57 gives us a function $r : \mathbb{N} \to \mathbb{N}$ such that $\{G_{n,r(n)}\}$ is a condEP-PRG with stretch $\gamma \log n$ and security $1/n^\gamma$. We can easily construct a uniform condEP-PRG with essentially the same stretch and security: We parse the input as an integer $r \le n$, a string x of length n, and some garbage w. Then we output $G_{n,r}(x) \circ w$.

Now we implement EP-PRG in alternating time $c \log n$, for some absolute constant $c > 0$ independent of γ. On input $(\gamma, x, f(x), z, i)$, we want to compute the i-th output bit of our condEP-PRG. This bit is either equal to some input bit, or the inner product of two length-n sub-strings of the input. It is easy to implement either case in alternating $O(\log n)$ time. ◀

5.3 Proof of Theorem 50

To prove Theorem 50, we replace the leftover hash lemma and GL by extractors and hardcore functions with very short seed length:

▶ **Theorem 58** ([36, Theorem 5.14]). *Let $d_{\mathsf{Ext}}(n,\epsilon) := O(\log n \cdot \log(n/\epsilon))$, then for every $1 \le k \le n$ and $\epsilon > 0$, there is a strong (k,ϵ)-extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^{d_{\mathsf{Ext}}(n,\epsilon)} \to \{0,1\}^m$, where $m = k - 2\log(1/\epsilon) - O(1)$ is optimal.*

[15] An $n \times m$ matrix M is *Toeplitz* if $M_{i,j} = M_{i+1,j+1}$ holds for every $1 \le i < n$, $1 \le j < m$. We can represent a Toeplitz matrix by $n + m - 1$ elements, namely the elements in the first row and the first column.

We observe that the "k-wise direct product generator" used in [42, 41] is a good hardcore function:

▶ **Theorem 59.** *Let $d_{\mathsf{HC}}(n,k,\epsilon) := O(k\log(n/\epsilon))$, then there is a hardcore function* $\mathsf{HC} : \{0,1\}^n \times \{0,1\}^{d_{\mathsf{HC}}(n,k,\epsilon)} \to \{0,1\}^k$ *with a* $\mathrm{poly}(n2^k/\epsilon)$*-time reconstruction algorithm R, distinguishing probability ϵ, and list size $L \le 2^k \cdot \mathrm{poly}(k/\epsilon)$.*

Proof Sketch. Consider the function $\mathsf{DP} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{d+k}$ defined in [41, Theorem 7.1]. The first d bits of $\mathsf{DP}(x,z)$ is always equal to z, and we let $\mathsf{HC}(x,z)$ be the remaining k bits of $\mathsf{DP}(x,z)$.

In [41], the reconstruction algorithm is stated as $R^{\mathcal{O}} : \{0,1\}^a \times \{0,1\}^r \to \{0,1\}^n$. Here, $a \le k + O(\log(k/\epsilon))$ is the "advice complexity" of DP, the first a input bits correspond to the advice, and the remaining $r = \mathrm{poly}(n/\epsilon)$ input bits are random coins used by R. For every $x \in \{0,1\}^n$ and every oracle $\mathcal{O}$ that ϵ-distinguishes $\mathsf{DP}(x, \mathcal{U}_d)$ from $\mathcal{U}_{d+k}$, we have

$$\Pr_{\mathbf{w}\leftarrow\mathcal{U}_r}[\exists\alpha \in \{0,1\}^a, R^{\mathcal{O}}(\alpha,\mathbf{w}) = x] \ge 3/4.$$

Our reconstruction algorithm simply samples a random $\mathbf{w} \leftarrow \mathcal{U}_r$, and outputs $R^{\mathcal{O}}(\alpha,\mathbf{w})$ for every $\alpha \in \{0,1\}^a$. It follows that the list size is $L(n,k,\epsilon) \le 2^a \le 2^k\mathrm{poly}(k/\epsilon)$. ◀

Now we use Construction 56 to prove Theorem 50.

▶ **Theorem 50.** *Let $S(n)$ be any monotone function such that $S(n + O(\log^2 n)) \le S(n) \cdot n^{O(\log n)}$. The following are equivalent:*

(a) *There is a weak one-way function with security $S(n) \cdot n^{\Theta(\log n)}$.*

(b) *There are polynomials p, t such that the search version of K^t requires $S(n) \cdot n^{\Theta(\log n)}$ size to compute on a $1 - 1/p(n)$ fraction of inputs.*

(c) *For every constant $\lambda > 0$, there are polynomials p, t, such that K^t requires $S(n) \cdot n^{\Theta(\log n)}$ size to $(\lambda \log n)$-approximate on a $1 - 1/p(n)$ fraction of inputs.*

Proof Sketch. (c) $\implies$ (b) is trivial.

(b) $\implies$ (a): Suppose that the search version of K^t requires $S(n)\cdot n^{\Theta(\log n)}$ size to solve on a $1/p(n)$ fraction of inputs, where p is a polynomial. The construction in [62, Section 4] shows that there is a weak one-way function f, such that every adversary of size $S(n) \cdot n^{\Theta(\log n)}$ only inverts an $1 - 1/q(n)$ fraction of inputs, where $q(n) := O(n \cdot p(n)^2)$.

(a) $\implies$ (c): Suppose there is a constant $\lambda > 0$ such that, for every polynomial p, there is an algorithm of size $S(n) \cdot n^{\Theta(\log n)}$ that approximates K^t on a $1 - 1/p(n)$ fraction of inputs, within an additive error of $\lambda \log n$. Let f be a candidate weak one-way function, $\delta := 1/q(n)$ for any polynomial q, and $\epsilon := 1/n^2$. Let $\alpha := (\lambda + C)\log n$ be the stretch of the condEP-PRG we construct, where C is a large absolute constant. Let $r : \mathbb{N} \to \mathbb{N}$ be any function. Consider the function $G_{n,r(n)}$ in Construction 56, where the input length of $G_{n,r(n)}$ is

$$\tilde{n} := n + 2d_{\mathsf{Ext}}(n,\epsilon) + d_{\mathsf{HC}}(n, O(\log(n/(\delta\epsilon))), \epsilon) = n + O(\log^2 n).$$

Suppose G runs in $t(\tilde{n})$ time, then every output y of G satisfies $\mathrm{K}^{t(|y|)}(y) \le \tilde{n} - (\lambda+2)\log\tilde{n}$.

Consider any sequence of subsets $\mathcal{E} = \{\mathcal{E}_n \subseteq \{0,1\}^n\}$ such that $\mathrm{H}(G_{n,r(n)}(\mathcal{E}_n)) \ge \tilde{n} - \Omega(\log n)$. The same argument as in Lemma 39 shows that there is an adversary of size

$$S(\tilde{n}) \cdot \tilde{n}^{\Theta(\log\tilde{n})} \le S(n) \cdot n^{\Theta(\log n)}$$

that 4ϵ-distinguishes $G_{n,r(n)}(\mathcal{E}_n)$ from the uniform distribution. It follows that there is an adversary of size $S(n) \cdot n^{\Theta(\log n)}$ that inverts f on a $1 - \delta$ fraction of inputs. Therefore, there is no weak one-way function with hardness $S(n) \cdot n^{\Theta(\log n)}$. ◀

5.4 Proof of Theorem 51

To prove Theorem 51, we need a family of universal hash functions that admit very efficient randomized encodings, constructed in [56, 10]. In [10], it was also proved that such hash functions are good extractors (by the leftover hash lemma) and hardcore functions (based on previous works [44, 15]).

In the construction of [11], for a (Boolean) function computable by a parity branching program of size S, its randomized encoding needs at least $\Omega(S^2)$ additional random input bits. Even worse, if such a function has m output bits, the randomized encoding requires $\Omega(mS^2)$ random input bits. However, to prove Theorem 51, we need to preserve *exponential* hardness of our one-way function, which means our extractors and hardcore functions can only have $O(n)$ random input bits. This is exactly what [10] does. In particular, for a "skew" circuit C of size S and possibly many outputs, the randomized encoding of C in [10] only requires $O(S)$ additional random inputs. Such circuits of linear size can already compute many powerful objects, e.g. universal hash functions [56].

5.4.1 Randomized Encodings for Skew Circuits

We introduce the randomized encodings in [10] in more detail.

We consider circuits that consist of AND and XOR gates of fan-in 2, with multiple output gates. Let C be such a circuit, X be a subset of input variables. (For example, let $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, we may think of X as the last d input variables.) We say C is *skew* with respect to X, if every AND gate in C has at least one child labeled by a constant or a variable in X. In particular, this implies that if we substitute the variables in X by (arbitrary) constants, the function that C computes is a *linear* function on variables not in X – each output bit is simply the XOR of a subset of these variables.

Let $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{d+m}$ be a skew circuit w.r.t. the last d inputs, such that the first d outputs of C is always equal to the last d inputs of C.[16] Let s be the number of internal (i.e. non-input, non-output) gates of C. The randomized encoding of C, denoted as $\tilde{C}$, is a function $\tilde{C} : \{0,1\}^n \times \{0,1\}^{d+s} \to \{0,1\}^{d+m+s}$ defined as follows:

- The inputs of $\tilde{C}$ are $x \in \{0,1\}^n$, $w \in \{0,1\}^d$, and $r \in \{0,1\}^s$.
- For each (input, internal, or output) gate $g \in C$, we associate a bit $r(g)$ with it. Each input gate is associated with its input value (i.e. $r(g) = x_i$ or w_i), the i-th internal gate is associated with $r(g) = r_i$, and every output gate is associated with $r(g) = 0$.
- The first d outputs of $\tilde{C}$ are simply w. The remaining $m + s$ outputs correspond to the internal gates and output gates of C. Let the i-th such gate be $g_i = g_j \triangledown g_k$ (where $\triangledown \in \{\mathsf{AND}, \mathsf{XOR}\}$), then the i-th output is $r(g_i)$ XOR $(r(g_j) \triangledown r(g_k))$.

5.4.2 Highly-Uniform Linear-Size Hash Functions

As we are dealing with KT complexity, we will need the randomized encoding to be computable in DLOGTIME. Therefore, our skew circuits need to be *very uniform.* We state our definition of *uniform skew circuits* as follows; it is easy to see that if a family of skew circuits $\{C_n\}$ is uniform, then their randomized encodings can indeed be computed in DLOGTIME.

[16] That is, we pad the last d inputs at the beginning of our outputs, and the remaining m output bits are the "real" outputs of C. This is a technical restriction on C to ensure its randomized encoding exists.

▶ **Definition 60** (Uniform Skew Circuits). *Let $C = \{C_n : \{0,1\}^n \times \{0,1\}^{d(n)} \to \{0,1\}^{s(n)}\}$ be a family of skew circuits, where $d(n)$ and $s(n)$ are computable in time $O(\log n)$. Moreover, assume that the fan-out of every gate is at most 2, and the last $s(n)$ gates (i.e., gates with the largest indices) are output gates.*

We say that C is a uniform *family of skew circuits, if there is an algorithm $\mathcal{A}$ with time complexity linear in its input length, that on inputs n, i (in binary), outputs the information about the i-th gate in C_n. This includes the gate type (input,* AND*, or* XOR*), indices of its input gates (if they exist), and indices of the (at most 2) gates it feeds to.*

▶ Remark 61. It may seem strange that we need to output not only predecessors but also successors of each gate. The reason is that in [56], we will need to *reverse* each wire when we transform an encoding circuit to an "exposure resilient function". In particular, after that construction, the predecessors of each gate will become their previous successors. See Appendix B.4 for details.

We need a family of universal hash functions $\mathcal{H} = \{h_{n,m} : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^m\}$ in [56], where $k = O(n + m)$. This family has the following important property: $\mathcal{H}$ can be computed by a family of linear-size uniform circuits that are skew w.r.t. the second argument (i.e. the last k bits).

▶ **Theorem 62.** *For every integer n, m where $m = O(n)$, there exists an integer $k = O(n)$, and a family of universal hash functions $\{h_{n,m} : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^m\}$, such that $h_{n,m}$ can be computed by a uniform family of linear-size circuits that are skew w.r.t. the second argument.*

In [56], the authors showed that $\mathcal{H}$ can be computed by a family of linear-size skew circuits, but they did not show that the circuits are uniform. Therefore, we include a proof sketch of Theorem 62 in Appendix B, with an emphasis on the uniformity of these circuits.

By the leftover hash lemma of [38], $\{h_{n,m}\}$ is a strong (k, ϵ)-extractor whenever $m = k - 2\log(1/\epsilon)$. It was proved by [15] (based on [44]) that $\{h_{n,m}\}$ are good hardcore functions:

▶ **Lemma 63.** *For every $\epsilon > 0$, $h_{n,m}$ is a hardcore function with distinguishing probability ϵ and a reconstruction algorithm of $\mathrm{poly}(2^m \cdot n/\epsilon)$ time. (As a result, the list size is also $\mathrm{poly}(2^m \cdot n/\epsilon)$.)*

5.4.3 Proof of Theorem 51

▶ **Theorem 51.** *Suppose there is a weak one-way function f with security $2^{\Omega(n)}$ computable in* DLOGTIME*. Then there is a polynomial p such that* KT *requires $2^{\Omega(n)}$ size to compute on a $1 - 1/p(n)$ fraction of inputs.*

Proof Sketch. Let $\delta = 1/\mathrm{poly}(n)$ such that f is hard to invert on a $(1 - \delta)$ fraction of inputs. We plug the hash functions h (which are also extractors and hardcore functions) into Construction 56, to build a condEP-PRG $G : \{0,1\}^{n_1} \to \{0,1\}^{n_1+\alpha}$ with stretch $\alpha := O(\log n)$ and security $4\epsilon \le 1/n^{10}$. Here, since the seed length of h is $O(n)$, we have $n_1 = O(n)$. Moreover, by Theorem 57, the condEP-PRG is 4ϵ-indistinguishable from the uniform distribution by $2^{\Omega(n)}$-size adversaries.

As the hash functions admit a uniform family of skew circuits, the following is true: There is a (uniform) circuit C such that $G(x, z) = C(x, f(x), z)$, and C is a size-$O(n)$ skew circuit w.r.t. the z argument. We replace C by its randomized encoding to obtain another condEP-PRG $\tilde{G} : \{0,1\}^{n_2} \to \{0,1\}^{n_2+\alpha}$, which is computable in DLOGTIME. Here, since the size of C is $O(n)$, we have $n_2 = O(n)$. As every output of $\tilde{G}$ has non-trivial KT complexity, and $\tilde{G}$ is 4ϵ-indistinguishable from the uniform distribution by $2^{\Omega(n)}$-size adversaries, we can see that KT is hard on average. ◀

5.5 The Perebor Hypotheses

We mention some Perebor hypotheses as further research directions. Each hypothesis states that to some extent, "Perebor," or brute-force search, is unavoidable to solve a certain meta-complexity problem. In this paper, we only consider the (bounded-error) average-case complexity of these problems, but similar hypotheses for the worst-case or zero-error average-case complexity can also be made. We only state these hypotheses against (uniform) randomized algorithms; the corresponding hypotheses against non-uniform algorithms (i.e., circuits) will be called "non-uniform Perebor hypotheses" accordingly.

These hypotheses are inspired by, and parallel to, the "exponential time hypotheses" for satisfiability [52, 53, 20]. The *exponential time hypothesis* (ETH) asserts that 3-SAT requires $2^{\epsilon n}$ time to solve, where $\epsilon > 0$ is some absolute constant and n is the number of variables. The *strong exponential time hypothesis* (SETH) asserts that for *any* constant $\epsilon > 0$, CNF-SAT requires $2^{(1-\epsilon)n}$ time to solve. There is a large body of work on these two hypotheses and their variants; in particular, SETH has been a central hypothesis in fine-grained complexity [91].

We believe that the future study of these Perebor hypotheses will bring us more insights into complexity theory, similar to what the study of ETH and SETH has brought us.

The weak Perebor hypotheses. We introduce the following two hypotheses for K^t and KT:

▶ **Hypothesis 64** (Weak Perebor Hypothesis for K^t)**.** *There is a polynomial $t(n) \geq 2n$ and an absolute constant $c \geq 1$ such that the following holds. Every randomized algorithm that runs in $2^{n/c}$ time and attempts to solve K^t fails w.p. at least $1/n^c$ over a uniformly random input.*

▶ **Hypothesis 65** (Weak Perebor Hypothesis for KT)**.** *There is an absolute constant $c \geq 1$ such that the following holds. Every randomized algorithm that runs in $2^{n/c}$ time and attempts to solve* KT *fails w.p. at least $1/n^c$ over a uniformly random input.*

Theorem 50 shows that the non-uniform version of Hypothesis 64 is equivalent to the existence of exponentially-hard weak one-way functions (against non-uniform adversaries). Theorem 51 shows that the non-uniform version of Hypothesis 65 is implied by the existence of exponentially-hard weak one-way function computable in DLOGTIME (also against non-uniform adversaries).

The strong Perebor hypotheses. We start with the following hypothesis:

▶ **Hypothesis 66** (Strong Perebor Hypothesis for K^t)**.** *There are polynomials $t(n) \geq 2n$ and $p(n)$, such that for every constant $\epsilon > 0$, every probabilistic algorithm that runs in $2^{(1-\epsilon)n}$ time and attempts to solve K^t fails w.p. at least $1/p(n)$ over a uniformly random input.*

By Theorem 50, the non-uniform version of Hypothesis 66 is equivalent to the existence of weak one-way functions with hardness $2^{(1-o(1))n}$ (against non-uniform adversaries).

However, Building on Hellman [39], Fiat and Naor [27] showed that no such one-way function exists in the *non-uniform RAM* model. In particular, for *any* function $f : \{0,1\}^n \to \{0,1\}^n$, there is an algorithm that runs in $2^{3n/4}$ time, with random access to an advice tape of length $2^{3n/4}$, and inverts f at *any* point. It is conceivable that a similar attack could also be implemented in circuits, i.e. every function f could be inverted by a circuit of size $2^{99n/100}$ in the worst-case. This gives strong evidence that the non-uniform version of Hypothesis 66 is false. To the best of our knowledge, (the uniform version of) Hypothesis 66 seems secure.

Following [16, Section 1.1], if we still want (non-uniform) maximum hardness, we can consider *collections* of one-way functions, which corresponds to the *conditional* (time-bounded) Kolmogorov complexity.

Fix a universal Turing machine U, let x, y be two strings and t be a time bound. Define $\mathrm{cK}^t(x \mid y)$ as the length of the smallest description d, such that for every $1 \le i \le |x|+1$ and $b \in \{0, 1, \star\}$, $U^{d,y}(i, b)$ accepts in time t if and only if $x_i = b$. Note that the universal Turing machine is given random access to y (for free), hence d is a description of x *conditioned on* y. We assume that the default input distribution of cK^t consists of a random string x and a random string y, both of input length n. Hence in the hypothesis below, we actually state that no non-uniform algorithm of $2^{(1-\epsilon)n}$ size can solve cK^t on input length $2n$.

▶ **Hypothesis 67** (Strong Perebor Hypothesis for cK^t; Non-uniform Version)**.** *There are polynomials $t(n) \ge 2n$ and $p(n)$, such that for every constant $\epsilon > 0$, the following holds. Every non-uniform algorithm of $2^{(1-\epsilon)n}$ size that attempts to solve cK^t fails on a $1/p(n)$ fraction of inputs.*

6 MCSP-Related Results

In this section, we generalize Theorem 1 to the case of MCSP. Throughout this section, we maintain the convention that our input is the truth table of a function $f : \{0, 1\}^n \to \{0, 1\}$, and $N = 2^n$ is the input length. We use tt to denote an input truth table. Recall that $\mathsf{Size}(tt)$ is the circuit complexity of tt. The size of a circuit is always measured in *gates*. We consider circuits over the B_2 basis, i.e., a gate can compute any function over its 2 inputs.

▶ **Theorem 3** (Informal)**.** *The following are true:*

- *If* MCSP *is exponentially hard on average, then there is a (super-polynomially hard) one-way function.*
- *If there is an exponentially hard weak one-way function in* NC^0*, then* MCSP *is (exponentially) hard on average.*

Ideally, we would like to prove that MCSP is bounded-error hard on average if and only if there is a one-way function in $\mathsf{DLOGTIME}$. However, we could only prove weaker results, since we do not have good understandings of the circuit complexity of a random Boolean function.

For KT complexity, we know that a random string x of length N is likely to satisfy that $\mathrm{KT}(x) \in [N - O(\log N), N + O(\log N)]$. That is, N is a good estimate of the KT complexity of a random string, within additive error $\eta := O(\log N)$. It turns out that the overhead of [62] is $2^{O(\eta)}$, which is polynomial in N.

What about MCSP? For the maximum circuit complexity function, we only know that:

▶ **Theorem 68** ([28])**.** *There is a constant c such that the following is true. Let $C(n)$ be the maximum circuit complexity of any function $f : \{0, 1\}^n \to \{0, 1\}$, then $C_{\mathsf{lb}}(n) \le C(n) \le C_{\mathsf{ub}}(n)$, where*

$$C_{\mathsf{lb}}(n) = \frac{2^n}{n}\left(1 + \frac{\log n}{n} - \frac{c}{n}\right), \text{ and } C_{\mathsf{ub}}(n) = \frac{2^n}{n}\left(1 + \frac{3\log n}{n} + \frac{c}{n}\right).$$

Therefore, given a random truth table tt of length $N = 2^n$, we could use any value between $C_{\mathsf{lb}}(n)$ and $C_{\mathsf{ub}}(n)$ as an estimate of $\mathsf{Size}(tt)$. However, we could only prove that our additive error is $\eta := (C_{\mathsf{ub}}(n) - C_{\mathsf{lb}}(n)) \cdot O(n) = O(2^n \log n/n)$.[17] The overhead in [62] would be

[17] The extra $O(n)$ factor is because we measure η by *bit*-complexity instead of *gate*-complexity, and every gate in the (maximum) circuit needs $O(n)$ bits to describe.

$$2^{O(\eta)} = 2^{O\left(N\frac{\log\log N}{\log N}\right)}.$$

Nevertheless, as $\eta = o(N)$ is non-trivial, we can still achieve non-trivial results for MCSP.

▶ Remark 69. Ilango encountered a similar issue in his search-to-decision reduction for MFSP (Minimum Formula Size Problem) [46]. The additive error for formula complexity is $\eta := O(N/\log\log N)$, thus Ilango only managed to show an (average-case) reduction with time complexity $2^{O(\eta)}$ unconditionally.

Comparing [46] and our work, the $2^{O(\eta)}$ factor comes from different reasons. Ilango's algorithm runs in time $\mathrm{poly}(t)$ where t is the number of "near-optimal" formulas for the input truth table; the current best upper bound of t for a random truth table is $2^{O(\eta)}$. In our paper, we need to sample a *uniformly random circuit* (w.r.t. some encoding), and let p be the probability that the truth table of a sampled circuit is equal to a given one; the current best lower bound of p is $2^{-N-O(\eta)}$. (See Section 6.2.) It is an interesting open problem to improve either estimate.

6.1 Preliminaries

6.1.1 Extreme Hardness Amplification for One-Way Functions

We will construct a one-way function f_{MCSP} based on the assumption that MCSP is *exponentially* hard on average. However, we are only able to prove that f_{MCSP} is hard to invert on an inverse-sub-exponential fraction ($2^{-o(N)}$) of inputs. We will need the following variant of Theorem 23, that constructs a strong one-way function (of super-polynomial hardness) from such a one-way function that is "exponentially hard" but also "(sub)exponentially weak".

▶ **Theorem 70.** *Let $p(n) = 2^{o(n)}$, f be a length-preserving function that is exponentially hard to invert on a $1/p(n)$ fraction of inputs. In other words, there is a constant $\epsilon > 0$ such that for every integer n and every randomized algorithm $\mathcal{A}$ that runs in $2^{\epsilon n}$ time,*

$$\Pr_{\mathbf{x}\leftarrow\mathcal{U}_n}\left[\mathcal{A}(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))\right] \le 1 - 1/p(n).$$

Then there exists a one-way function.

Proof Sketch. We verify that the standard proof for Theorem 23 also works in our setting. We use notations in [30, Theorem 2.3.2]. Let f be a candidate weak one-way function, and $m(n) := n^2 \cdot p(n) < 2^{o(n)}$. By [30, Theorem 2.3.2], we can construct a function g on $m(n)$ inputs bits, such that the following holds. Given any adversary B that inverts g w.p. $1/q(m)$, we can construct an adversary that makes $a(n) := 2n^2p(n)q(m(n))$ calls to B on input length $m(n)$, and inverts f w.p. $1 - 1/p(n)$.

Suppose that g is not a one-way function. Then there is a polynomial q and an adversary B that runs in $q(m)$ time and inverts g w.p. $1/q(m)$. We can invert f w.p. $1 - 1/p(n)$ by an adversary of time complexity

$$O(a(n) \cdot q(m(n))) < 2^{o(n)},$$

contradicting the hardness of f. ◀

6.1.2 Maximum Circuit Complexity

It will be convenient to fix an encoding of circuits into binary strings, so that we can sample a uniformly random circuit with a certain description length. Fortunately, such an encoding scheme naturally occurs in the lower bound proofs for the maximum circuit complexity, which usually use a counting argument [77, 28]: If every circuit of size $\mathrm{LB}(n)$ can be encoded as a string of length $2^n - 1$, then there must exist an n-bit Boolean function without size-$\mathrm{LB}(n)$ circuits.

In particular, in the lower bound proof of [28], the authors represented a circuit as a *stack program*. For a detailed description of stack programs, the reader is referred to [28]. We only need the following property of them:

▶ **Theorem 71.** *There is a constant c such that every size-s circuit on n inputs can be encoded into a stack program of bit-length $(s + 1)(c + \log(n + s))$.*

We also need the fact that given the description of a stack program, we can compute its truth table (the truth table of the circuit corresponding to it) in polynomial time.

We define

$$C'_{\mathsf{ub}}(n) := (C_{\mathsf{ub}}(n) + 1)(\log(n + C_{\mathsf{ub}}(n)) + O(1)) \le 2^n \left(1 + \frac{2\log n}{n} + \frac{O(1)}{n}\right).$$

By Theorem 71, every Boolean function over n inputs has a stack program of bit-length $C'_{\mathsf{ub}}(n)$.

We also need the following theorem, which says that for any Boolean function f on n input bits, there is a circuit of size roughly $2^n/n$ that computes f *simultaneously on multiple inputs*.

▶ **Theorem 72** ([83, 84]; see also [88, p. 304]). *Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function, r be a constant. There is a circuit C of size at most $(1 + o(1))2^n/n$ such that for every $x_1, x_2, \dots, x_r \in \{0,1\}^n$, $C(x_1, x_2, \dots, x_r) = f(x_1) \circ f(x_2) \circ \cdots \circ f(x_r)$.*

6.2 One-Way Functions from Hardness of MCSP

In this section, we construct a one-way function assuming MCSP is (exponentially) hard on average.

▶ **Theorem 73.** *Suppose that* MCSP *is exponentially hard on average. In particular, there is a constant $\epsilon > 0$ and a function $q(N) = 2^{o(N)}$, such that for every randomized algorithm $\mathcal{A}$ running in $2^{\epsilon N}$ time,*

$$\Pr_{\mathbf{tt} \leftarrow \mathcal{U}_N}[\mathcal{A}(\mathbf{tt}) = \mathsf{Size}(\mathbf{tt})] \le 1 - 1/q(N).$$

Then there exists a one-way function.

Proof. By Theorem 70, it suffices to construct a length-preserving function f that satisfies the following one-wayness property: There is a function $p(\tilde{N}) = 2^{o(\tilde{N})}$, such that for every integer $\tilde{N}$ and every randomized algorithm $\mathcal{A}$ that runs in $2^{\epsilon\tilde{N}/10}$ time,

$$\Pr_{\mathbf{x} \leftarrow \mathcal{U}_{\tilde{N}}}[\mathcal{A}(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))] \le 1 - 1/p(\tilde{N}). \tag{3}$$

Let $\tilde{N}$ be the input length of f, n be the largest integer such that $n + C'_{\mathsf{ub}}(n) \le \tilde{N}$. (Recall that every Boolean function over n inputs can be represented by a circuit, or stack program, of bit-length $C'_{\mathsf{ub}}(n)$.) The first n bits of the input denote an integer $s \le C_{\mathsf{ub}}(n)$, and the next $C'_{\mathsf{ub}}(n)$ bits denote a circuit C of size at most s. If the input is invalid (e.g., if $s > C_{\mathsf{ub}}(n)$ or the size of C is strictly larger than s), our function outputs $\bot$. Otherwise it outputs s and $tt(C)$, where $tt(C)$ is the length-2^n truth table of C. In other words, our weak one-way function is defined as follows:

$$f(s, C) = s \circ tt(C).$$

Let $\mathcal{A}_{\mathsf{owf}}$ be any candidate adversary that tries to invert f. We will construct an algorithm $\mathcal{A}_{\mathrm{MCSP}}$ based on $\mathcal{A}_{\mathsf{owf}}$ as in Algorithm 3. In particular, $\mathcal{A}_{\mathrm{MCSP}}$ attempts to solve MCSP on truth tables of length $N := 2^n$, using $\mathcal{A}_{\mathsf{owf}}$ that attempts to invert f on input length $\tilde{N}$. For large enough n, we have $\tilde{N} \le 2N$, thus if $\mathcal{A}_{\mathsf{owf}}$ runs in $2^{\epsilon\tilde{N}/10}$ time, then $\mathcal{A}_{\mathrm{MCSP}}$ runs in $2^{\epsilon N}$ time. Then, by the hardness of MCSP, $\mathcal{A}_{\mathrm{MCSP}}$ does not compute the circuit complexity correctly on a significant fraction of truth tables. Based on that, we can show that $\mathcal{A}_{\mathsf{owf}}$ satisfies Equation (3).

Algorithm 3 Bounded-Error Heuristic $\mathcal{A}_{\mathrm{MCSP}}$ for MCSP from Inverter $\mathcal{A}_{\mathsf{owf}}$ for f.

```
function $\mathcal{A}_{\mathrm{MCSP}}(tt)$
    $opt \leftarrow +\infty$
    for $s \in [C_{\mathsf{ub}}(n)]$ do
        $(s', C) \leftarrow \mathcal{A}_{\mathsf{owf}}(s, tt)$
        if $tt(C) = tt$ then
            $opt \leftarrow \min\{opt, |C|\}$
    return $opt$
```

Let Err be the set of truth tables $tt \in \{0,1\}^N$ on which $\mathcal{A}_{\mathrm{MCSP}}$ fails to output the correct answer w.p. $\ge 1/2q(N)$. By the hardness of MCSP and a Markov bound, we have

$$|\mathsf{Err}|/2^N \ge 1 - \frac{1 - 1/q(N)}{1 - 1/2q(N)} \ge \frac{1}{2q(N) - 1}.$$

We can see that $\mathcal{A}_{\mathsf{owf}}$ fails on every input of the form $(\mathsf{Size}(tt), tt)$ where $tt \in \mathsf{Err}$, also w.p. $\ge 1/2q(N)$. Every such input is generated in the OWF experiment w.p. at least $1/2^{C'_{\mathsf{ub}}(n)+n}$. That is:

$$\Pr[f(\mathcal{U}_{\tilde{N}}) = (\mathsf{Size}(tt), tt)] \ge 1/2^{C'_{\mathsf{ub}}(n)+n}.$$

It follows that

$$\begin{aligned}\Pr_{\mathbf{x} \leftarrow \mathcal{U}_{\tilde{N}}}[\mathcal{A}_{\mathsf{owf}}(f(\mathbf{x})) \notin f^{-1}(f(\mathbf{x}))] &\ge (|\mathsf{Err}|/2^{C'_{\mathsf{ub}}(n)+n}) \cdot (1/2q(N)) \\ &\ge (|\mathsf{Err}|/2^{N+O\left(\frac{N \log\log N}{\log N}\right)}) \cdot (1/2q(N)) \\ &\ge \frac{1}{(2q(N)-1)2q(N)2^{O\left(\frac{N\log\log N}{\log N}\right)}}.\end{aligned}$$

Let $p(\tilde{N}) := (2q(N) - 1)2q(N)2^{O\left(\frac{N\log\log N}{\log N}\right)}$. It is indeed the case that $p(\tilde{N}) = 2^{o(\tilde{N})}$, since $N = 2^n \ge \Omega(\tilde{N})$, and $q(N) = 2^{o(N)}$. We can see that every adversary $\mathcal{A}_{\mathsf{owf}}$ that runs in $2^{\epsilon\tilde{N}/10}$ time fails to invert a random output of f w.p. $\ge p(\tilde{N})$. ◀

6.3 Hardness of MCSP from DLOGTIME One-Way Functions

We establish a weak converse of Theorem 73. We show that if there is an exponentially hard weak one-way function in $\mathsf{DLOGTIME}$, then MCSP is (also exponentially) hard on average.

▶ **Theorem 74.** *Suppose that there is a weak one-way function f computable in* $\mathsf{DLOGTIME}$ *with security $2^{\Omega(N)}$. (See Definition 49.) Then, no nonuniform algorithm of size $2^{o(N)}$ can solve* MCSP *on a $1 - 2^{-o(N)}$ fraction of inputs.*

Proof. Fix an input length M. Let $\delta := 1/\text{poly}(M)$, so there is a constant κ_1 such that every adversary of size $2^{\kappa_1 M}$ fails to invert f on a $1 - \delta$ fraction of inputs. We construct a condEP-PRG G according to Construction 56. The stretch of G is $\alpha := \kappa_2 M$ for some small enough constant $\kappa_2 > 0$. Its outputs are 4ϵ-indistinguishable from true random strings, where $\epsilon := 1/M^{10}$. We use the hash functions in Section 5.4 as the extractors and hardcore functions. Note that the list size of the hardcore function is $L := 2^{O(\alpha)}$. Still, for some positive constant $\kappa_3 = \kappa_1 - O(\kappa_2) > 0$, no adversary of size $2^{\kappa_3 M}$ could 4ϵ-distinguish the outputs of G from random strings.

Let $\tilde{G}$ denote the randomized encoding of G (as in Section 5.4.1). Then, $\tilde{G}$ is a $\mathsf{DLOGTIME}$-computable condEP-PRG that maps KM input bits to $(K + \kappa_2)M$ input bits, where K is some absolute constant. W.l.o.g. we may assume that KM is a power of 2 (by padding a random string to both the input and output of $\tilde{G}$). Again, no adversary of size $2^{\kappa_3 M}$ could 4ϵ-distinguish the outputs of $\tilde{G}$ from random strings. It suffices to prove that the outputs of $\tilde{G}$, when viewed as truth tables (and padded to length $2KM$), have non-trivial circuit complexity. (As a result, if MCSP can be solved by a size-$2^{o(M)}$ circuit on average, then $\tilde{G}$ is not exponentially secure.)

Now, let $N := KM$, $n := \log N$, and $\kappa_4 := \frac{\kappa_2}{K}$, then $\tilde{G}$ is a condEP-PRG that maps N input bits to $(1 + \kappa_4)N$ input bits. Let $tt^{\mathsf{in}} \in \{0,1\}^N$ be an input, $tt^{\mathsf{out}} \in \{0,1\}^{2N}$ be the string whose first $(1 + \kappa_4)N$ bits are $\tilde{G}(tt^{\mathsf{in}})$, and other bits are zero.

▷ Claim 75. $\mathsf{Size}(tt^{\mathsf{out}}) \le (1 + o(1))2^n/n$.

Proof. Let r be a constant such that $\tilde{G}$ is a non-adaptive function that makes r queries to its input. That is, on input i, $\tilde{G}(tt^{\mathsf{in}})$ computes the indices $q(i,1)$, $q(i,2)$, $\ldots$, $q(i,r)$, queries $(tt^{\mathsf{in}})_{q(i,j)}$ for each $1 \le j \le r$, and computes $(tt^{\mathsf{out}})_i$ based on these answers. Note that every $\mathsf{DLOGTIME}$ machine making r *adaptive* queries is equivalent to a $\mathsf{DLOGTIME}$ machine making 2^r *non-adaptive* queries, thus it is without loss of generality to assume $\tilde{G}$ is non-adaptive.

By Theorem 72, there is a circuit C of size $(1 + o(1))2^n/n$ that on input $(x_1, x_2, \ldots, x_r)$, outputs the concatenation of $(tt^{\mathsf{in}})_{x_1}$, $(tt^{\mathsf{in}})_{x_2}$, $\ldots$, $(tt^{\mathsf{in}})_{x_r}$. We design a circuit for tt^{out} as follows.

- On input i, if $i > (1 + \kappa_4)N$, then output 0.
- Otherwise we simulate $\tilde{G}(i)$ to obtain the indices $q(i,j)$ for every $1 \le j \le r$. This step takes $O(n)$ time, and thus can be implemented in size poly(n).
- Use the circuit C of size $(1 + o(1))2^n/n$ to obtain $(tt^{\mathsf{in}})_{q(i,j)}$ for every $1 \le j \le r$.
- Finally, we can simulate $\tilde{G}(i)$ to obtain $(tt^{\mathsf{out}})_i$. Again, this step can be implemented in size poly(n).

It follows that the circuit complexity of tt^{out} is at most $(1 + o(1))2^n/n$. ◁

On the other hand, let $r \in \{0,1\}^{(1+\kappa_4)N}$ be a truly random string. We also append zeros in the end of r to make it a truth table of length $2N$. Denote $\mathsf{Size}(r)$ the circuit complexity of this length-$2N$ truth table. Let $\kappa_5 := \kappa_4/10$, and $s := (1 + \kappa_5)2^n/n$. By Theorem 71, the number of strings r such that $\mathsf{Size}(r) \le s$ is at most

$$2^{(s+1)(O(1)+\log(n+s))} \le 2^{(1+2\kappa_5)N} \ll 2^{(1+\kappa_4)N}.$$

It follows that with overwhelming probability, for a random string $r \in \{0,1\}^{(1+\kappa_4)N}$, $\mathsf{Size}(r) \geq (1+\kappa_5)2^n/n$. If we can solve MCSP by a nonuniform algorithm of size $2^{o(N)}$, then $\tilde{G}$ would not be a secure condEP-PRG. ◀

▶ Remark 76. Theorem 73 and 74 are not exactly converses of each other, as there are two gaps. First, there is a loss of $2^{O\left(\frac{N \log\log N}{\log N}\right)}$. Second, Theorem 73 only produces a (polynomial-time computable) one-way function, but Theorem 74 requires a DLOGTIME-computable one-way function to start with.

The first gap seems unavoidable given current knowledge about the maximum circuit complexity. However, we believe that the second gap can be eliminated. In particular, exponential average-case hardness of MCSP should imply a one-way function in DLOGTIME.

If there is a $\oplus$L *heuristic* algorithm for evaluating the truth table of a stack program, then it is indeed true that exponential hardness of MCSP implies a one-way function in DLOGTIME. Note that this heuristic only needs to succeed on *most* stack programs.[18] For example, if the circuit that corresponds to a uniformly random description has depth at most $O(\log n)$ with high probability, then a $\oplus$L heuristic can evaluate the circuit up to a particular depth, and still be correct on most inputs. We believe that a random stack program should represent a shallow circuit (w.h.p.), but we are unable to prove it.

▶ Remark 77 (Results for MFSP). It is possible to extend Theorem 73 to the case of MFSP (Minimum Formula Size Problem). In particular, suppose that MFSP is exponentially hard on average, then there is a (super-polynomially hard) one-way function. Moreover, we only need to compute truth tables of *formulas* to evaluate this one-way function, which is in ALOGTIME [19], hence this one-way function is in ALOGTIME, and we obtain DLOGTIME-computable one-way functions from Theorem 32. We omit the proof here, as it is essentially the same as Theorem 73 except that it uses the best bounds for maximum formula complexity (see [57] and references therein).

However, we are not aware of any "mass production theorem" (Theorem 72) for formulas. Therefore we are not able to prove an MFSP-version of Theorem 74.

7 The Average-Case Complexity of MKtP

7.1 Characterizing One-Way Functions Using MKtP

We recall the main result of [62] showing an equivalence between the average-case hardness of K^p for some polynomial p and the existence of one-way functions.

▶ **Theorem 78** ([62]). *The following are equivalent:*

1. *There is a polynomial p such that K^p is bounded-error hard on average.*
2. *One-way functions exist.*
3. *For every polynomial $p(n) \geq 2n$ and constant $\lambda > 0$, K^p is bounded-error hard on average to approximate within an additive factor of $\lambda \log n$.*

Somewhat counter-intuitively, we show that a similar equivalence between the average-case hardness of Kt and the existence of one-way functions. (Note that Kt is known to be EXP-hard in the worst case under polynomial-size reductions [4].) The proof is very closely analogous to the proofs in Section 4, exploiting the fact that "typical" strings of high Kt complexity can be generated from their optimal descriptions in polynomial time. Hence we just provide a sketch.

[18] If this heuristic is always true (i.e. it is a worst-case algorithm), then $\oplus\mathsf{L} = \mathsf{P}$.

▶ **Theorem 79.** *The following are equivalent:*

1. Kt *is bounded-error hard on average.*
2. *One-way functions exist.*
3. *For every constant $\lambda > 0$,* Kt *is bounded-error hard on average to approximate within an additive factor of $\lambda \log n$.*

Proof Sketch. (3) $\implies$ (1) is trivial.

(1) $\implies$ (2): the proof closely follows the proof of Theorem 33.

Suppose that there is a constant c such that every PPT algorithm computes Kt complexity correctly on at most a $1-1/n^c$ fraction of inputs. We observe that for all but a $1/n^{2c}$ fraction of inputs $x \in \{0,1\}^n$, optimal pairs (d,t) such that $d + \log t = \mathrm{Kt}(x)$ have the property that $t \le O(n^{2c+1})$. Actually, for all but a $1/n^{2c}$ fraction of inputs, $\mathrm{K}(x) \ge n - 2c\log n - 1$, while for all inputs x we have $\mathrm{Kt}(x) \le n + \log n + O(1)$. Hence for all strings x with $\mathrm{K}(x) \ge n - 2c\log n - 1$, x can be generated from its optimal description in time $O(n^{2c+1})$.

We define a weak one-way function f as follows. It takes as input a triple (ℓ, k, M), where $\ell \in [n + \log n]$, $k \in [(2c+1)\log n]$, and $M \in \{0,1\}^{n+\log n}$, and outputs (ℓ, k, out). Here out is the result of running $U^{M'}$ for at most 2^k steps, where M' is the ℓ-bit prefix of M. Just as in Claim 35, the output distribution of f on a uniformly chosen input "almost dominates" the uniform distribution. Assume there is an inverter for f, a heuristic algorithm for the search version of $\mathrm{Kt}(x)$ can cycle over all possible ℓ and k, and find the optimal description of x. As Kt is bounded-error hard on average, our candidate one-way function f is secure.

(2) $\implies$ (3): We use Theorem 25 to construct a condEP-PRG G with stretch $\gamma \log n$ and security $1/n^\gamma$ from the presumed one-way function. Here, let c be a constant such that G is computable in time n^c, we choose $\gamma = \lambda + c + 2$.

We use an argument closely analogous to that of Lemma 39 to show that Kt is bounded-error hard on average to approximate within an additive factor of $\lambda \log n$. The idea is simple: every output of the condEP-PRG has Kt complexity at most $n + c\log n + O(1)$, while a random string of length $n + \gamma \log n$ is likely to have Kt complexity close to $n + \gamma \log n$. Hence, for our choice of parameters, an efficient heuristic algorithm that approximates Kt complexity within an additive factor of $\lambda \log n$ can distinguish the outputs of G from random. ◀

Theorem 78 and Theorem 79 yield the following corollary.

▶ **Corollary 80.** Kt *is bounded-error hard on average iff there is a polynomial p such that* K^p *is bounded-error hard on average.*

Corollary 80 gives a new non-trivial connection between meta-complexity problems that seems hard to argue without using one-way functions as an intermediate notion.

7.2 A Complexity Theoretic Analogue

Theorem 79 shows that the weak average-case hardness of Kt is equivalent to the existence of cryptographic pseudo-random generators. We next show that for a slightly different setting of parameters, the average-case hardness of Kt is equivalent to the existence of complexity-theoretic pseudo-random generators against non-uniform adversaries. Thus average-case complexity of a single natural problem, namely Kt, can be used to characterize both cryptographic pseudorandomness and complexity-theoretic pseudorandomness.

Recall that cryptographic PRGs are required to be computable in fixed polynomial time but to be secure against adversaries that can run within any polynomial time bound. In contrast, complexity-theoretic PRGs are allowed to use more resources than the adversary.

▶ **Definition 81.** *Given functions $t : \mathbb{N} \to \mathbb{N}$, $\ell : \mathbb{N} \to \mathbb{N}$ (satisfying $\ell(n) \leq n$ for each n) and $s : \mathbb{N} \to \mathbb{N}$, we say that a family of functions $\{G_n\}$, where $G_n : \{0,1\}^{\ell(n)} \to \{0,1\}^n$ is a time t pseudo-random generator (PRG) with seed length ℓ against size s if $G(z)$ is computable in time $t(|z|)$ and for each n, $G_n(\mathcal{U}_{\ell(n)})$ is $1/s(n)$-indistinguishable from $\mathcal{U}_n$ by size $s(n)$ circuits. The PRG is said to be seed-extending if z is a prefix of $G(z)$ for each seed z.*

Nisan and Wigderson [69, 13] showed how to base seed-extending complexity-theoretic PRGs on the hardness of E (exponential-time). The parameters in the following theorem statement are implicit in their main result.

▶ **Theorem 82** ([69, 13])**.** *If $\mathsf{DTIME}(2^n \text{poly}(n)) \not\subseteq \mathsf{P}_{/\text{poly}}$, then for each ℓ such that $\ell(n) = n^{\Omega(1)}$, there is a seed-extending time $2^\ell \text{poly}(\ell)$ PRG with seed length ℓ against polynomial size.*

We use Theorem 82 to derive an equivalence between the worst-case hardness of Kt, the existence of complexity-theoretic PRGs with non-trivial seed length, and very mild average-case hardness of Kt, where the hardness is against non-uniform adversaries. The idea of the proof is similar to that of [4], who showed that computing Kt complexity is hard for exponential-time under polynomial-size reductions.

▶ **Theorem 83.** *The following are equivalent:*

1. $\mathsf{EXP} \not\subseteq \mathsf{P}_{/\text{poly}}$.
2. *For each $\epsilon > 0$, there is a time $2^\ell \text{poly}(\ell)$ PRG with seed length n^ϵ against polynomial size.*
3. *There are no polynomial size circuits for* Kt.
4. *For each $\epsilon > 0$, there is a seed-extending time $2^\ell \text{poly}(\ell)$ PRG with seed length n^ϵ against polynomial size.*
5. *For any constant $\delta > 1/2$, there are no polynomial size circuits computing* Kt *on a $1 - 1/2^{\delta n}$ fraction of inputs.*

Proof. (1) $\iff$ (3) is shown in [4].

(1) $\iff$ (2) is shown in [69, 13].

(5) $\implies$ (3) is trivial.

(3) $\implies$ (4): We use Theorem 82. Kt can be computed in time $O(2^n \text{poly}(n))$, and we can define a decision version of Kt that is equivalent to the search version and computable in time $2^n \text{poly}(n)$ as follows: For $x \in \{0,1\}^n$ and $k \in [n + \log n]$, (x, k) is a Yes instance of the decision version of Kt iff $\text{Kt}(x) \leq k$. By Theorem 82, the hardness of the decision version implies that for each $\epsilon > 0$, there is a seed-extending time $2^m \text{poly}(m)$ PRG with seed length n^ϵ against polynomial size.

(4) $\implies$ (5): Consider a seed-extending $2^m \text{poly}(m)$ time PRG $G = \{G_n\}$ with seed length γn, where $1/2 > \gamma > 1 - \delta$. Such a PRG is implied by a PRG with smaller seed length, simply by truncating the output. Since the seed length is γn and the PRG is computable in time $2^{\gamma n}\text{poly}(n)$, we have that each output of the PRG has Kt complexity at most $2\gamma n + O(\log n) < n - \log n$. On the other hand, a uniformly chosen input of length n has Kt complexity very close to n, with high probability.

Suppose that there are polynomial size circuits $\{C_n\}$ computing Kt on a $1 - 1/2^{\delta n}$ fraction of inputs. By our choice of δ, this means that they are correct on at least a $2/3$ fraction of strings $G_n(z)$ for seed z of length γn. Now we can define a distinguisher D as follows: D computes $C_n(x)$ and accepts iff $C_n(x) \leq n - \log(n)$. D accepts with probability $2/3$ on $G_n(z)$ for uniformly chosen z, but with probability at most $1/3$ on x for uniformly chosen x of length n, since all but a $o(1)$ fraction of strings have $\text{Kt}(x) > n - \log(n)$ and C_n answers correctly on all but a $o(1)$ fraction of these strings with high Kt complexity. Therefore D is a distinguisher of polynomial size, contradicting the assumption that G is a PRG. ◀

8 Open Problems

We conclude this paper with a few open questions.

Perebor hypotheses. How plausible are the Perebor hypotheses in Section 5.5? We believe it is within reach to refute the non-uniform version of Hypothesis 66, by e.g. implementing the inverter in [27] as circuits.

It would be exciting to refute the other Strong Perebor Hypotheses. Let $t(n)$ be a polynomial (say $t(n) = 10n$ for simplicity). Is there a (probabilistic) algorithm running in $2^n/n^{\omega(1)}$ time that computes K^t in the worst-case? What about the average-case? Does such algorithm imply new circuit lower bounds, as in the case of SAT algorithms [90] and learning algorithms [72]? Is there a circuit family of $2^n/n^{\omega(1)}$ size that computes cK^t (on input length $2n = n + n$)?

The Strong Exponential Time Hypothesis is used extensively in *fine-grained complexity*. Conditioning on SETH, we can prove many polynomial lower bounds for problems in P (e.g. the Orthogonal Vectors problem requires $n^{2-o(1)}$ time [89]). Do the Strong Perebor Hypotheses imply non-trivial conditional lower bounds for natural problems in P?

Random circuits. Due to our limited knowledge about circuit complexity, the relations presented in Section 6 are not tight. We point out a few questions whose resolution would tighten the relationship between MCSP and one-way functions.

First, is there an efficiently samplable distribution over circuits, such that for most truth table $tt \in \{0,1\}^N$, the probability that the optimal circuit for tt is sampled is at least $2^{-N}/\mathrm{poly}(N)$? Such a distribution would imply a one-way function from super-polynomial hardness of MCSP. The trivial solution as presented in Section 6.2 is to sample a uniformly random circuit according to some encoding. The probability that the optimal circuit is sampled is $2^{-N}/2^{O\left(N\frac{\log\log N}{\log N}\right)}$.

Second, is there a $\oplus\mathsf{L}$ heuristic algorithm for evaluating a random circuit, that succeeds on *most* circuits? Of course, this depends on the exact definition of "random" circuits. Such a heuristic implies a $\mathsf{DLOGTIME}$-computable one-way function from hardness of MCSP, establishing a tighter converse of Theorem 74.

Last, does the existence of $\mathsf{DLOGTIME}$-computable one-way functions imply the hardness of MFSP? The main technical difficulty is that we do not have a formula version of Theorem 72.

Other cryptographic primitives? Meta-complexity can characterize the existence of one-way functions [62] and one-way functions in NC^0 (this paper). Is there a similar characterization for other cryptographic primitives, such as public-key encryption [75, 25], or indistinguishability obfuscation [14]?

Is there a meta-complexity characterization of exponentially-hard *strong* one-way functions? This would bring new insights to the old question of hardness amplification for one-way functions that preserve exponential security [31].

References

1 Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 99–108, 1996. doi:10.1145/237814.237838.

2 Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on NP-hardness. In *Proc. 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 701–710, 2006. doi:10.1145/1132516.1132614.

3 Eric Allender. When worlds collide: Derandomization, lower bounds, and Kolmogorov complexity. In *Proc. 21st Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 2245 of *Lecture Notes in Computer Science*, pages 1–15, 2001. doi:10.1007/3-540-45294-X_1.

4 Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal of Computing*, 35(6):1467–1493, 2006. doi:10.1137/050628994.

5 Eric Allender, Mahdi Cheraghchi, Dimitrios Myrisiotis, Harsha Tirumala, and Ilya Volkovich. One-way functions and a conditional variant of MKTP. *Electronic Colloquium on Computational Complexity (ECCC)*, 2021. URL: https://eccc.weizmann.ac.il/report/2021/009/.

6 Eric Allender and Shuichi Hirahara. New insights on the (non-)hardness of circuit minimization and related problems. *ACM Transactions on Computation Theory*, 11(4):27:1–27:27, 2019. doi:10.1145/3349616.

7 Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, 1992. doi:10.1109/18.119713.

8 Benny Applebaum. *Cryptography in Constant Parallel Time.* Information Security and Cryptography. Springer, 2014. doi:10.1007/978-3-642-17367-7.

9 Benny Applebaum. Cryptographic hardness of random local functions - survey. *Computational Complexity*, 25(3):667–722, 2016.

10 Benny Applebaum. Exponentially-hard Gap-CSP and local PRG via local hardcore functions. In *Proc. 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 836–847, 2017. doi:10.1109/FOCS.2017.82.

11 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM Journal of Computing*, 36(4):845–888, 2006. doi:10.1137/S0097539705446950.

12 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, 2009.

13 László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computatioanl Complexity*, 3:307–318, 1993. doi:10.1007/BF01275486.

14 Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6:1–6:48, 2012. doi:10.1145/2160158.2160159.

15 Joshua Baron, Yuval Ishai, and Rafail Ostrovsky. On linear-size pseudorandom generators and hardcore functions. *Theoretical Computer Science*, 554:50–63, 2014. doi:10.1016/j.tcs.2014.06.013.

16 Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions. In *Proc. 5th Theory of Cryptography Conference (TCC)* , volume 4948 of *Lecture Notes in Computer Science*, pages 55–72, 2008. doi:10.1007/978-3-540-78524-8_4.

17 Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. *SIAM Journal of Computing*, 36(4):1119–1159, 2006. doi:10.1137/S0097539705446974.

18 J. L. Bordewijk. Inter-reciprocity applied to electrical networks. *Applied Scientific Research, Section A*, pages 1–74, 1957. doi:10.1007/BF02410413.

19 Samuel R. Buss. The Boolean formula value problem is in ALOGTIME. In *Proc. 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 123–131, 1987. doi:10.1145/28395.28409.

20 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Parameterized and Exact Computation, 4th International Workshop, (IWPEC) 2009*, volume 5917 of *Lecture Notes in Computer Science*, pages 75–85. Springer, 2009. doi:10.1007/978-3-642-11269-0_6.

21 Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1807 of *Lecture Notes in Computer Science*, pages 453–469, 2000. doi:10.1007/3-540-45539-6_33.

22 Lijie Chen, Ce Jin, and R. Ryan Williams. Hardness magnification for all sparse NP languages. In *Proc. 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1240–1255, 2019. doi:10.1109/FOCS.2019.00077.

23 Lijie Chen and Hanlin Ren. Strong average-case lower bounds from non-trivial derandomization. In *Proc. 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1327–1334, 2020. doi:10.1145/3357713.3384279.

24 Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions (preliminary version). In *Proc. 26th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 396–407, 1985. doi:10.1109/SFCS.1985.55.

25 Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. doi:10.1109/TIT.1976.1055638.

26 Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On beating the hybrid argument. *Theory of Computing*, 9:809–843, 2013. doi:10.4086/toc.2013.v009a026.

27 Amos Fiat and Moni Naor. Rigorous time/space trade-offs for inverting functions. *SIAM Journal of Computing*, 29(3):790–803, 1999. doi:10.1137/S0097539795280512.

28 Gudmund Skovbjerg Frandsen and Peter Bro Miltersen. Reviewing bounds on the circuit size of the hardest functions. *Information Processing Letters*, 95(2):354–357, 2005. doi:10.1016/j.ipl.2005.03.009.

29 Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981. doi:10.1016/0022-0000(81)90040-4.

30 Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001. doi:10.1017/CBO9780511546891.

31 Oded Goldreich, Russell Impagliazzo, Leonid A. Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 318–326, 1990. doi:10.1109/FSCS.1990.89550.

32 Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proc. 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 25–32, 1989. doi:10.1145/73007.73010.

33 Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum. Verifying and decoding in constant depth. In *Proc. 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 440–449, 2007. doi:10.1145/1250790.1250855.

34 Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. $\mathsf{AC}^0[p]$ lower bounds against MCSP via the coin problem. In *Proc. 46th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 132 of *LIPIcs*, pages 66:1–66:15, 2019. doi:10.4230/LIPIcs.ICALP.2019.66.

35 Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 658–667, 2001. doi:10.1109/SFCS.2001.959942.

36 Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM*, 56(4):20:1–20:34, 2009. doi:10.1145/1538902.1538904.

37 Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM Journal of Computing*, 42(3):1405–1430, 2013. doi:10.1137/100814421.

38 Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal of Computing*, 28(4):1364–1396, 1999. doi:10.1137/S0097539793244708.

39 Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, 1980. doi:10.1109/TIT.1980.1056220.

40 Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *Proc. 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 247–258, 2018. doi:10.1109/FOCS.2018.00032.

41 Shuichi Hirahara. Characterizing average-case complexity of PH by worst-case meta-complexity. In *Proc. 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 50–60, 2020. doi:10.1109/FOCS46700.2020.00014.

42 Shuichi Hirahara. Unexpected hardness results for Kolmogorov complexity under uniform reductions. In *Proc. 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1038–1051, 2020. doi:10.1145/3357713.3384251.

43 Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In *Proc. 32nd Computational Complexity Conference (CCC)*, volume 79 of *LIPIcs*, pages 7:1–7:20, 2017. doi:10.4230/LIPIcs.CCC.2017.7.

44 Thomas Holenstein, Ueli M. Maurer, and Johan Sjödin. Complete classification of bilinear hard-core functions. In *Proc. 24th Annual International Cryptology Conference (CRYPTO)*, volume 3152 of *Lecture Notes in Computer Science*, pages 73–91. Springer, 2004. doi:10.1007/978-3-540-28628-8_5.

45 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, pages 439–561, 2006. doi:10.1090/S0273-0979-06-01126-8.

46 Rahul Ilango. Connecting perebor conjectures: Towards a search to decision reduction for minimizing formulas. In *Proc. 35th Computational Complexity Conference (CCC)*, volume 169 of *LIPIcs*, pages 31:1–31:35, 2020. doi:10.4230/LIPIcs.CCC.2020.31.

47 Rahul Ilango. Constant depth formula and partial function versions of MCSP are hard. In *Proc. 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 424–433, 2020. doi:10.1109/FOCS46700.2020.00047.

48 Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. NP-hardness of circuit minimization for multi-output functions. In *Proc. 35th Computational Complexity Conference (CCC)*, volume 169 of *LIPIcs*, pages 22:1–22:36, 2020. doi:10.4230/LIPIcs.CCC.2020.22.

49 Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proc. 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 538–545, 1995. doi:10.1109/SFCS.1995.492584.

50 Russell Impagliazzo. A personal view of average-case complexity. In *Proc. 10th Annual Structure in Complexity Theory Conference*, pages 134–147, 1995. doi:10.1109/SCT.1995.514853.

51 Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 812–821, 1990. doi:10.1109/FSCS.1990.89604.

52 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

53 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

54 Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *Proc. 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 294–304, 2000. doi:10.1109/SFCS.2000.892118.

55 Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *Proc. 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 244–256, 2002. doi:10.1007/3-540-45465-9_22.

56 Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 433–442, 2008. doi:10.1145/1374376.1374438.

57 Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-24508-4.

58 Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *Proc. 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 73–79, 2000. doi:10.1145/335305.335314.

59 Ker-I Ko. On the complexity of learning minimum time-bounded Turing machines. *SIAM Journal of Computing*, 20(5):962–986, 1991. doi:10.1137/0220059.

60 Leonid A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61(1):15–37, 1984. doi:10.1016/S0019-9958(84)80060-1.

61 Leonid A. Levin. The tale of one-way functions. *Problems of Information Transmission*, 39(1):92–103, 2003.

62 Yanyi Liu and Rafael Pass. On one-way functions and Kolmogorov complexity. In *Proc. 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1243–1254, 2020. doi:10.1109/FOCS46700.2020.00118.

63 Yanyi Liu and Rafael Pass. On the possibility of basing cryptography on $\mathsf{EXP} \neq \mathsf{BPP}$. *Electronic Colloquium on Computational Complexity (ECCC)*, 28:56, 2021. URL: https://eccc.weizmann.ac.il/report/2021/056.

64 G. A. Margulis. Explicit constructions of concentrators. *Probl. Peredachi Inf.*, pages 71–80, 1973.

65 Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Proc. 51st Annual ACM Symposium on Theory of Computing (STOC)*, pages 1215–1225, 2019. doi:10.1145/3313276.3316396.

66 Cody D. Murray and R. Ryan Williams. On the (non) NP-hardness of computing circuit complexity. *Theory of Computing*, 13(1):1–22, 2017. doi:10.4086/toc.2017.v013a004.

67 Mikito Nanashima. On basing auxiliary-input cryptography on NP-hardness via nonadaptive black-box reductions. In *Proc. 12th Conference on Innovations in Theoretical Computer Science (ITCS)*, volume 185 of *LIPIcs*, pages 29:1–29:15, 2021. doi:10.4230/LIPIcs.ITCS.2021.29.

68 Noam Nisan. Extracting randomness: How and why. A survey. In *Proc. 11th Annual IEEE Conference on Computational Complexity (CCC)*, pages 44–58, 1996. doi:10.1109/CCC.1996.507667.

69 Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

70 Noam Nisan and David Zuckerman. More deterministic simulation in logspace. In *Proc. 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 235–244, 1993. doi:10.1145/167088.167162.

71 Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. In *Proc. 34th Computational Complexity Conference (CCC)*, volume 137 of *LIPIcs*, pages 27:1–27:29, 2019. doi:10.4230/LIPIcs.CCC.2019.27.

72 Igor Carboni Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Proc. 32nd Computational Complexity Conference (CCC)*, volume 79 of *LIPIcs*, pages 18:1–18:49, 2017. doi:10.4230/LIPIcs.CCC.2017.18.

73 Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *Proc. 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 65–76, 2018. doi:10.1109/FOCS.2018.00016.

74 Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997. doi:10.1006/jcss.1997.1494.

75 Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. doi:10.1145/359340.359342.

76 Rahul Santhanam. Pseudorandomness and the minimum circuit size problem. In *Proc. 11th Conference on Innovations in Theoretical Computer Science (ITCS)*, volume 151 of *LIPIcs*, pages 68:1–68:26, 2020. doi:10.4230/LIPIcs.ITCS.2020.68.

77 Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell System technical journal*, 28(1):59–98, 1949. doi:10.1002/j.1538-7305.1949.tb03624.x.

78 Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996. doi:10.1109/18.556668.

79 Amnon Ta-Shma and David Zuckerman. Extractor codes. *IEEE Transactions on Information Theory*, 50(12):3015–3025, 2004. doi:10.1109/TIT.2004.838377.

80 Roei Tell. Quantified derandomization of linear threshold circuits. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 855–865, 2018. doi:10.1145/3188745.3188822.

81 Boris A. Trakhtenbrot. A survey of Russian approaches to perebor (brute-force searches) algorithms. *IEEE Annals of the History of Computing*, 6(4):384–400, 1984. doi:10.1109/MAHC.1984.10036.

82 Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001. doi:10.1145/502090.502099.

83 D. Uhlig. On the synthesis of self-correcting schemes from functional elements with a small number of reliable elements. *Mathematical notes of the Academy of Sciences of the USSR*, 15:558–562, 1974. doi:10.1007/BF01152835.

84 D. Uhlig. Zur parallelberechnung boolescher funktionen. *TR Ing.hochsch. Mittweida*, 1984.

85 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012. doi:10.1561/0400000010.

86 Salil P. Vadhan and Colin Jia Zheng. A uniform min-max theorem with applications in cryptography. In *Proc. 33rd Annual International Cryptology Conference (CRYPTO)*, volume 8042 of *Lecture Notes in Computer Science*, pages 93–110, 2013. doi:10.1007/978-3-642-40041-4_6.

87 Hoeteck Wee. Finding Pessiland. In *Proc. 3rd Theory of Cryptography Conference (TCC)* , volume 3876 of *Lecture Notes in Computer Science*, pages 429–442, 2006. doi:10.1007/11681878_22.

88 Ingo Wegener. *The complexity of Boolean functions.* Wiley-Teubner, 1987. URL: http://ls2-www.cs.uni-dortmund.de/monographs/bluebook/.

89 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2-3):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.

90 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal of Computing*, 42(3):1218–1244, 2013. doi:10.1137/10080703X.

91 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proc. of the ICM*, volume 3, pages 3431–3472, 2018.

92 Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 80–91, 1982. doi:10.1109/SFCS.1982.45.

93 Yu Yu, Xiangxue Li, and Jian Weng. Pseudorandom generators from regular one-way functions: New constructions with improved parameters. *Theoretical Computer Science*, 569:58–69, 2015. doi:10.1016/j.tcs.2014.12.013.

A Proof of Theorem 57

▶ **Theorem 57.** *Let $\epsilon, \delta, \alpha, f$ be defined as in Construction 56. If $\epsilon \geq 1/\mathrm{poly}(n)$ and $L \leq \mathrm{poly}(n)$, then there is a function $r : \mathbb{N} \to \mathbb{N}$ such that $G = \{G_{n,r(n)}\}_{n\in\mathbb{N}}$ is a condEP-PRG with stretch α and security 4ϵ.*

More precisely, let $\tilde{n} = n + 2d + d'$. Suppose that for every subset $\mathcal{D} \subseteq \{0,1\}^{\tilde{n}}$ such that $\mathrm{H}(G(\mathcal{D})) \geq \tilde{n} - \Omega(\log(\frac{n}{\delta\epsilon}))$ and every k, there is an adversary of size s that 4ϵ-distinguishes $G_{n,k}(\mathcal{D})$ from the uniform random distribution. Then there is an adversary of size $s \cdot \mathrm{poly}(nL/\epsilon)$ that inverts f on a $1-\delta$ fraction of inputs.

For convenience, we only consider non-uniform adversaries in this section. (See also Remark 52.) Recall that we sometimes use a (multi-)set $\mathcal{S}$ to represent the uniform distribution over $\mathcal{S}$, and we assume that every one-way function is length-preserving.

A.1 Impagliazzo's Hardcore Lemma

▶ **Lemma 84** ([49]; see e.g., [12]). *Let f be a candidate (weak) one-way function, $\epsilon, \delta > 0$. Suppose for every $\mathcal{E} \subseteq \{0,1\}^n$ with $|\mathcal{E}| \geq \frac{\delta}{2} \cdot 2^n$, there is a circuit C of size $s(n)$ such that*

$$\Pr_{\mathbf{x}\leftarrow\mathcal{E}}[C(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))] \geq \epsilon.$$

Then there is a circuit of size $O(s(n) \cdot n\epsilon^{-2})$ that inverts f on a $1-\delta$ fraction of inputs.

A.2 Step I: Making f Strong and Regular

Let f be a weak one-way function. The first step is to transform f into a strong and regular one-way function, but only under a certain input distribution. In particular, we will define a sequence of subsets $\mathcal{X} = \{\mathcal{X}_n\}$ (that is not necessarily easy to sample), such that on the uniform distribution over $\mathcal{X}_n$, f is both *strong* and *regular*. Here:

- We say f is *strong* on $\mathcal{X}$, if every polynomial-size adversary $\mathcal{A}$ fails to invert $f(\mathcal{X})$ except with negligible probability. (For comparison, we are only given that f is a *weak* one-way function on a uniform random input: No PPT adversary inverts f on *a* $(1-1/n^c)$ *fraction of inputs*, for some fixed constant $c > 0$.)
- For a function $r : \mathbb{N} \to \mathbb{N}$, we say f is r*-regular* on $\mathcal{X}$, if for every $n \in \mathbb{N}$ and every $y \in f_n(\mathcal{X}_n)$, we have $|f_{\mathcal{X}}^{-1}(y)| \in [2^{r(n)-1}, 2^{r(n)}]$. Here, $f_{\mathcal{X}}^{-1}(y) = \{x \in \mathcal{X} : f(x) = y\}$, and $|f_{\mathcal{X}}^{-1}(y)|$ denotes the size of the above set.

As discussed in Section 5.2, we use the hardcore lemma to find a subset of inputs on which f is strong. In particular, applying Lemma 84, we have:

▷ Claim 85. There is a sequence of subsets $\mathcal{X}' = \{\mathcal{X}'_n \subseteq \{0,1\}^n\}$ with $|\mathcal{X}'_n| \geq 2^n/\mathrm{poly}(n)$, such that for every polynomial-size adversary $\mathcal{A}$,

$$\Pr_{\mathbf{x}\leftarrow\mathcal{X}'_n}[\mathcal{A}(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))] < \mathrm{negl}(n).$$

More precisely, suppose that for every subset $\mathcal{X}'_n \subseteq \{0,1\}^n$ with $|\mathcal{X}'_n| \geq \frac{\delta}{2} \cdot 2^n$, there is an adversary of size s that inverts $f_n(\mathcal{X}'_n)$ w.p. at least ϵ. Then there is an adversary of size $O(s \cdot n\epsilon^{-2})$ that inverts f on a $1-\delta$ fraction of inputs. ◀

Now, for every string $y \in \{0,1\}^n$, let $|f^{-1}_{\mathcal{X}'}(y)|$ denote the number of inputs $x \in \mathcal{X}'_n$ such that $f(x) = y$. Let $r \in [1,n]$, W_r be the number of strings $x \in \mathcal{X}'_n$ such that $|f^{-1}_{\mathcal{X}'}(f(x))| \in [2^{r-1}, 2^r]$. Then we have $\sum_{r=1}^{n} W_r \geq |\mathcal{X}'_n|$. By averaging, there is an integer $r \in [1,n]$ such that $W_r \geq |\mathcal{X}'_n|/n$. We denote $r(n)$ to be this integer r, and

$$\mathcal{X}_n := \{x \in \mathcal{X}'_n : |f^{-1}_{\mathcal{X}'}(f(x))| \in [2^{r-1}, 2^r]\}.$$

By definition, f is r-regular on $\mathcal{X} := \{\mathcal{X}_n\}$. Since $|\mathcal{X}_n| \geq |\mathcal{X}'_n|/n$, any adversary that inverts $\mathcal{X}_n$ on an ϵ fraction of inputs also inverts $\mathcal{X}'_n$ on an ϵ/n fraction of inputs. To summarize:

▷ Claim 86. There is a function $r(n) \leq n$ and a sequence of subsets $\mathcal{X} = \{\mathcal{X}_n\}$ with $|\mathcal{X}_n| \geq 2^n/\mathrm{poly}(n)$, such that f is r-regular on $\mathcal{X}$, and for every polynomial-size adversary $\mathcal{A}$,

$$\Pr_{\mathbf{x} \leftarrow \mathcal{X}_n}[\mathcal{A}(f(\mathbf{x})) \in f^{-1}(f(\mathbf{x}))] < \mathrm{negl}(n).$$

More precisely, suppose that for every function $r : \mathbb{N} \to \mathbb{N}$ and sequence of subsets $\mathcal{X} = \{\mathcal{X}_n\}$ such that $|\mathcal{X}_n| \geq \frac{\delta}{2n} \cdot 2^n$ and f is r-regular on $\mathcal{X}$, there is an adversary of size s that inverts $f_n(\mathcal{X}_n)$ w.p. at least ϵ. Then there is an adversary of size $s \cdot \mathrm{poly}(n/\epsilon)$ that inverts f on a $1 - \delta$ fraction of inputs. ◀

A.3 Step II: An Intermediate Function

We define another function ensemble $\tilde{f} = \{\tilde{f}_n\}_{n \in \mathbb{N}}$. Let $k_1 = r - 1$, $k_2 = \lfloor n - r - \log(2n/\delta) \rfloor$, and $d = d_{\mathsf{Ext}}(n, \epsilon)$. We need the following two extractors:

- a strong (k_1, ϵ)-extractor $\mathsf{Ext}_1 : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{m_1}$, where $m_1 := k_1 - 2\log(1/\epsilon) - O(1)$;
- a strong (k_2, ϵ)-extractor $\mathsf{Ext}_2 : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{m_2}$, where $m_2 := k_2 - 2\log(1/\epsilon) - O(1)$.

The function $\tilde{f}_n : \{0,1\}^n \times \{0,1\}^{2d} \to \{0,1\}^{m_1+m_2+2d}$ is defined as follows.

$$\tilde{f}_n(x, z_1, z_2) := z_1 \circ \mathsf{Ext}_1(x, z_1) \circ z_2 \circ \mathsf{Ext}_2(x, z_2).$$

Denote $\ell(n) := m_1 + m_2 + 2d$. The following lemma summarizes the properties of $\tilde{f}_n$ we need:

▶ **Lemma 87.** *For every integer n, the function $\tilde{f}_n$ satisfies the following properties:*

1. *(Uniformity) For every integer n, $\mathsf{SD}(\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d}), \mathcal{U}_{\ell(n)}) \leq 2\epsilon$.*
2. *(Hiding) For every polynomial-size adversary $\mathcal{A}$ and every integer n,*

$$\Pr_{\mathbf{x} \leftarrow \mathcal{X}_n}\left[\mathcal{A}(\tilde{f}_n(\mathbf{x}, \mathcal{U}_{2d})) = \mathbf{x}\right] \leq \mathrm{negl}(n).$$

More precisely, if there is an adversary $\mathcal{A}$ of size s that on input $\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d})$, guesses $\mathcal{X}_n$ with success probability γ, then there is an adversary $\mathcal{A}'$ of size $O(s)$ that inverts $f_n(\mathcal{X}_n)$ w.p. at least $O(\gamma/\epsilon^2)$.

Proof. (Uniformity) A sample from $\mathcal{X}_n$ can be obtained from two steps. First, we sample a string y_0 with probability $p(y_0) := \Pr_{\mathbf{x} \leftarrow \mathcal{X}_n}[f_n(\mathbf{x}) = y_0]$. Then we sample a string x_0 with probability $p(x_0 \mid y_0) := \Pr_{\mathbf{x} \leftarrow \mathcal{X}_n}[\mathbf{x} = x_0 \mid f_n(\mathbf{x}) = y_0]$.

Suppose y_0 is fixed. Since f is r-regular, we have $|f^{-1}_{\mathcal{X}}(y_0)| \geq 2^{r-1}$. Therefore, conditioned on y_0, the min-entropy of the distribution of x_0 is at least $r - 1 \geq k_1$.

Let $\mathbf{x} \leftarrow \mathcal{X}_n$ and $\mathbf{z}_1 \leftarrow \mathcal{U}_d$. Since Ext_1 is a strong (k_1, ϵ)-extractor, we have

$$\mathsf{SD}(\mathbf{z}_1 \circ \mathsf{Ext}_1(\mathbf{x}, \mathbf{z}_1), \mathcal{U}_{d+m_1} \mid f(\mathbf{x})) \le \epsilon.$$

Now, for every $y_0 \in f_n(\mathcal{X}_n)$, since $|f_{\mathcal{X}}^{-1}(y_0)| \le 2^r$, the probability that a sample of $f_n(\mathcal{X}_n)$ is equal to the particular y_0 is at most $2^r/|\mathcal{X}_n|$. It follows that the min-entropy of the distribution of y_0 is at least $\log(|\mathcal{X}_n|/2^r) \ge n - r + \log(\delta/2n) \ge k_2$. Since Ext_2 is a strong (k_2, ϵ)-extractor, we have

$$\mathsf{SD}(\mathbf{z}_2 \circ \mathsf{Ext}_2(f(\mathbf{x}), \mathbf{z}_2), \mathcal{U}_{d+m_2}) \le \epsilon.$$

It follows that

$$\begin{aligned} &\mathsf{SD}(\mathbf{z}_1 \circ \mathsf{Ext}_1(\mathbf{x}, \mathbf{z}_1) \circ \mathbf{z}_2 \circ \mathsf{Ext}_2(f(\mathbf{x}), \mathbf{z}_2), \mathcal{U}_{\ell(n)}) \\ \le &\mathsf{SD}(\mathbf{z}_1 \circ \mathsf{Ext}_1(\mathbf{x}, \mathbf{z}_1) \circ \mathbf{z}_2 \circ \mathsf{Ext}_2(f(\mathbf{x}), \mathbf{z}_2), \mathcal{U}_{d+m_1} \circ \mathbf{z}_2 \circ \mathsf{Ext}_2(f(\mathbf{x}), \mathbf{z}_2)) \\ + &\mathsf{SD}(\mathcal{U}_{d+m_1} \circ \mathbf{z}_2 \circ \mathsf{Ext}_2(f(\mathbf{x}), \mathbf{z}_2), \mathcal{U}_{\ell(n)}) \\ \le &\epsilon + \epsilon = 2\epsilon. \end{aligned}$$

(Hiding) Let $\mathcal{A}$ be any adversary that violates the Hiding property. Suppose that

$$\Pr_{\mathbf{x} \leftarrow \mathcal{X}_n}[\mathcal{A}(\tilde{f}_n(\mathbf{x}, \mathcal{U}_{2d})) = \mathbf{x}] \ge \gamma.$$

We will use $\mathcal{A}$ to build an algorithm $\mathcal{A}'$ that inverts $f(\mathcal{X}_n)$ w.p. $O(\gamma/\epsilon^2)$.

Let $\mathbf{x} \leftarrow \mathcal{X}_n$ be a hidden string, and $\mathbf{y} = f_n(\mathbf{x})$ be the input of $\mathcal{A}'$. We sample $\mathbf{z}_1, \mathbf{z}_2 \leftarrow \mathcal{U}_d$. We also "guess" a string $\mathbf{z} \leftarrow \mathcal{U}_{m_1}$, with the hope that $\mathsf{Ext}_1(\mathbf{x}, \mathbf{z}_1) = \mathbf{z}$. Then we output $\mathcal{A}'(\mathbf{y}) := \mathcal{A}(\mathbf{z}_1 \circ \mathbf{z} \circ \mathbf{z}_2 \circ \mathsf{Ext}_2(\mathbf{y}, \mathbf{z}_2))$.

Conditioned on $\mathsf{Ext}_1(\mathbf{x}, \mathbf{z}_1) = \mathbf{z}$, the distribution of $\mathbf{z}_1 \circ \mathbf{z} \circ \mathbf{z}_2 \circ \mathsf{Ext}_2(\mathbf{y}, \mathbf{z}_2)$ is exactly $\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d})$. Therefore,

$$\Pr[\mathcal{A}'(\mathbf{y}) = \mathbf{x}] \ge \gamma \cdot \Pr[\mathsf{Ext}_1(x, \mathbf{z}_1) = \mathbf{z}] = \gamma \cdot 2^{-m_1}.$$

Note that besides $\mathbf{y} = f(\mathbf{x})$, $\mathcal{A}'$ does not know any information about $\mathbf{x}$. Therefore, for every $x' \in f^{-1}(\mathbf{y})$, the probability that $\mathcal{A}'(\mathbf{y}) = x'$ should also be at least $\gamma \cdot 2^{-m_1}$. We have

$$\begin{aligned} \Pr_{\mathbf{y} = f(\mathcal{X}_n)}[\mathcal{A}'(\mathbf{y}) \in f^{-1}(\mathbf{y})] &\ge \gamma \cdot 2^{-m_1} \cdot |f_{\mathcal{X}}^{-1}(\mathbf{y})| \\ &\ge \gamma \cdot 2^{r-1-m_1} = O(\gamma/\epsilon^2). \end{aligned}$$ ◀

A.4 Step III: Appending a Hardcore Function

Note that the output length of $\tilde{f}_n$ is $\tau := \log(n/\delta) + 4\log\frac{1}{\epsilon} + O(1)$ bits shorter than the input length of $\tilde{f}_n$. In this section, we append a hardcore function at the end of $\tilde{f}_n$, making it a pseudorandom generator with stretch $\alpha > 0$. In particular, we need:

- a hardcore function $\mathsf{HC} : \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^k$ with distinguishing probability ϵ, where $k := \tau + \alpha$, and $d' := d_{\mathsf{HC}}(n, k, \epsilon)$. Let R be the reconstruction algorithm of this hardcore function, and $L := L(n, k, \epsilon)$ be the list size.

Let $\tilde{n} := n + 2d + d'$. Recall that $G_{n,r} : \{0,1\}^{\tilde{n}} \to \{0,1\}^{\tilde{n}+\alpha}$ is defined as

$$G_{n,r}(x, z_1, z_2, z_3) := z_1 \circ \mathsf{Ext}_1(x, z_1) \circ z_2 \circ \mathsf{Ext}_2(f(x), z_2) \circ z_3 \circ \mathsf{HC}(x, z_3).$$

Let $\mathcal{E}_{\tilde{n}} := \mathcal{X}_n \times \{0,1\}^{2d+d'}$. In other words, a uniform random string from $\mathcal{E}_{\tilde{n}}$ can be sampled as $\mathbf{x} \circ \mathbf{z}$, where $\mathbf{x} \leftarrow \mathcal{X}_n$ and $\mathbf{z} \leftarrow \mathcal{U}_{2d+d'}$. We will show that $G_{n,r}$ is a condEP-PRG whose "condition" is $\mathcal{E}_{\tilde{n}}$. In particular, Lemma 88 shows that $G_{n,r}(\mathcal{E}_{\tilde{n}})$ is pseudorandom, and Lemma 89 shows that $G_{n,r}(\mathcal{E}_{\tilde{n}})$ is entropy-preserving.

▶ **Lemma 88.** *Every polynomial-size adversary $\mathcal{A}$ fails to 4ϵ-distinguish $G_{n,r}(\mathcal{E}_{\tilde{n}})$ from $\mathcal{U}_{\ell(n)+d'+k}$.*

More precisely, if there is an adversary $\mathcal{A}$ of size s that 4ϵ-distinguishes $G_{n,r}(\mathcal{E}_{\tilde{n}})$ from $\mathcal{U}_{\ell(n)+d'+k}$, then there is an adversary $\mathcal{A}'$ of size $s \cdot \mathrm{poly}(n/\epsilon)$ that on input $\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d})$, guesses $\mathcal{X}_n$ with success probability $\epsilon/2L$.

Proof. Suppose $\mathcal{A}$ is an adversary that 4ϵ-distinguishes $G_{n,r}(\mathcal{E}_{\tilde{n}})$ from $\mathcal{U}_{\ell(n)+d'+k}$.

Since $\mathsf{SD}(\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d}), \mathcal{U}_{\ell(n)}) \le 2\epsilon$, it must be the case that $\mathcal{A}$ could 2ϵ-distinguish $G_{n,r}(\mathcal{E}_{\tilde{n}})$ from $\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d}) \circ \mathcal{U}_{d'+k}$. Equivalently, let $\mathbf{x} \leftarrow \mathcal{X}_n$, then given the information of $\tilde{f}(\mathbf{x}, \mathcal{U}_{2d})$, $\mathcal{A}$ could 2ϵ-distinguish $\mathcal{U}_{d'} \circ \mathsf{HC}(\mathbf{x}, \mathcal{U}_{d'})$ from $\mathcal{U}_{d'+k}$. We say a string $w := (x, z_1, z_2)$ is *good* if $\mathcal{A}$ could ϵ-distinguish $\tilde{f}_n(w) \circ \mathcal{U}_{d'} \circ \mathsf{HC}(x, \mathcal{U}_{d'})$ from $\tilde{f}_n(w) \circ \mathcal{U}_{d'+k}$. Then by a Markov bound, a random $\mathbf{w} \leftarrow \mathcal{X}_n \circ \mathcal{U}_{2d}$ is good w.p. at least ϵ.

The adversary $\mathcal{A}'$ that violates the (Hiding) property of $\tilde{f}_n$ simply follows from the reconstruction algorithm R. In particular, on input $\tilde{f}_n(w) = \tilde{f}_n(x, z_1, z_2)$, $\mathcal{A}'$ constructs the following oracle:

$$\mathcal{O}(r) := \mathcal{A}(\tilde{f}_n(w) \circ r),$$

runs the algorithm $R^{\mathcal{O}}$ to obtain a list of size L, and outputs a random element in the list.

We analyze $\mathcal{A}'$. Suppose $\mathcal{A}'$ is given $\tilde{f}_n(w)$ for a good w, then $\mathcal{O}$ indeed ϵ-distinguishes $\mathcal{U}_{d'} \circ \mathsf{HC}(x, \mathcal{U}_{d'})$ from $\mathcal{U}_{d'+k}$. Therefore, w.p. $\ge 1/2$, x is in the list outputted by $R^{\mathcal{O}}$. If this is the case, we will correctly output x w.p. $\ge 1/L$. It follows that on input $\tilde{f}_n(\mathbf{x}, \mathcal{U}_{2d})$ where $\mathbf{x} \leftarrow \mathcal{X}_n$, $\mathcal{A}'$ outputs $\mathbf{x}$ w.p. $\ge \epsilon/2L$. Finally, as R is a polynomial-size oracle circuit (actually a PPT oracle machine), the size of $\mathcal{A}'$ is $s(n) \cdot \mathrm{poly}(n/\epsilon)$. ◀

▶ **Lemma 89.** *Suppose that $\epsilon < \frac{1}{10n^2}$. Then $\mathrm{H}(G_{n,r}(\mathcal{E}_{\tilde{n}})) \ge \tilde{n} - \tau - 2$.*

Proof. Since $\mathsf{SD}(\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d}), \mathcal{U}_{\ell(n)}) \le 2\epsilon < \frac{1}{\ell(n)^2}$, by [62, Lemma 2.2], we have $\mathrm{H}(\tilde{f}_n(\mathcal{X}_n, \mathcal{U}_{2d})) \ge \ell(n) - 2$. It follows that $\mathrm{H}(G_{n,r}(\mathcal{E}_{\tilde{n}})) \ge (\ell(n) - 2) + d' \ge \tilde{n} - \tau - 2$. ◀

A.5 Putting It Together

Proof of Theorem 57. Suppose that for every $\mathcal{X} = \{\mathcal{X}_n\}$ that satisfies the premise of Claim 86, and $\mathcal{E}_{\tilde{n}}$ defined above, there is an adversary of size $s(n)$ that 4ϵ-distinguishes $G_{n,r}(\mathcal{E}_{\tilde{n}})$ from the uniform distribution. Then:

- By Lemma 88, there is an adversary of size $s(n) \cdot \mathrm{poly}(n/\epsilon)$ that on input $\tilde{f}(\mathcal{X}_n, \mathcal{U}_{d_1+d_2})$, guesses $\mathcal{X}_n$ w.p. $\ge \epsilon/2L$.
- By Lemma 87, there is an adversary of size $s(n) \cdot \mathrm{poly}(n/\epsilon)$ that inverts $f_n(\mathcal{X}_n)$ w.p. $\ge \frac{1}{2\epsilon L}$.

It follows from Claim 86 that there is an adversary of size $s \cdot \mathrm{poly}(nL/\epsilon)$ that inverts f on a $1 - \delta$ fraction of inputs. ◀

B Proof of Theorem 62

In this section, we briefly review the universal hash functions in [56] that are computable by linear-size circuits, with an emphasis on the uniformity of these circuits. Throughout this section, a circuit family is uniform if it satisfies Definition 60. An XOR-circuit is a (multi-output) circuit that only uses XOR gates of fan-in 2. To match Definition 60, we also require that every gate in an XOR-circuit has fan-out at most 2.

For convenience, we denote $[n] = \{0, 1, \ldots, n-1\}$, and $(n) = \{0, 1\}^n$.

Outline. Our start point is the strongly explicit family of expanders by [64, 29]. Spielman [78] showed that these expanders imply asymptotically optimal error-correcting codes (i.e., with constant rate and constant relative distance). Using an expander walk trick, for any constant $\epsilon > 0$, one could construct error-correcting codes with relative distance $1 - \epsilon$, constant rate, and constant alphabet size. By the construction of [56], such codes imply universal hash functions.

B.1 Strongly Explicit Expanders

We use the following construction due to [64, 29]. (See also [45, Construction 8.1].) For every integer n, we have a graph $\mathcal{G}_n$ with n^2 vertices such that every vertex has degree 8. The vertex set of $\mathcal{G}_n$ is $\mathbb{Z}_n \times \mathbb{Z}_n$. Each vertex $v = (x, y)$ is adjacent to the following vertices

$$\gamma_1(v) = (x + 2y, y), \gamma_2(v) = (x + 2y + 1, y), \gamma_3(v) = (x, y + 2x), \gamma_4(v) = (x, y + 2x + 1).$$

Here the additions are modulo n. Note that $\gamma_1, \dots, \gamma_4$ are bijections, and the other four neighbors of v are simply $\gamma_1^{-1}(v), \dots, \gamma_4^{-1}(v)$. The graph might contain self-loops or parallel edges.

▶ **Theorem 90** ([29]). *For every integer $n \geq 1$, the second largest eigenvalue of the adjacency matrix of $\mathcal{G}_n$ is at most $5\sqrt{2} < 8$.*

In our construction, we need the degree of the expanders to be a large enough constant. We can simply pick a large enough constant k and take the k-th power of $\mathcal{G}_n$. Let $\mathcal{G}_n^k$ be the k-th power of $\mathcal{G}_n$, i.e., for every $u, v \in V(\mathcal{G}_n)$, the number of (parallel) edges between u and v in $\mathcal{G}_n^k$ is equal to the number of length-k paths between u and v in $\mathcal{G}_n$. Then the degree of $\mathcal{G}_n^k$ is $d := 8^k$, and the second largest eigenvalue of the adjacency matrix of $\mathcal{G}_n^k$ is at most $(5\sqrt{2})^k < d$.

▶ Remark 91. It will be convenient to define an explicit mapping (bijection) between $E(\mathcal{G}_n^k)$ and $[dn^2/2]$. Note that each edge $(u, v) \in \mathcal{G}_n^k$ can be represented by a start vertex u and a string $\sigma_1\sigma_2 \dots \sigma_k$ where each $\sigma_i \in \Sigma := \{\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_1^{-1}, \gamma_2^{-1}, \gamma_3^{-1}, \gamma_4^{-1}\}$. The meaning of this representation is that $(\sigma_1 \circ \sigma_2 \circ \cdots \circ \sigma_k)(u) = v$. Each edge has two representations: $(u, \sigma_1\sigma_2 \dots \sigma_k)$ or $(v, \sigma_k^{-1}\sigma_{k-1}^{-1} \dots \sigma_1^{-1})$. We arbitrarily choose a size-$(d/2)$ subset S of Σ^k such that for each $\sigma_1, \sigma_2, \dots, \sigma_k \in \Sigma$, exactly one of $\sigma_1\sigma_2 \dots \sigma_k$ and $\sigma_k^{-1}\sigma_{k-1}^{-1} \dots \sigma_1^{-1}$ is in S. We fix and hardcode a bijection between $[d/2]$ and S. Given an integer $i \in [dn^2/2]$, we interpret i as a pair of $v \in V(\mathcal{G}_n^k)$ and $\sigma_1\sigma_2 \dots \sigma_k \in S$, and the edge corresponding to i is represented as $(v, \sigma_1\sigma_2 \dots \sigma_k)$. This bijection and its inverse are computable in time $O(\log n)$.

B.2 Error-Reduction Codes

An intermediate step in [78] is to construct *error-reduction codes*, which are weaker primitives compared to error-correcting codes.

Let $r, \delta, \epsilon > 0$ be constants. Recall that we defined $(n) = \{0, 1\}^n$ for convenience. An error-reduction code of rate r, error reduction ϵ and reducible distance δ is a function $\mathcal{C} : (rn) \to ((1 - r)n)$ mapping rn “message” bits into $(1 - r)n$ “check” bits, such that the following holds. The codeword of a message x is $x \circ \mathcal{C}(x)$. For any message x, if we are given a corrupted codeword that differs from $x \circ \mathcal{C}(x)$ with at most $v \leq \delta n$ message bits and at most $t \leq \delta n$ check bits, then we can recover a codeword that differs from $x \circ \mathcal{C}(x)$ in at most ϵt bits. (We will not be particularly interested in the complexity of recovery or decoding algorithms.)

▶ **Lemma 92.** *For some absolute constants $\epsilon < 1$ and $\delta > 0$, there is a family of error-reduction codes $\mathcal{R} = \{\mathcal{R}_n : (n) \to (\lfloor n/2 \rfloor)\}$ with error-reduction ϵ and reducible distance δ. Moreover, the sequence of functions $\{\mathcal{R}_n\}$ can be computed by a uniform family of linear-size* XOR-*circuits.*

Proof Sketch. First, let m be the smallest integer such that $dm^2/2 \ge n$. Note that m can be computed in $O(\log n)$ time. Let $r = 9/10$, then for large enough n, $dm^2(1-r)/r \le n/2$. It suffices to construct an error-reduction code with $dm^2/2$ message bits and $dm^2(1-r)/r$ check bits.

We use [78, Definition 16], where B is the edge-vertex incidence graph of $\mathcal{G}_m^k$, and $\mathcal{S}$ is a good (linear) error-correcting code on d-bit messages that has rate r. (Since d is a constant, we can hardcode $\mathcal{S}$ in our algorithm. on the other hand, since d is large enough, $\mathcal{S}$ exists.)

An equivalent formulation is as follows. We assign a message bit to every edge of $\mathcal{G}_m^k$. For each vertex $v \in V(\mathcal{G}_m^k)$, let $b_1 b_2 \dots b_d$ be the bits on the d incident edges of v. This vertex outputs $d(1-r)/r$ check bits which are the check bits of $\mathcal{S}$ on message $b_1 b_2 \dots b_d$. Concatenating the outputs of each vertex, we obtain an error-reduction code of $dm^2/2$ message bits and $dm^2(1-r)/r$ check bits. By [78, Lemma 18], for some absolute constants $\epsilon < 1$ and $\delta > 0$, this error-reduction code has error-reduction ϵ and reducible distance δ.

Computing the i-th gate of the encoding circuit reduces to computing the indices of the incident edges of a vertex $v \in V(\mathcal{G}_m^k)$. By Remark 91, this is computable in $O(\log n)$ time. ◀

We actually need error-reduction codes with error-reduction $\epsilon = 1/2$. We can simply iterate the code in Lemma 92 for $O(1)$ times. The encoding circuit is still uniform. Therefore, we have:

▶ **Corollary 93.** *Lemma 92 holds for $\epsilon = 1/2$.*

B.3 Error-Correcting Codes

The construction in [78, Section II] transforms an error-reduction code into an error-correcting code. Here we only review its encoding algorithm and check that they can be implemented by uniform XOR circuits. The correctness of this error-correcting code is proved in [78].

▶ **Lemma 94.** *There is a constant $n_0 > 1$ and a family $\mathcal{C} = \left\{\mathcal{C}_k : (n_0 2^{k-2}) \to (n_0 2^k)\right\}$ of error-correcting codes with constant relative distance. Moreover, $\mathcal{C}$ can be encoded by a uniform family of linear-size* XOR *circuits.*

Proof Sketch. We recursively define $\mathcal{C}_k$ as follows. First, $\mathcal{C}_0 : (n_0/4) \to (n_0)$ is any good enough error-correcting code. Since n_0 is a constant, our algorithm can hardcode $\mathcal{C}_0$.

Now, let $k \ge 1$, we define $\mathcal{C}_k$ as follows. Let $x \in (n_0 2^{k-2})$ be the inputs of $\mathcal{C}_k$.

- The first $n_0 2^{k-2}$ outputs of $\mathcal{C}_k$ will always be x itself.[19] Note that we require the fan-out of gates to be at most 2, therefore we need to make a copy of x. Similarly, we may need to copy the A_k, B_k, C_k defined below. The circuit size is still linear in 2^k.
- We pick an error-reduction code $\mathcal{R}_{k-2} : (n_0 2^{k-2}) \to (n_0 2^{k-3})$, and output $A_k := \mathcal{R}_{k-2}(x)$.
- Let $\mathcal{C}_{k-1} : (n_0 2^{k-3}) \to (n_0 2^{k-1})$ be the error-correcting code we recursively defined. Let $A_k \circ B_k := \mathcal{C}_{k-1}(A_k)$, and we output B_k. (Recall that the first $n_0 2^{k-3}$ outputs of $\mathcal{C}_{k-1}$ is equal to its inputs, i.e., A_k.)
- We pick an error-reduction code $\mathcal{R}_{k-1} : (n_0 2^{k-1}) \to (n_0 2^{k-2})$, and output $C_k := \mathcal{R}_{k-1}(A_k \circ B_k)$.

[19] We can assume this is also true for $\mathcal{C}_0$.

The required error-reduction codes are constructed in Corollary 93. The total number of output bits of $\mathcal{C}_k$ is $|x| + |A_k| + |B_k| + |C_k|$ which is indeed $n_0 2^k$.

The i-th gate of the encoding circuit of $\mathcal{C}_{k-1}$ can be computed as follows. Let $c2^k$ be the circuit complexity of the first, second, and fourth bullet. (That is, circuit complexity of $\mathcal{C}_k$ not counting the recursive part for $\mathcal{C}_{k-1}$.) We may assume c is a power of 2. The encoding circuit for $\mathcal{C}_k$ has $|\mathcal{C}_0| + \sum_{i=1}^{k} c2^i = |\mathcal{C}_0| + c(2^{k+1} - 1)$ gates. Taking the (base-2) logarithm of $(i - |\mathcal{C}_0|)/c$, we can find the "level of recursion" that the i-th gate is constructed. Then the problem reduces to computing the encoding circuit of $\mathcal{R}_j$ for some integer j, which is computable in $O(\log n)$ time. ◀

For every constant $\epsilon > 0$, the construction of [56] needs a code with relative distance $1 - \epsilon$ and constant alphabet size. As in [7, 35], we can "amplify" the code in Lemma 94 by an expander:

▶ **Lemma 95.** *For every constant $\epsilon > 0$, there is a constant $D > 0$ and a family of error-correcting codes $\{\mathcal{C}'_k : (n_0 2^{k-2}) \to [2^D]^{O(2^k)}\}$ that has relative distance $1 - \epsilon$. Moreover, if we interpret $[2^D]$ as length-D strings, then $\mathcal{C}'_k$ can be encoded by a uniform family of linear-size* XOR *circuits.*

Proof Sketch. Recall that $\{\mathcal{G}_n\}$ is the expander family constructed in Theorem 90, and $\{\mathcal{C}_k : (n_0 2^{k-2}) \to (n_0 2^k)\}$ is the family of error-correcting codes constructed in Lemma 94. Let m be the smallest integer such that $m^2 \geq n_0 2^k$. We pad zeros to the outputs of $\mathcal{C}_k$, thus $\mathcal{C}_k$ can be regarded as a code that outputs m^2 bits. We assign an output bit of $\mathcal{C}_k$ to each vertex in $\mathcal{G}_m$. The relative distance of $\mathcal{C}_k$ is still lower bounded by an absolute constant $\delta > 0$.

We will pick a large enough constant p, such that $\mathcal{G}_m^p$ has good expansion property: Every subset of $V(\mathcal{G}_m^p)$ with size at least $\delta \cdot m^2$ has at least $(1 - \epsilon)m^2$ neighbors. (See e.g. [7, Corollary 1].) Let $D := 8^p$, so every vertex in $\mathcal{G}_m^p$ has degree D. On input $x \in (n_0 2^{k-2})$, recall that we assigned each vertex in $V(\mathcal{G}_m^p)$ a bit of the codeword $\mathcal{C}_k(x)$. For every $v \in V(\mathcal{G}_m^p)$, the vertex v will output the concatenation of the bits assigned to its neighbor, which can be interpreted as an element in $[2^D]$. The code $\mathcal{C}'_k(x)$ simply concatenates the outputs of each vertex $v \in V(\mathcal{G}_m^p)$ together.

Consider the encoding circuits of $\mathcal{C}'_k$. As we need each gate to have fanout at most 2, we make D copies of the encoding circuit of $\mathcal{C}_k$. For every $\sigma = \sigma_1\sigma_2 \ldots \sigma_k \in \Sigma^k$, we have a copy of $\mathcal{C}_k$ denoted as $\mathcal{C}_k^\sigma$. For each vertex v, and each $\sigma \in \Sigma^k$, let u be the σ-th neighbor of v. The σ-th bit of the output of v is the u-th output of the circuit $\mathcal{C}_k^\sigma$. We can see this encoding circuit is uniform. ◀

▶ Remark 96. Lijie Chen (personal communication) suggested a similar approach based on expander random walks [80, Proposition 6.6]. As the p-th power of an expander graph G consists of length-p walks in G, the two approaches are essentially the same.

B.4 Universal Hash Functions

Finally, we are ready to verify that the universal hash functions in [56] are uniform.

▶ **Theorem 62.** *For every integer n, m where $m = O(n)$, there exists an integer $k = O(n)$, and a family of universal hash functions $\{h_{n,m} : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^m\}$, such that $h_{n,m}$ can be computed by a uniform family of linear-size circuits that are skew w.r.t. the second argument.*

Proof Sketch. Let $n_1 := cn$ for a large enough constant c, ϵ be a small enough constant, and D be the constant in Lemma 95 depending on ϵ. We need three ingredients:

- An *ℓ-exposure resilient function* (ERF) $\mathsf{ERF} : \{0,1\}^{n_1 D} \to \{0,1\}^m$ [21]. It is shown in [24] that for any (linear) error-correcting code $\mathcal{C} : \{0,1\}^m \to \{0,1\}^n$ with generator matrix G and minimum distance d, the transpose matrix G^{T} mapping n input bits to m output bits is a perfect ℓ-ERF where $\ell := n - d + 1$.

  For an XOR-circuit C that computes the linear transform G over GF(2), we can obtain a circuit computing the linear transform G^{T}, by exchanging the input gates and output gates and reversing the directions of every wire [18, 56]. In particular, every gate $g \in C$ whose output feeds to the gates $g_1, g_2, \dots, g_k$ becomes, in the new circuit, an XOR gate g whose *inputs* are $g_1, g_2, \dots, g_k$.

  Therefore, Lemma 94 shows that an ℓ-ERF $\mathsf{ERF} : \{0,1\}^{n_1 D} \to \{0,1\}^m$ is computable by a uniform family of linear-size XOR circuits.
- An error-correcting code $\mathcal{C}'_k : \{0,1\}^n \to [2^D]^{n_1}$ with relative distance $1-\epsilon$, as in Lemma 95.
- A hash family $H : \{0,1\}^D \times \{0,1\}^{2D-1} \to \{0,1\}^D$, computable by a skew circuit w.r.t. the second argument. As D is a constant, we can simply hardcode this hash family. (See e.g. Section 5.2.1 for an instantiation based on Toeplitz matrices.)

The construction of [56] goes as follows. On input $x \in \{0,1\}^n$, we first compute $\mathcal{C}'(x) \in [2^{h+1}]^{n_1}$. Next, we receive n_1 keys $k_1, k_2, \dots, k_{n_1} \in \{0,1\}^{2h+1}$ which are the keys for our hash function. Let $t \in [2^{h+1}]^{n_1}$ be the following message: $t_i := H(\mathcal{C}'(x)_i, k_i)$. We treat t as a string of length $m(h+1)$, and the output of our hash function is $\mathsf{ERF}(t)$.

It is easy to see that this family is uniform. ◀

On the Pseudo-Deterministic Query Complexity of NP Search Problems

Shafi Goldwasser
University of California, Berkeley, CA, USA

Russell Impagliazzo
University of California, San Diego, CA, USA

Toniann Pitassi
University of Toronto, Canada
Columbia University, New York, NY, USA
Institute of Advanced Study, Princeton, NJ, USA

Rahul Santhanam
University of Oxford, UK

Abstract

We study *pseudo-deterministic* query complexity – randomized query algorithms that are required to output the *same* answer with high probability on all inputs. We prove $\Omega(\sqrt{n})$ lower bounds on the pseudo-deterministic complexity of a large family of search problems based on unsatisfiable random CNF instances, and also for the promise problem (FIND1) of finding a 1 in a vector populated with at least half one's. This gives an exponential separation between randomized query complexity and pseudo-deterministic complexity, which is tight in the quantum setting. As applications we partially solve a related combinatorial coloring problem, and we separate random tree-like Resolution from its pseudo-deterministic version. In contrast to our lower bound, we show, surprisingly, that in the zero-error, average case setting, the three notions (deterministic, randomized, pseudo-deterministic) collapse.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Oracles and decision trees; Theory of computation → Proof complexity

Keywords and phrases Pseudo-determinism, Query complexity, Proof complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.36

Funding *Shafi Goldwasser*: Research supported in part by DARPA under Contract No. HR001120C0015.
Russell Impagliazzo: Research supported by NSF and the Simons Foundation.
Toniann Pitassi: Research supported by NSERC, the IAS School of Mathematics and NSF Grant No. CCF-1900460.

Acknowledgements We thank Ofer Grossman, Ran Raz, Avi Wigderson and Ryan Williams for helpful discussions. The quantum query upper bound for FIND1 was pointed out to the fourth author by Igor Oliveira. We also thank the anonymous CCC reviewers for very helpful comments.

1 Introduction

The natural and beautiful notion of *pseudo-determinism* which formalizes random search algorithms that are required on every input, to output the *same* solution with high probability, was introduced by Gat and Goldwasser in [12]. A motivating example is the problem of finding an n-bit prime number in time polynomial in n. Since primality testing is in P, and the primes are dense within the natural numbers, we can efficiently find a prime with high probability by repeatedly selecting a random number, test it for primality, and halt if a prime is found. In contrast the fastest *deterministic* algorithm for finding primes is exponential in n. A pseudo-deterministic algorithm lies between a randomized search algorithm (which on each input may output a large number of different solutions as we vary the random coins), and a deterministic algorithm. Here we are allowed unlimited use of randomness, but the search algorithm is required to output a *canonical* answer $f(x)$ on each input x (with very high probability).

Pseudodeterminism is important, both because of the intrinsic nature of the underlying questions that it raises, and because of its strong connections to other phenomena. First, it relates to the *reproducibility* question in science – empirical research has unavoidable randomness in many phases of research, from data generation/collection, to experiment design and testing. Pseudodeterministic algorithms correspond to *reproducible* experiments where the same (or a very similar) outcome will usually be obtained if the experiment is reproduced under a different set of (random) conditions [12, 22]. Pseudodeterminism also is related to the notion of *global stability* in machine learning, which is closely tied to generalization in machine learning

Starting with [12], a growing body of research has laid much of the groundwork for a theory of pseudo-deterministic complexity theory, establishing the power and limitations of pseudo-determinism for a variety of computational models (See for example [12, 13, 22, 14, 15, 16].) Assuming $\mathsf{P} = \mathsf{BPP}$, polynomial-time pseudo-deterministic search is equivalent to deterministic polynomial-time search. This implies for example that finding an n-bit prime is in polytime assuming $\mathsf{P} = \mathsf{BPP}$, but this is far from giving a efficient deterministic or pseudo-deterministic algorithm that generates primes. Oliveira and Santhanam [30] demonstrated the power of pseudo-determinism by proving unconditionally that finding primes could be carried out (for infinitely many n) in subexponential-time.

1.1 Our Results

In this paper we study the power of pseudo-determinism in the context of *query complexity*, which was first defined and studied by Goldreich, Goldwasser and Ron [13]. We focus on search problems with solutions that can be verified easily by deterministic query algorithms, similarly to the complexity class FNP, and that have an abundance of solutions[1]. In other words, we consider search problems where a solution can be found randomly simply by guessing and then verifying the guess, but for which deterministically finding a solution is difficult. The most natural problems we consider are promise problems, but we prove lower bounds for these via reduction to problems which have the above property on the full domain, i.e., we prove lower bounds for the analogs of TFNP problems with an abundance of witnesses.

This scenario is of central importance in complexity theory, where many longstanding open problems are closely connected to explicit constructions of objects that exist in abundance. For example, explicit constructions of rigid matrices imply circuit lower bounds, and explicit constructions of functions that are hard to compute (or approximate) imply derandomization.

1. We define an elementary promise search problem, FIND1: given an n bit string with the promise that it contains at least $n/2$ 1's, output a coordinate i such that $x_i = 1$. FIND1 is easy for randomized query complexity, and we observe (Section 3) that FIND1 is complete for easily verifiable search problems with randomized query algorithms. [2]

[1] In contrast, the linear query lower bounds of [13] are not for a problem with easily verifiable solutions.

[2] A similar problem titled Find-Support-Elem was considered in the context of studying the space complexity of pseudo-deterministic streaming algorithms [17]

2. We prove (Section 4) a lower bound of $\Omega(\sqrt{n})$ on the pseudo-determinsitic query complexity of a broad class of search problems associated with random unsatisfiable CNF formulas, a problem in the query analog of TFNP. As a corollary we prove the same lower bound for FIND1, thus separating randomized from pseudo-deterministic query complexity for a problem in the analog of FNP. Our lower bound also holds in the quantum setting where a simple binary search plus Grover's result shows that our lower bound is tight. A key idea in our proof is to look at a different *structured* family of search problems associated with highly unsatisfiable CNF formulas. Our lower bound for these structured search problems follows by combining Huang's Sensitivity Theorem with known linear lower bounds on the Nullstellensatz/SOS degree for refuting random unsatisfiable CNF instances.
3. Applications. We study two questions related to our lower bound in Section 5. First as a corollary, we obtain a lower bound for a related combinatorial coloring problem that we define and find independently interesting. Secondly, we extend our results to give an exponential separation between the *size* of randomized decision trees and the *size* of pseudo-deterministic decision trees. Our size separation in turn implies an exponential separation between *pseudo-deterministic* tree-like Resolution refutations and *random* tree-like Resolution refutations (defined in [7]).
4. In contrast to our lower bounds which expose the limitations of pseudo-deterministic query algorithms, we prove (Section 6) that in the zero-error average setting, the three notions (deterministic, randomized, and pseudo-deterministic) collapse.

1.2 Our Ideas

We discuss our results and the ideas behind them at a high level.

Our observation that FIND1 is a canonical problem for pseudo-deterministic query complexity for problems in FNP follows from the fact that every randomized query algorithm can be assumed to have as support a linear-size set B of deterministic decision trees. Assume that FIND1 has an efficient pseudodeterministic query algorithm, and let $\mathcal{S}$ be a problem in FNP. We define a pseudodeterministic algorithm for $\mathcal{S}$ by simulating the protocol for FIND1. Every time the protocol for FIND1 queries a bit, we run the corresponding decision tree in the linear-size set B and return 1 iff the decision tree returns a valid solution to $\mathcal{S}$. Note that since $\mathcal{S}$ is in FNP, we can check that a solution is valid efficiently. When the protocol for FIND1 concludes and outputs an index j of a bit, we simulate the corresponding decision tree in B and return the solution for $\mathcal{S}$ that it outputs.

Our lower bound for a TFNP problem is for the search problem associated with a randomly chosen k-CNF formula ϕ of linear size. The main property we require from this formula is that the factor graph is a strong enough expander. The search problem associated with ϕ is to return the index of an unsatisfied clause, given an assignment to the variables. We choose the size of the CNF large enough so that for each assignment to variables, a constant fraction of clauses are violated. Thus there is a trivial randomized protocol for the search problem with cost $O(1)$: output a random clause.

We show that any pseudo-deterministic query algorithm for this problem requires $\Omega(\sqrt{n})$ queries, using a novel connection to proof complexity. We use the known result [21, 6, 3] that the random CNFs we consider require linear degree to refute in the Nullstellensatz proof system to show a lower bound on the Fourier degree of the search problem associated with these CNFs. We then use the recent breakthrough of Huang on the Sensitivity Conjecture [25] to lower bound the sensitivity by $\Omega(\sqrt{n})$, and show that the pseudo-deterministic query complexity is lower bounded by the sensitivity. By using the very recent result of [2] instead of [25], we can even lower bound the pseudodeterministic quantum query complexity by $\Omega(\sqrt{n})$. For quantum query complexity, this is actually tight, as a matching upper bound follows from combining binary search with Grover's algorithm.

Our quest for an improved linear lower bound for FIND1 raises an interesting combinatorial question: given any coloring of the hypercube (omitting the all zeroes vertex) with n colors such that each vertex is colored with the index of one of its 1s, must there be vertex with a constant fraction of 1s so that a constant fraction of its neighbours are colored differently from it? If the answer to this question is yes, we would be able to show that FIND1 requires linear pseudo-deterministic query complexity. The question above is about the sensitivity of a coloring; we can ask an analogous question for block-sensitivity and in this case, it turns out that we can prove an $\Omega(\sqrt{n})$ lower bound, which also implies our $\Omega(\sqrt{n})$ lower bound for FIND1.

Our proof of a pseudo-deterministic query lower bound uses ideas from proof complexity. We show that there is a connection in the reverse direction too, by defining pseudo-deterministic versions of propositional proof systems such as Resolution. A broad question in proof complexity is whether we can use proof systems to capture the behaviour of randomized algorithms. Motivated in part by this and in part by a question about bounded-depth Frege proof systems, [7] defined Random Resolution: a randomized version of Resolution. This is quite a powerful system which even refutes random k-CNFs in constant size, contrary to our intuition that random k-CNFs should be hard to solve. We define pseudo-deterministic Resolution and pseudo-deterministic Tree Resoluton, and we show that pseudo-deterministic Tree Resolution is efficiently verifiable, suggesting that it is a more viable candidate for capturing the behaviour of randomized algorithms. We apply the ideas of our separation between randomized query complexity and pseudo-deterministic query complexity to get a strong separation between Random Tree Resolution and pseudo-deterministic Tree Resolution: random k-CNFs can be refuted in linear size in Random Tree Resolution but require $2^{\Omega(\sqrt{n})}$ size in pseudo-deterministic Tree Resolution.

Finally, we turn our attention from lower bounds to algorithms. We show that perhaps surprisingly, there is a close connection between randomized query complexity and pseudo-deterministic query complexity on average. Specifically, for zero-error algorithms (where the query algorithm is not allowed to make a mistake), we show that over any distribution D, the randomized, pseudo-deterministic and deterministic query complexity are all within a polylogarithmic factor of each other. Similarly, we show that for any approximation problem (such as the problem of approximating the Hamming weight of an input considered in [13], for which there is a constant-query randomized algorithm) and distribution D, there is an efficient bounded-error pseudo-deterministic query algorithm which asks few queries on average over D. Note that we require the algorithm to be pseudo-deterministic on *every input*, which is a pretty strong guarantee.

As a toy problem for our result on zero-error query algorithms, consider the FIND1 problem, which does have a very efficient zero-error randomized algorithm. Given any distribution D, we can use an averaging argument to identify a small set of decision trees from the support of our randomized query algorithm such that at least one of the trees from this set outputs a correct solution with probability at least $1 - 1/n$ over the distribution. We can also efficiently check if this is indeed the case. If not, we simply query every bit, and this doesn't cost too much on average because this case happens with very low probability.

Generalizing to efficient average-case zero-error algorithms is somewhat more involved, and requires an interleaving simulation of decision trees together with a Markov argument at different scales. We use similar ideas for our bounded-error pseudo-deterministic algorithms - the challenge is to meet the pseudo-determinsitic guarantee on every input.

1.3 Related Work

Optimal query separations were already proven by [13] but their search problem is not in FNP – that is, for the problem that they studied, solutions are not verifiable with a polylogarithmic number of queries. In particular, they studied the search problem of estimating the number of ones in a binary string to within an additive ϵn. They proved that this search problem has low randomized query complexity, but requires linear pseuododeterministic query complexity.

2 Definitions

▶ **Definition 1.** *A search problem over domain $\mathcal{X}$ and range $\mathcal{O}$ is defined to be a relation $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{O}$. For $x \in \mathcal{X}$, the feasible solutions for $\mathcal{S}$ on x are the elements $o \in \mathcal{O}$ such that $(x, o) \in \mathcal{S}$. $\mathcal{S}$ is total if there is at least one feasible solution for every $x \in \mathcal{X}$. A function $f : \mathcal{X} \to \mathcal{O}$ solves the search problem $\mathcal{S}$ if for every $x \in \mathcal{X}$ with at least one feasible solution for $\mathcal{S}$, $(x, f(x)) \in \mathcal{O}$.*

Deterministic Query Complexity. Let $\mathcal{X} = \{0,1\}^n$. A determininistic decision tree T over $x_1, \dots, x_n$ with outputs from $\mathcal{O}$ is a binary tree where each internal node is labelled with a variable x_i, and with outedges labelled by $x_i = 0$ and $x_i = 1$. Each leaf of the tree is labelled with some $o \in \mathcal{O}$. A deterministic decision tree T computes $f : \{0,1\}^n \to \mathcal{O}$ if for every input $x \in \{0,1\}^n$, the (unique) path in T consistent with x has leaf label $f(x)$. Let $\mathsf{P}^{\mathsf{dt}}(f)$ be the minimum depth of a deterministic decision tree computing f. [3] For a search problem $\mathcal{S} \subseteq \{0,1\}^n \times \mathcal{O}$, The (deterministic) query complexity of $\mathcal{S}$, $\mathsf{P}^{\mathsf{dt}}(\mathcal{S})$ is the minimum of $\mathsf{P}^{\mathsf{dt}}(f)$ over all functions f solving $\mathcal{S}$.

Randomized and Quantum Query Complexity. A randomized decision tree over $x_1, \dots, x_n$ with outputs from $\mathcal{O}$ is a distribution $\mathcal{T}$ over deterministic decision trees. A randomized decision tree $\mathcal{T}$ computes $f : \{0,1\}^n \to \mathcal{O}$ with error at most ϵ if for every input x, the probability (over T drawn from $\mathcal{T}$) that $T(x)$ ouputs $f(x)$ is at least $1 - \epsilon$. The bounded-error randomized query complexity of search problem $\mathcal{S}$, denoted by $\mathsf{BPP}^{\mathsf{dt}}(\mathcal{S})$, is the minimum over all functions f computing $\mathcal{S}$ of the depth of a randomized decision tree computing f with error $1/3$.

We can also define zero-error randomized query complexity for f and $\mathcal{S}$. In this case $\mathcal{T}$ is a distribution over decision trees, but with the property that for every x, the probability that $\mathcal{T}(x) = f(x)$ is one. Whereas before the depth was defined to be the maximum depth over all decision trees in the distribution, in the zero-error case, we define the depth to be the expected depth. The quantum query complexity for functions and search problems is defined analogously. (e.g., see [9].)

Nondeterministic Query Complexity. Let $\mathcal{S} \subseteq \{0,1\}^n \times [m]$ be a search problem. A *verification* decision tree for f is a decision tree $\mathcal{T}$ over the Boolean variables $x_1, \dots, x_n$, $y_1, \dots, y_{\log m}$ with outputs $\{0,1\}$ such that for every input pair $(x, y) \in \{0,1\}^n \times [m]$, $\mathcal{T}(x, y) = 1$ if and only if $(x, y) \in \mathcal{S}$. The verification query complexity of $\mathcal{S}$ is the minimum depth over all verification decision trees for $\mathcal{S}$. A search problem $\mathcal{S} \subseteq \{0,1\}^n \times [m]$ with $m = O(n)$ is an NP-search problem if there is a verification decision tree for $\mathcal{S}$ of depth polynomial in $\log m$.

[3] We note that since f may not be Boolean, $\mathsf{FP}^{\mathsf{dt}}(f)$ is a more accurate notation, but we slightly abuse notation and use P^{dt} to be consistent with prior work/notation.

Pseudodeterministic Query Complexity. Finally we define the bounded-error and zero-error pseudo-deterministic query complexity for total search problems $\mathcal{S}$. A bounded-error pseudo-deterministic decision tree for S is a distribution over decision trees with the following property: For every input x, there is a *canonical* value $o \in \mathcal{O}$ such that with probability at least $2/3$, $\mathcal{T}(x) = o$. In other words, $\mathcal{T}$ is a bounded-error randomized decision tree for a particular function f that solves $\mathcal{S}$. Let $\mathsf{psP}^{\mathsf{dt}}(\mathcal{S})$ denote the (bounded-error) pseudo-deterministic query complexity of $\mathcal{S}$. Similarly let $\mathsf{psQ}^{\mathsf{dt}}(\mathcal{S})$ denote the pseudo-deterministic bounded-error quantum query complexity of $\mathcal{S}$.

We note that for bounded-error randomized and pseudo-determinstic query algorithms, by repeatedly running the query algorithm $O(log(1/\delta))$ times, we can amplify the success probability from $2/3$ to $1 - \delta$.

Sensitivity and Block Sensitivity. Let $f : \{0,1\}^n \rightarrow \mathcal{O}$. A block $B \subseteq [n]$ is *sensitive* for f on input x if $f(x \oplus 1_B) \neq f(x)$, where 1_B is the n-bit string that is 1 on bits in B and 0 otherwise. In other words, if we change x by flipping all of the bits in B to get x^B, then the value of f changes (so $f(x) \neq f(x^B)$). The *block sensitivity* of x with respect to f, $\boldsymbol{bs}_x(f)$, is the maximal number of disjoint blocks that are all sensitive for x. We define $\boldsymbol{bs}(f) = max_{x \in \{0,1\}^n} \boldsymbol{bs}_x(f)$.

A bit $i \in [n]$ is sensitive for x with respect to f if the block $\{i\}$ is sensitive for x. The sensitivity of x with respect to f, $\mathbf{s}_x(f)$, is the maximal number of sensitive bits for x, and $\mathbf{s}(f) = max_{x \in \{0,1\}^n} \mathbf{s}_x(f)$.

Degree. A polynomial $q \in \mathbb{R}[x_1, \ldots, x_n]$ is said to *represent* the function $f : \{0,1\}^n \rightarrow \{0,1\}$ if $q(x) = f(x)$ for all $x \in \{0,1\}^n$. The (Fourier) degree of f, $\boldsymbol{d}(f)$ is the degree of the (unique) polynomial representing f. A multioutput function $f : \{0,1\}^n \rightarrow [m]$, induces a partition of $\{0,1\}^n$ into m classes, where the i^{th} class contains those inputs that are mapped to i (i.e., those x such that $f(x) = i$). Thus we can define m associated Boolean functions, f^i, $i \in [m]$, where $f^i(x)$ is 1 if and only if $f(x) = i$. The Fourier degree of $f : \{0,1\}^n \rightarrow [m]$ is defined as $max_{i \in [m]} \boldsymbol{d}(f^i)$, and the Fourier degree of a total search problem $\mathcal{S}$ is the minimum of $\boldsymbol{d}(f)$ over all functions f solving the search problem $\mathcal{S}$.

Known Relationships. Pioneering work of Nisan [28], Nisan and Szegedy [29] and Beals-et-al [4] studied the above query measures and showed that nearly all of them are polynomially equivalent. (See [5] for a nice exposition.) The two exceptions are pseudo-deterministic complexity (which was defined later) and sensitivity, which remained a longstanding open problem for thirty years. In recent breakthrough work, Huang [25] resolved the conjecture by proving $s(f) \geq deg(f)^{1/2}$. The exact quantitative relationships between the measures has been intensively studied; a table summarizing the state-of-the-art pairwise relationsihps (pre-Huang) is given in [1]. Post-Huang, [2] improved the relationships between deterministic query complexity, quantum query complexity and degree to near-optimal (ignoring polylog factors).

We summarize here the relationships that will be important for us. First, the following basic relationships are known:

$$\mathsf{Q}^{\mathsf{dt}}(f) = O(\mathsf{BPP}^{\mathsf{dt}}(f)) = O(\mathsf{P}^{\mathsf{dt}}(f))$$

$$\boldsymbol{d}(f) = O(\mathsf{P}^{\mathsf{dt}}(f))$$

$$\boldsymbol{s}(f) = O(\boldsymbol{bs}(f)) = O(\mathsf{BPP}^{\mathsf{dt}}(f)).$$

The following nontrivial relationships have recently been proven using Huang's theorem [2]:

$$\boldsymbol{d}(f) = O(\mathsf{Q}^{\mathsf{dt}}(f)^2)$$

$$\mathsf{P}^{\mathsf{dt}}(f) = O(\mathsf{Q}^{\mathsf{dt}}(f)^4).$$

These results are know to be tight within polylog factors. Before these results the best known (pre-Huang) was $\boldsymbol{d}(f), \mathsf{P}^{\mathsf{dt}}(f) = O(\mathsf{Q}^{\mathsf{dt}}(f)^6)$.

We now consider the relationship between the pseudo-deterministic, deterministic and randomized query classes. Let $\mathcal{S}$ be a FNP search problem, we have the easy inclusions:

$$\mathsf{P}^{\mathsf{dt}}(\mathcal{S}) \geq \mathsf{psBPP}^{\mathsf{dt}}(\mathcal{S}) \geq \mathsf{BPP}^{\mathsf{dt}}(\mathcal{S})$$

$$\mathsf{P}^{\mathsf{dt}}(\mathcal{S}) \geq \mathsf{psQ}^{\mathsf{dt}}(\mathcal{S}) \geq \mathsf{Q}^{\mathsf{dt}}(\mathcal{S}).$$

3 Search Problems in TFNP

We define $\mathsf{TFNP}^{\mathsf{dt}}$, the query analog of TFNP to be the class of all search problems $f : \{0,1\}^n \to [m]$ that admit a nondeterministic decision tree of complexity $\mathsf{polylog}(n)$. (Equivalently, f can be written as a $\mathsf{polylog}(n)$-width DNF.)

▶ **Definition 2.** *Let $X = \{x \in \{0,1\}^n, \mid |x| \geq n/2\}$ where $|x|$ is the number of 1's in x. The search problem FIND1 $\subseteq X \times [n]$ is defined by: $(x,i) \in$ FIND1 if and only if $x \in X$ and $x_i = 1$.*

It is not hard to see that the deterministic query complexity of FIND1 is $\Omega(n)$, but the randomized query complexity (and therefore also the quantum query complexity) is constant. Here we show that for any search problem in $\mathsf{TFNP}^{\mathsf{dt}}$ for which solutions are verifiable using few queries, a gap between randomized and pseudo-deterministic query complexity implies a gap between randomized and pseudo-deterministic query for FIND1.

Call a function $f : \mathbb{N} \to \mathbb{N}$ *reasonable* if $f(\Theta(n)) = \Theta(f(n))$. Note that functions such as $f(n) = n^\epsilon$ for $\epsilon < 1$, $f(n) = \log(n)$ and $f(n) = O(1)$, which often occur as bounds on query complexity, are all reasonable.

▶ **Theorem 3.** *Let $r, q, v : \mathbb{N} \to \mathbb{N}$ be reasonable functions. Let $\mathcal{S}$ be a search problem verifiable with $v(n)$ queries such that $\mathsf{BPP}^{\mathsf{dt}}(\mathcal{S}) \leq r(n)$ and $\mathsf{psP}^{\mathsf{dt}}(\mathcal{S}) \geq q(n)$. Then $\mathsf{psP}^{\mathsf{dt}}(\text{FIND1}) = \Omega(q(n)/(r(n)+v(n)))$.*

Proof. Since $\mathcal{S}$ has randomized decision tree complexity at most $r(n)$, there is a family $\mathcal{F}$ of deterministic decision trees of depth $r(n)$ such that for each $x \in \mathcal{I}$ of length n, a uniformly chosen tree from $\mathcal{F}$ solves $\mathcal{S}$ on x with probability at least 3/4. If we uniformly and independently pick a subfamily $\mathcal{F}'$ of cn trees from $\mathcal{F}$ for large enough constant c, it follows using Chernoff bounds and a union bound that with positive probability over the choice of $\mathcal{F}'$, for each $x \in \mathcal{X}$ of length n, a uniformly chosen tree from $\mathcal{F}'$ solves $\mathcal{S}$ on x with probability at least 2/3. Hence, by the probabilistic method, there must exist such a subfamily $\mathcal{F}'$. Fix such a subfamily, and let $T_1 \ldots T_m$ be an arbitrary enumeration of the decision trees in $\mathcal{F}'$, where $m = cn$.

Assume that FIND1 can be solved pseudo-deterministically with at most $p(m)$ queries on inputs of length m. We show how to solve $\mathcal{S}$ pseudo-determistically on inputs of length n with at most $p(m)(r(n)+v(n))$ queries. The pseudo-deterministic query algorithm A for $\mathcal{S}$ on input x of length n is as follows. We simulate the pseudo-deterministic query algorithm A' for FIND1 that makes at most $p(m)$ queries. If A' asks whether bit $i \in [m]$ is 1 in the input to FIND1, we run the query algorithm for $\mathcal{S}$ corresponding to tree T_i. By assumption, at most $r(n)$ queries are made, and some output y is produced. We verify that $(x, y) \in \mathcal{S}$ by using the $v(n)$ query verification algorithm for the search problem $\mathcal{S}$. If the verification succeeds, we assume the answer to the query made by A' is 1 and proceed, otherwise we proceed with the simulation of A assuming that the answer is 0. When we finish simulating A', some index $j \in [m]$ is output. We proceed to run the query algorithm corresponding to T_j on x and return the output z of this algorithm.

The cost of this query algorithm A is at most $p(m)(r(n) + v(n))$ since the simulation of each query of A' has cost at most $r(n) + v(n)$, and there are at most $p(m)$ queries along any computation path. It remains to argue that A pseudo-deterministically solves $\mathcal{S}$. By assumption, a uniformly chosen tree from $\mathcal{F}'$ solves $\mathcal{S}$ on x with probability at least 2/3 - this implies that for at least 2/3 fraction of indices $i \in [m]$, the simulation of a query made by A' to i returns 1. By assumption, A' pseudo-deterministically solves FIND1, hence there is a fixed $j \in [m]$ for which the query made by A' to j returns 1 such that A' outputs j with probability at least 2/3. But since T_j solves $\mathcal{S}$ correctly, this means that A outputs a fixed solution to the search problem $\mathcal{S}$ with probability at least 2/3.

Thus we have that $p(m) \geq q(n)/(r(n) + v(n))$ This implies that $p(m) = \Omega(q(m)/(r(m) + v(m))$ using $m = \Theta(n)$ and our assumption that the functions r, q, v are all reasonable. ◀

4 Lower Bounds for Pseudo-deterministic Query Complexity

▶ **Theorem 4.** *There is a $\sqrt{n}$ gap between the randomized and pseudo-deterministic query complexity of FIND1:*

(1) $\mathsf{BPP}^{\mathsf{dt}}(\mathit{FIND1}) = O(1)$*, and therefore* $\mathsf{Q}^{\mathsf{dt}}(\mathit{FIND1}) = O(1)$ *as well;*

(2) $\mathsf{psQ}^{\mathsf{dt}}(\mathit{FIND1}) = \Omega(\sqrt{n})$ *and thus* $\mathsf{psP}^{\mathsf{dt}}(\mathit{FIND1}) = \Omega(\sqrt{n})$ *as well.*

The proof of the above theorem follows from Theorem 3 together with our main theorem below which proves a $\sqrt{n}$ separation between randomized and pseudo-deterministic quantum query complexity for a broad family of $\mathsf{TFNP}^{\mathsf{dt}}$ search problems that are associated with expanding unsatisfiable CNF formulas.

▶ **Definition 5.** *Let $C = C_1 \wedge \ldots \wedge C_m$ be an unsatisfiable k-CSP problem over Boolean variables $x_1, \ldots, x_n$, where each C_i is a constraint involving at most k variables. The search problem associated with C, $\mathcal{S}_C \subseteq \{0,1\}^n \times [m]$, consists of all pairs (x, i) such that $x \in \{0,1\}^n$, and $C_i(x) = 0$. A query algorithm for $\mathcal{S}_C$ on input x outputs a constraint C_i that is falsified by x.*

$\mathcal{S}_C$ has been studied extensively in proof complexity and communication complexity, where lower bounds on its deterministic query complexity have been used to obtain, via lifting, exponential lower bounds on the monotone circuit size of a monotone function associated with C. Similarly, these search problems play a prominent role in lower bounds in proof complexity and extended formulations (e.g., [10, 8, 11]).

▶ **Definition 6.** *Let $C = C_1 \wedge \ldots \wedge C_m$ be a k-CSP over Boolean variables $x_1, \ldots, x_n$. Consider the bipartite graph with m left vertices (one for each constraint) and n right vertices (one for each variable), such that (i, j) is an edge if and only if variable x_j occurs in constraint C_i. C is (r, s)-expanding if for every subset $S \subseteq [m]$ of left vertices, $|S| \leq r$, the set of right elements adjacent to S, $N(S)$, has size at least s.*

▶ **Theorem 7.** *Let C be a k-CNF or k-XOR over $x_1, \ldots, x_n$, that is $(\epsilon n, c)$-expanding for $\epsilon = 1/100$, $c \geq k/2$. Then* $\mathsf{psQ}^{\mathsf{dt}}(\mathcal{S}_C) = \Omega(\sqrt{n})$.

▶ **Corollary 8.** *Let $k \geq 3$, $c = c(k)$ a sufficiently large constant, n sufficiently large and $m = cn$. Let $\mathcal{C}_n^m$ be the distribution over random k-CNF (k-XOR) formulas with m constraints, where each constraint is chosen uniformly at random from the set of all size-k clauses (size-k XOR formulas). Then with probability $1 - o(1)$, a random C drawn from $\mathcal{C}_n^m$ will have* $\mathsf{BPP}^{\mathsf{dt}}(\mathcal{S}_C) \in \mathcal{O}(1)$, *and* $\mathsf{psQ}^{\mathsf{dt}}(\mathcal{S}_C) \in \Omega(\sqrt{n})$.

Proof of Corollary 8. For $c = c(k)$ a sufficiently large constant, with high probability a random k-CNF from $\mathcal{C}_n^m$ will have the property that every assignment x falsifies a constant fraction of the clauses of C. Assuming that C drawn from $\mathcal{C}_n^m$ satisfies this property, there is a constant depth randomized query algorithm for $\mathcal{S}_C$. Namely, pick a random subset S of $O(1)$ clauses from C, and query all of the variables underlying these clauses. Output the first clause from S that is falsified, if one exists, and otherwise output error. Since every assignment falsifies a constant fraction, ϵ, of clauses, the probability that all clauses in S are satisfied (so the algorithm errs) is at most $(1-\epsilon)^{|S|}$, so we can choose $|S|$ to be a sufficiently large constant so that the probability of error is at most $1/3$. Therefore with probability $1 - o(1)$, $\mathsf{BPP}^{\mathsf{dt}}(\mathcal{S}_C) = O(1)$. For the lower bound, a standard calculation shows that a random k-CNF (or k-XOR) formula will be $(n/100, k/2)$ expanding with high probability. Therefore by Theorem 7, $\mathsf{psQ}^{\mathsf{dt}}(\mathcal{S}_C) = \Omega(\sqrt{n})$. ◀

Our lower bound proceeds by first proving linear lower bounds on the Fourier degree of $\mathcal{S}_C$, by a reduction to known lower bounds on the Nullstellensatz degree of refuting C. With this linear degree bound at hand, we obtain our lower bound by applying Huang's sensitivity theorem (showing that sensitivity and degree are quadratically related) together with the fact that sensitivity lower bounds randomized query complexity.

A alternative proof which also gives us the $\sqrt{n}$ quantum pseudo-deterministic lower bound can be obtained by combining our linear degree bound for $\mathcal{S}_C$ with the result of [2], showing that quantum query complexity is quadratically related to degree. We begin with the definition of Nullstellensatz degree.

▶ **Definition 9.** *For $C = C_1 \wedge \ldots \wedge C_m$ be an unsatisfiable k-CNF formula, we define the standard representation of C by a set of $m+n$ polynomial equations (each of degree at most k) such that C is satisfiable if and only if there is an assignment such that all polynomials evaluate to zero. For a clause C_i, let C_i^+ denote the set of variables occurring positively in C_i and let C_i^- denote the set of variables occurring negatively in C_i; with this notation we can write $C_i = \bigvee_{x_j \in C_i^+} x_j \vee \bigvee_{x_j \in C_i^-} \overline{x}_j$. From C_i define the polynomial*

$$Q(C_i) = \Pi_{x_j \in C_i^+}(1 - x_j)\Pi_{x_j \in C_i^-} x_j.$$

Let $\mathcal{Q}(C) = \{Q_1, \ldots, Q_{m+n}\}$ denote the set of polynomials $\{Q(C_i) : C_i \in C\} \cup \{x_i^2 - x_i : i \in \ [n]\}$.

▶ **Definition 10.** *Let C be an unsatisfiable k-CNF formula and let $\mathcal{Q}(C)$ be the associated set of polynomials as in Definition 9. A Nullstellensatz refutation of C (over a field F) is a set of polynomials $\{P_i\}, i = 1 \ldots m+n$ such that*

$$\sum_{i \in [m+n]} P_i Q_i = 1$$

holds over the ring $F[x_1 \ldots x_n]$. Any such sequence $\{P_i\}$ is called a Nullstellensatz refutation of C, and the degree of the refutation is $max_{i \in [m+n]} \boldsymbol{d}(P_i)$. The Nullstellensatz degree of C, $\mathsf{NS}(C)$, *is the minimum degree over all Nullstellensatz refutations of C.*

We will use the following linear lower bounds on the Nullstellensatz degree for random formulas.

▶ **Theorem 11** ([21, 6, 3])**.** *Let $C = C_1 \wedge \ldots \wedge C_m$ be a k-CNF or k-XOR formula over $x_1, \ldots, x_m$, with $m = O(n)$ and such that C is $(\epsilon n, k/2)$-expanding. Then $\mathsf{NS}(C) = \Omega(n)$ (over any field).*

The next lemma shows that $\boldsymbol{d}(\mathcal{S}_C)$ is lower bounded by Nullstellensatz degree (over any field).

▶ **Lemma 12.** *Let C be an unsatisfiable k-CNF formula, and let f be any function solving the search problem $S_{\mathcal{C}}$. Then $\mathsf{NS}(C) \leq \boldsymbol{d}(f)$. Conversely, for any finite field $\mathbb{F}$, $O(\boldsymbol{d}(\mathcal{S}_C) \log n) \leq \mathsf{NS}(C) \leq \boldsymbol{d}(\mathcal{S}_C)$.*

Proof of Lemma 12. Suppose that $f : \{0,1\}^m \to [m]$ solves the search problem for C, and let $d = \boldsymbol{d}(f) = \max_i \boldsymbol{d}(f^i)$. Consider the polynomial $\sum_{i \in [m]} f^i Q_i$. First, we claim that the polynomial $\sum_{i \in [m]} f^i Q_i$ evaluates to 1 on all inputs in $\{0,1\}^n$. Since the functions $\{f^i \mid i \in [m]\}$ form a partition of $\{0,1\}^n$, for every $\alpha \in \{0,1\}^n$, there is exactly one $i \in [m]$ such that $f^i(\alpha) = 1$, and for all other $j \neq i$, $f^j(\alpha) = 0$. Since $f^i(\alpha) = 1$ implies $C_i(\alpha) = 0$, it follows that $Q_i(\alpha) = 1$. Thus, $\sum_{i \in [m]} f^i Q_i$ evaluates to 1 for all $\alpha \in \{0,1\}^n$ as claimed. Now using the axioms $\{Q_{m+1}, \ldots, Q_{m+n}\} = \{x_i^2 - x_i \mid i \in [n]\}$, we can derive the identically 1 polynomial as:

$$\sum_{i \in [m]} f^i Q_i + \sum_{i \in [m+1, m+n]} h_i Q_i,$$

where each h_i is of degree at most d. Thus we have a degree d Nullstellsatz refutation of C, so $\mathsf{NS}(C) \leq \boldsymbol{d}(f)$.

In the other direction, let $\mathcal{Q}(C)$ be the set of polynomials associated with C, and assume that we have degree-d polynomials $P_1, \ldots, P_m$ such that $\sum_i P_i Q_i = 1 (mod 2)$, where $\mathbb{F} = GF(2)$. (A similar argument works over any finite field.) We want to define polynomials f^i such that: $f^i(\alpha) = 1$ implies that $C_i(\alpha) = 1$ and for all C_j, $j < i$, $C_j(\alpha) = 0$. For any α, we know that $\sum_i P_i(\alpha) Q_i(\alpha) = 1$. In order to determine whether or not $f^i(\alpha) = 1$, we want to do a binary search in order to find a term $P_i(\alpha) Q_i(\alpha)$ that evaluates to 1. For example suppose that $m = 16$. Then since $\sum_{i=1}^{16} P_i(\alpha) Q_i(\alpha)$ is odd either (a) $\sum_{i=1}^{8} P_i Q_i$ is odd, or (b) $\sum_{i=9}^{16} P_i Q_i$ is odd. If (a) is odd, then we recurse on the left (smaller) side and otherwise if (a) is even then we recurse on the right side. Viewing the binary search as a decision tree, at the root we query $\sum_{i=1}^{8} P_i Q_i$ and if it evaluates to 1 we go left and otherwise we go right. This gives a height $\log m$ decision tree where internal vertices are labelled with degree d polynomials, and the leaves are labelled with the index $i \in [m]$ such that $P_i(\alpha) = 1$. Let p_i be the path from the root to the leaf labelled by i. We can define a polynomial f^i associated with p_i which is the product of $\log m$ polynomials (along the path) such that $f^i(\alpha) = 1$ if and only if α is consistent with the path p_i. Thus the f^i's solve the search problem $\mathcal{S}_C$ and have degree $d \log m$. ◀

Proof of Theorem 7. Let $C = C_1 \wedge \ldots \wedge C_m$ be a k-CNF or k-XOR CSP over $x_1, \ldots, x_n$ that is $(\epsilon n, k/2)$ expanding, where $m = O(n)$. By Theorem 11, $\mathsf{NS}(C) = \Omega(n)$, and thus by Theorem 12, $\boldsymbol{d}(\mathcal{S}_C) = \Omega(n)$.

Assume that $\mathcal{T}$ is a pseudo-deterministic query algorithm for $\mathcal{S}_C$. Then for every input $x \in \{0,1\}^n$, there is a canonical solution $f(x)$ such that $\mathcal{T}(x)$ outputs $f(x)$ with probability at least $2/3$. Thus $\mathcal{T}$ is a randomized query algorithm for f. By Nisan [28] $\boldsymbol{s}(f) = O(\mathsf{BPP}^{\mathsf{dt}}(f))$, and by Huang [25], $\boldsymbol{s}(f) \geq \sqrt{\boldsymbol{d}(f)}$. Thus since $\boldsymbol{d}(\mathcal{S}_C) = \Omega(n)$, it follows that $\mathsf{BPP}^{\mathsf{dt}}(f) = \Omega(\sqrt{n})$, and thus $\mathsf{psP}^{\mathsf{dt}}(\mathcal{S}_C) = \Omega(\sqrt{n})$. ◀

Proof of Theorem 4. This follows from Theorem 7 and Theorem 3. We apply Theorem 3 to the search problem $S_{\mathcal{C}}$. By Theorem 7, we have that $r(n) = O(1)$ and $q(n) = \Omega(\sqrt{n})$. Also $v(n) = O(1)$ since we can verify a solution to $S_{\mathcal{C}}$ by just querying the variables in the clause that is the candidate solution. Clearly, r, q, v are all reasonable, hence it follows from Theorem 3 that FIND1 has pseudo-deterministic query complexity $\Omega(\sqrt{n})$. ◀

We observe that our $\Omega(\sqrt{n})$ separation between pseudo-deterministic quantum query complexity and quantum query complexity is tight. Grover [23] discovered a quantum query algorithm of complexity $O(\sqrt{n})$ for solving the following search problem: Given an n-bit binary string x, the goal is to find a coordinate i such that $x_i = 1$ (or to indicate that no such i exists). (See e.g., [9] for a survey.) This implies that FIND1 has pseudo-deterministic quantum query complexity $\tilde{O}(\sqrt{n})$, using a simple binary search algorithm to find the lexicographically first 1. For the quantum lower bound, we combine the result of [2] that quantum query complexity is at least $\sqrt{deg(f)}$ with Theorem 7.

5 Applications

5.1 A Related Combinatorial Problem

Our pseudo-deterministic query lower bound is related to a natural problem in extremal graph theory, which states that any proper coloring of the hypercube has high (block) sensitivity.

▶ **Definition 13.** *A proper coloring of the m-dimensional Boolean cube is any function $c : \{0,1\}^m - \{0^m\} \to [m]$ such that for all $\beta \in \{0,1\}^m - \{0^m\}$, $\beta_{c(\beta)} = 1$.*

▶ **Theorem 14.** *Let c be any proper coloring of the Boolean cube. Then there must exist $\beta \in \{0,1\}^m$ such that: (i) β contains at least a constant fraction of 1's, and (ii) β has block sensitivity $d = \Omega(\sqrt{m})$. That is, there are d disjoint blocks of inputs, $B_1, \ldots, B_d$ such that for all $i \in [d]$, $c(\beta) \neq c(\beta^{B_i})$.*

We remark that the above theorem implies a lower bound of $\Omega(\sqrt{n})$ on the pseudo-deterministic query complexity of FIND1.

Proof. At a high level, we will convert our sensitivity lower bound for the search problem associated with a random unsat k-XOR formula into a block sensitivity lower bound for the above coloring problem. Fix an expanding k-XOR formula C with $m = O(n)$ constraints and n variables such that for any assignment $\alpha \in \{0,1\}^n$, at least a constant fraction of the parity constraints are falsified by α. Further we will assume that the constraint-to-variable graph is expanding and in particular, for any subset $S \subseteq [n]$, there exists a large subset $S' \subseteq S$, $|S'| = O(|S|)$ such that for all $i \neq j \in S'$, the constraints containing x_i are disjoint from the constraints containing x_j.

First, we define a simple transformation that maps each input $\alpha \in \{0,1\}^n$ to an associated m-dimensional Boolean vector, $\beta(\alpha) \subseteq \{0,1\}^m$.

▶ **Definition 15.** *Let $\alpha \in \{0,1\}^n$. The constraint vector, $\beta(\alpha) \in \{0,1\}^m$ associated with α is defined as follows. For each $j \in [m]$, $\beta(\alpha)_m = 1$ if and only if $C_j(\alpha) = 0$. That is, the constraint vector associated with α has a 1 in coordinate j exactly when the j^{th} constraint of C is falsified by α. Let $S(C)$ denote the image of this map; that is, $S(C) \subseteq \{0,1\}^m$ is the set of all length m vectors that are constraint vectors for some $\alpha \in \{0,1\}^n$.*

Since C has the property that every assignment falsifies a constant fraction of the constraints in C, it follows that for every α, $\beta(\alpha)$ contains at least a constant fraction of 1's. Now consider a pair of adjacent assignments α and α^i where α^i is obtained from α by toggling the value of x_i, $i \in [n]$. Let $B(x_i) \subseteq [m]$ denote the set of coordinates j such that constraint C_j in C contains x_i. Because the constraints in C are all parity constraints, the constraint vector, $\beta(\alpha^i)$ associated with α^i can be obtained from $\beta(\alpha)$ by toggling the coordinates in $B(x_i)$. Thus for every $\alpha \in \{0,1\}^n$ and $i \in [n]$, we have:

$$\beta(\alpha^i) = (\beta(\alpha))^{B(x_i)},$$

where $\beta(\alpha)^{B(x_i)}$ is obtained by starting with $\beta(\alpha)$ and flipping the coordinates in $B(x_i)$.

Now suppose that $c : \{0,1\}^m \to [m]$ is a proper coloring of the m-dimensional Boolean hypercube. Then c restricted to the constraint vectors $S(C)$ defines a function $f_c : \{0,1\}^n \to [m]$ that solves the search problem associated with C. By the proof of Theorem 7, any function that solves the search problem for C has sensitivity $\Omega(\sqrt{n})$. Let $\alpha \in \{0,1\}^n$ be an input of maximal sensitivity, and let $S \subseteq [n]$, $|S| = \Omega(\sqrt{n})$, be the set of sensitive coordinates: for all $i \in S$, $f_c(\alpha) \neq f_c(\alpha^i)$.

By our assumption on C (which follows by expansion), there exists a subset $S' \subseteq S$ of size at least $\epsilon|S|$ such that the sets of coordinates/constraints, $\{B(x_i) \mid i \in S'\}$ are pairwise disjoint. Now we claim that $\beta(\alpha) \in \{0,1\}^m$ has block sensitivity $|S'|$, where the sensitive blocks are: $\{B(x_i) \mid i \in S'\}$.

First, by construction the blocks are pairwise disjoint. Secondly we want to show that for each $i \in [S']$, $c(\beta(\alpha)) \neq c(\beta(\alpha)^{B(x_i)})$. Since the constraints of C are parity constraints, flipping the value of any variable x_i flips the value of each constraint containing x_i. That is, the assignment α^i corresponds to the constraint vector $\beta(\alpha^i) = \beta(\alpha)^{B(x_i)}$. Since i is a sensitive coordinate for f_c with respect to α, $f_c(\alpha) \neq f_c(\alpha^i)$, and therefore $c(\beta(\alpha)) \neq c(\beta(\alpha)^{B(x_i)})$. ◀

We leave open the following conjecture which is a strengthening of the above theorem.

▶ **Conjecture 16.** *Let c be any proper coloring of the Boolean cube. Then there exists an assignment $\beta \in \{0,1\}^m$ such that β has at least a constant fraction of 1's and such that β has $\Omega(n)$ sensitivity.*

We also state another conjecture that strengthens the theorem in a different way. A vector $\beta \in \{0,1\}^m$ is *b-colorful* with respect to a proper coloring c if the set of colors associated with β plus all of the neighbors of β is at least b.

▶ **Conjecture 17.** *Let c be any proper coloring of the m-dimensional Boolean hypercube. then there exists $\beta \in \{0,1\}^m$ such that β has a constant fraction of 1's, and β is $\Omega(n)$-colorful.*

5.2 Size Lower Bounds and Pseudo-deterministic Resolution

The rich theory of TFNP and its subclasses (PPA, PPAD, PLS, etc) are defined based on the underlying combinatorial axiom required to *prove* the totality of functions in the class. Thus it is not surprising that there are strong connections between many TFNP subclasses and corresponding proof systems. For example it is known that FP is complete for the bounded arithmetic theory S^1_2 (in the sense that the TFNP problems definable in S^1_2 are the functions in FP), and similarly PLS is complete for the theory T^1_2.

The query complexity of subclasses of TFNP corresponds to studying the subclasses relative to an oracle. In the query world FP becomes P^{dt} and PLS becomes $\mathsf{PLS}^{\mathsf{dt}}$. The corresponding relativized systems of bounded arithmetic, $S^1_2(R)$ and $T^1_2(R)$, are uniform versions of the propositional proof systems TreeRes (Tree-like Resolution) and Res (dag-like Resolution).

For many weak propositional proof systems, there is an *equivalence* between minimal-size proofs of unsatisfiable formulas C and the query complexity of solving the search problem $\mathcal{S}_C$ in a corresponding query model. In this section we will use this equivalence to define *pseudo-deterministic* Resolution – a new notion that lies between ordinary Resolution and the much stronger notion of *Random Resolution*. Building on our pseudo-deterministic query lower bound, we exponentially separate *pseudo-deterministic* tree-like Resolution from Random Resolution.

5.2.1 Pseudo-deterministic Resolution

We start by defining some dag-like query models and review the known equivalences between Resolution and its common subsystems and their query model counterparts.

▶ **Definition 18** (Conjunction DAGs)**.** *Consider the n-bit input domain $\{0,1\}^n$ and let $\mathcal{F}$ be the set of all conjunctions of literals over the n input variables. An $\mathcal{F}$-DAG, Π, solving a search problem $\mathcal{S} \subseteq \{0,1\}^n \times [m] \in \mathsf{TFNP}^{\mathsf{dt}}$ is a directed acyclic graph of fanout at most two, where each node v is associated with a function $f_v \in \mathcal{F}$. (The set $f_v^{-1}(1)$ is called the feasible set for v) and satisfying the following conditions:*

- *There is a distinguished root node r and $f_r = 1$ (the constant 1 function).*
- *For each non-leaf node v with children u, u', we have $f_v^{-1}(1) \subseteq f_u^{-1}(1) \cup f_{u'}^{-1}(1)$.*
- *Each leaf node v is labelled with an output $o_v \in [m]$ such that $f_v^{-1}(1) \subseteq \mathcal{S}^{-1}(o_v)$.*

The size of Π is the number of vertices in the dag. The width of Π is the maximum width of a conjunction associated with a node of Π.

▶ **Theorem 19.** *Let C be an unsatisfiable k-CNF formula and let $\mathcal{S}_C$ be the associated search problem. The following equivalences hold:*

1. *The minimum width Resolution refutation of C is equivalent (to within constant factors) to the minimum width of a conjuction-DAG for $\mathcal{S}_C$ [31, 32].*
2. *The minimum size Resolution refutation of C is equivalent (to within constant factors) to the minimum size conjunction-DAG for $\mathcal{S}_C$. [31, 32].*
3. *The minimum size Regular Resolution refutation of C is equivalent to the minimum-size read-once Branching program for $\mathcal{S}_C$ [27].*
4. *The minimum size tree-like Resolution refutation of C is equivalent to the minimum size deterministic decision tree for $\mathcal{S}_C$.*

With these equivalences in hand, we easily obtain natural *pseudo-deterministic* versions of these proof systems, stated next for Resolution and its common subsystems.

▶ **Definition 20.** *Let C be an unsatisfiable k-CNF formula. A pseudo-deterministic tree-like Resolution refutation of C is a pseudo-deterministic decision tree for $\mathcal{S}_C$. Let the minimal-size pseudo-deterministic* TreeRes *refutation for C be equal to $\mathsf{psP}^{\mathsf{dt}}(\mathcal{S}_C)$. Similarly the pseudo-deterministic regular Resolution complexity of C is the pseudo-deterministic read-once branching program size for $\mathcal{S}_C$, and the pseudo-deterministic Resolution complexity of C is the pseudo-deterministic dag-like query complexity of $\mathcal{S}_C$.*

It is not hard to see that pseudo-deterministic TreeRes, Res refutations are sound, and at least for TreeRes, pseudo-deterministic proofs can be efficiently verified. We want to compare pseudo-deterministic Resolution (and its subsystems) to Random Resolution (defined in [7] (following a suggestion by S. Danchev), where it was motivated by the open problem of proving a strict depth hierarchy for bounded-depth Frege systems.

▶ **Definition 21.** *A random Resolution refutation* (RR) *of an unsat CNF formula F over $x_1, \ldots, x_n$ is a distribution π on pairs (w_i, E_i), $i \in [q]$ such that:*

1. *Each E_i is a CNF formula in $x_1, \ldots, x_n$;*
2. *For each $i \in [q]$, w_i is a Resolution refutation of $F \wedge E_i$;*
3. *For all $\alpha \in \{0,1\}^n$, $Pr_{i \sim \pi}[E_i(\alpha) = 1] \geq 3/4$*

The size of the proof is $\sum_i (|w_i| + size(E_i))$.

Similarly one can define random tree-like and regular) Resolution proofs, where now each w_i is a tree-like (regular) Resolution refutation of $F \wedge E_i$. Random Cutting Planes refutations were also defined in a similar manner by Sokolov [32].

Random Resolution turns out to be quite powerful, as is evidenced by the fact that random unsatisfiable k-CNF formulas have short RR refutations, and even short random tree-like refutations. For a random k-CNF with sufficiently many clauses, every assignment will falsify a constant fraction of the clauses and thus we can create the distribution $\{(w_i, E_i), i \in [q]\}$ to mimic the randomized strategy for finding a violated clause: for each clause C_i in F, let E_i be the negation of C_i. Clearly each formula $F \wedge E_i$ is unsatisfiable and has a very short tree-like proof, since C_i together with E_i is contradictory. Secondly since every assignment is falsified by $1 - \epsilon$ fraction of clauses, $Pr_i[E_i(\alpha) = 1] \geq 1 - \epsilon$. Using this fact together with the PCP theorem, Pudlak and Thapen [7] observed that no polynomial-time verifier, or even a randomized verifier, can check a RR refutation (or even a tree-like refutation) efficiently unless P = NP (or BPP = NP).

The following theorem shows that a natural random distribution of formulas exponentially separates pseudo-deterministic TreeRes size from random TreeRes size.

▶ **Theorem 22.** *For all constant $k \geq 3$, there exists a family of k-CNF (k-XOR) formulas $\{F_n\}_{n \in \mathbb{N}}$ such that:*

- *The formulas F_n admit linear-size random* TreeRes *refutations;*
- *For n sufficiently large and $m = O(n)$ sufficiently large, any pseudo-deterministic* TreeRes *refutation of F_n requires size $exp(\Omega(\sqrt{n}))$.*

Proof. The formula F_n will be obtained by two steps. First we will choose a $k/2$-CNF ($k/2$-XOR) formula, f_n, such that its clause variable graph is expanding. For example a random formula chosen with $m = O(n)$ clauses (XOR equations) will suffice. Secondly we obtain F_n by composing f_n with a 2-bit gadget g. That is, each variable x_i will be replaced by $g(x_i^a, x_i^b)$, where x_i^a, x_i^b are twin variables replacing x_i. For f_n an expanding CNF formula, we define the gadget g to be the parity function, $g(a,b) = a \oplus b$ and for f_n an XOR formula, $g(a,b) = a \vee b$. We then rewrite $f_n \circ g_n$ as a k-CNF, clause-by-clause. Since f_n is a $k/2$-CNF formula with n variables and m clauses, F_n will be a k-CNF formula with $2n$ variables and $m \cdot 2^k$ clauses.

Fix n sufficiently large, and let $\mathcal{T}$ be a pseudo-deterministic TreeRes refutation of F_n, where each tree $T_i \in \mathcal{T}$ has size at most s. Define $size(\mathcal{T})$ to be the sum of the sizes of all trees in $\mathcal{T}$. First we remark that by Newman's theorem, we can assume that the number of trees (i.e. the amount of randomness required) is polynomial in the size of each tree, and thus counting the total size of all trees combined, rather than the max tree size, is justified.

Let $\mathcal{R} \subseteq \{0,1,*\}^{2n}$ be the uniform distribution over the family of restrictions ρ such that: for all $i \in n$, exactly one variable in the pair $(x_i^a,\, x_i^b)$ is set to 0 or 1 and the other variable in the pair is set to $*$. That is, $(x_i^a|_\rho, x_i^b|_\rho) \subseteq \{(*,0),(*,1),(0,*),(1,*)\}$. Let $\mathcal{T}$ be a size s pseudo-deterministic TreeRes refutation of F_n. Let $terms(\mathcal{T})$ be the set of all terms (partial assignments) associated with all paths in all trees, T_i, and let $wide \subseteq terms(\mathcal{T})$ be those terms in $terms(\mathcal{T})$ of width at least w, $w = O(\sqrt{n})$.

For a fixed term $t \in wide(\mathcal{T})$, the probability that a random $\rho \in \mathcal{R}$ does not set t to zero is at most $(3/4)^w$. By the union bound, the probability that there exists $\rho \in \mathcal{R}$ that sets all wide terms to zero is at least $1 - s(3/4)^w$ which is greater than zero for $\log s = O(w)$. Thus there exists a restriction setting all wide terms of $\mathcal{T}$ to zero.

Applying ρ to $\mathcal{T}$, and to F_n, we obtain a pseudo-deterministic TreeRes refutation $\mathcal{T}'$ of $F_n|_\rho$ of size at most n and of depth at most w. Since $F_n|_\rho$ is just a copy of f_n, by the expansion properties of f_n, we can apply Theorem 7 which states that any pseudo-deterministic decision tree for f_n must have depth $\Omega(\sqrt{n})$, and thus $s = \Omega(exp(\sqrt{n}))$. ◀

5.2.2 Pseudo-deterministic Algebraic Proofs

By the relationship between low-degree polynomials solving $\mathcal{S}_C$ and low-degree Nullstellensatz refutations of C given in Lemma 12, we can define pseudo-deterministic Nullstellensatz refutations to be pseudo-deterministic polynomials solving $\mathcal{S}_C$.

▶ **Definition 23.** *Let C be an unsatisfiable k-CNF and let $\mathcal{S}_C$ be the corresponding search problem. Then a pseudo-deterministic degree d Nullstellensatz refutation over $\mathbb{F}$ is a distribution over polynomials $\mathcal{P} = \{P^1, \ldots P^q\}$ over $\mathbb{F}$ such that each $P^i : \{0,1\}^n \to [m]$ has degree at most d and such that there exists a function f solving $\mathcal{S}_C$ such that $\mathcal{P}$ probabilistically computes f: for all inputs $x \in \{0,1\}^n$, $Pr_{i \in [q]}[P^i(x) = f(x)] \geq 3/4$.*

We note that the degree of Nullstellensatz refutations of C over $\mathbb{F}_2$ have also been shown to be equivalent to the $\mathsf{PPA}^{\mathsf{dt}}$ query complexity of $\mathcal{S}_C$ [19]. (Intuitively there is a degree-d $\mathsf{PPA}^{\mathsf{dt}}$ query algorithm for $\mathcal{S}$ if there is a depth-d decision tree reduction from $\mathcal{S}$ to an instance of PPA. See [19] for a formal definition.)

Over the reals, $\Omega(n^\epsilon)$ lower bounds for pseudo-deterministic Nullstellensatz refutations follow from our pseudo-deterministic query lower bound for $\mathcal{S}_C$ for random C. This is because a family of polynomials computing a function f that solves the search problem implies the existence of an *approximate* polynomial of the same degree for solving f (that is, polynomials p_i that pointwise are within ϵ of $f^i(x)$ for all x.) And polynomial degree is polynomially related to approximate-degree for Boolean functions over the reals [29].

It is interesting to study similar relationships for other, stronger algebraic proof systems such as Sherali Adams (SA) and Sum-of-Squares. Can low degree proofs be characterized or lower bounded by the complexity of a family of pseudo-deterministic algebraic objects for solving the associated search problem?

6 Average Case Pseudo-deterministic Simulations

In this section, we study pseudo-deterministic simulations of randomized query algorithms in the average-case setting. We first show that for any search problem $\mathcal{S}$, the existence of zero-error randomized algorithms with low query complexity on average over a distribution D implies the existence of *deterministic* algorithms with low query complexity on average over D (and hence also of zero-error pseudo-deterministic algorithms). In the bounded-error setting, we show that for any search problem $\mathcal{S}$ solving an approximation problem, the existence of bounded-error randomized algorithms with low query complexity implies that for any D, there is a bounded-error pseudo-deterministic algorithm with low query complexity on average over D.

We first define what it means to solve search problems efficiently on average by a pseudo-deterministic algorithm. We adopt the strongest reasonable definition of average-case solvability: the algorithm must be pseudo-deterministic and solve the problem correctly on *every* input, and must have low query complexity on average over the distribution on inputs (and randomness of the algorithm). Adopting a strong notion of solvability makes our results stronger, as our results are mainly simulation results.

▶ **Definition 24.** *Let D be a distribution over $\mathcal{X} \subseteq \{0,1\}^n$. We say that a search problem $\mathcal{S}$ over domain $\mathcal{X}$ is solvable on average over D by a bounded-error pseudo-deterministic query algorithm with complexity q if there is a randomized query algorithm A that is bounded-error pseudo-deterministic and solves $\mathcal{S}$ correctly with probability $\geq 2/3$ on each input in $\mathcal{X}$, and moreover the expected number of queries of A (over the randomness of A and the distribution D) is at most q. Similarly, we say that a search problem $\mathcal{S}$ over domain $\mathcal{X}$ is solvable on average over D by a zero-error pseudo-deterministic query algorithm with complexity q if there is a randomized query algorithm A that is zero-error pseudo-deterministic and solves $\mathcal{S}$ correctly with probability 1 on each input in $\mathcal{X}$, and moreover the expected number of queries of A (over the randomness of A and the distribution D) is at most q. If A is deterministic, we say that $\mathcal{S}$ is solvable on average over D by a deterministic query algorithm with complexity q.*

We first show that the canonical problem FIND1 (which is solvable efficiently by zero-error query algorithms) has low average-case deterministic query complexity over any distribution.

▶ **Proposition 25.** *Let D be any distribution on the domain of FIND1 restricted to n-bit inputs. FIND1 is solvable on average over D by a deterministic query algorithm with complexity $\log(n) + 1$.*

Proof. Let D be any distribution on the domain of FIND1 restricted to n-bit inputs. Let R be a subset of $[n]$ of size $\log(n)$ where R is chosen uniformly at random over all such subsets. Since FIND1 is defined over inputs x with $|x| \geq n/2$, we have that for each x in the domain of FIND1, the probability that there is a $j \in R$ such that $x_j = 1$ is at least $1 - 1/n$. By averaging, there is a subset B of $[n]$ of size $\log(n)$ such that with probability at least $1 - 1/n$ over D, $x_j = 1$ for some $j \in B$ when x is chosen from D.

Consider the following deterministic query algorithm A. A queries the indices in B in lexicographic order, and outputs the first such index j for which $x_j = 1$, if such an index exists. If no such index exists, A queries the indices in $[n] \setminus B$ in lexicographic order, and outputs the first index j for which $x_j = 1$. Since FIND1 is only defined over n-bit inputs with at least one 1, this query algorithm is correct. Call an input x in the domain of FIND1 "good" if there is a $j \in B$ such that $x_j = 1$. With probability at least $1 - 1/n$ over x chosen from D, x is good and the query algorithm A uses at most $\log(n)$ queries. When x is not good, A uses at most n queries. Thus the query complexity is at most $(1 - 1/n) \cdot \log(n) + n \cdot 1/n \leq \log(n) + 1$ on average over D. ◀

Next we significantly generalize Proposition 25 and show that efficient average-case solvability by zero-error randomized algorithms is in fact equivalent to efficient average-case solvability by deterministic algorithms (and hence also by zero-error pseudo-deterministic algorithms).

▶ **Theorem 26.** *Let $\mathcal{S}$ be a total search problem over domain $\mathcal{X} \subseteq \{0,1\}^n$, D a distribution over $\mathcal{X}$, and $q : \mathbb{N} \to \mathbb{N}$ a function. The following are equivalent:*

1. *$\mathcal{S}$ is solvable on average over D by a zero-error query algorithm with complexity $O(q(n)\mathsf{polylog}(n))$.*
2. *$\mathcal{S}$ is solvable on average over D by a zero-error pseudo-deterministic query algorithm with complexity $O(q(n)\mathsf{polylog}(n))$.*
3. *$\mathcal{S}$ is solvable on average over D by a deterministic query algorithm with complexity $O(q(n)\mathsf{polylog}(n))$.*

Proof. The third item trivially implies the second, and the second item trivially implies the first. We show that the first item implies the third.

Suppose $\mathcal{S}$ is solvable on average over D by a zero-error query algorithm with complexity $r(n) = O(q(n)\mathsf{polylog}(n))$. This implies that there is a distribution D' over deterministic query algorithms such that for every input x in $\mathcal{X}$, a query algorithm A chosen from D' solves $\mathcal{S}$ with probability at least $2/3$ over D', and moreover the expected number of queries over A chosen from D' and x chosen from D is at most $r(n)$. Without loss of generality, we can assume that D' is uniform over a multi-set Y of deterministic query algorithms. If this multi-set has size K, sampling from D' is equivalent to sampling uniformly from $[K]$. From now on, we assume a bijection between $[K]$ and Y, and also assume without loss of generality that $K \geq 4\log(n)$.

For positive integral t, define an input $x \in \mathcal{X}$ to be t-good if it is the case that with probability at least $1/6$ over choice of A from D', A solves S correctly on x making at most $2tr(n)$ queries. We argue that for each t, x chosen from D is t-good with probability at least $1 - 1/t$. The proof is by contradiction. Suppose this were not the case. Then for some positive integer t, with probability greater than $1/t$ over x chosen from D, x is not t-good. If x is not t-good, then with probability at least $5/6$ over choice of A from D', A either does not return a solution for S or makes more than $2tr(n)$ queries. Since for any $x \in \mathcal{X}$, A solves x with probability at least $2/3$, it must be the case that with probability at least $1/2$ over choice of A from D', A makes more than $2tr(n)$ queries on x when x is not t-good. Since the probability over D that x is not t-good is greater than $1/t$, this implies that when x is sampled from D and A from D', the expected number of queries is greater than $r(n)$, in contradiction to the assumption that the zero-error query algorithm corresponding to D' has complexity at most $r(n)$.

Now consider a t-good $x \in \mathcal{X}$. Say that $k \in [K]$ is t-suitable for x if running the k'th deterministic query algorithm from Y on x succeeds in solving $\mathcal{S}$ on x while making at most $2tr(n)$ queries. Since x is t-good, k chosen uniformly from $[K]$ is suitable for x with probability at least $1/6$. Let R be a subset of $[K]$ of size $4\log(n)$ chosen uniformly at random from all subsets of this size. With probability at least $1 - 1/n$, R contains $j \in [K]$ such that j is t-suitable for x. Say that R is t-suitable for x if this is the case.

Let $\mu(x)$ be the smallest positive integer t such that x is t-good. For any x, we have that $\mu(x) \leq n$.

By averaging, there is a subset B of $[K]$ of size $4\log(n)$ such that with probability at least $1 - 1/n$ over x sampled from D, B is $\mu(x)$-suitable for x. Consider the query algorithm A that works as follows. It runs the query algorithms corresponding to the elements of B in an interleaving fashion. Namely, if the elements of B are $b_1 \ldots b_{4\log(n)}$, it makes the first query of the b_j'th algorithm for each $j \in [4\log(n)]$ in order, then the second query for each of these algorithms, and so on until it has made enough queries for a given algorithm so that the algorithm outputs an answer. Naturally, it never repeats a query that it has already been made. Note that A halts after making at most $8\mu(x)\log(n)r(n)$ queries.

We bound the expected number of queries made by A for x chosen from distribution D. With probability at most $1/n$, B is not $\mu(x)$-suitable for x, and in this case A makes at most n queries on x. When B is $\mu(x)$-suitable, A halts and outputs a correct solution for $\mathcal{S}$ on x after making at most $8\mu(x)\log(n)r(n)$ queries. For each integer $i \in [\lceil \log(n) \rceil]$, we have that the probability over x sampled from D that $\mu(x) \leq 2^i$ is at least $1 - 1/2^i$. Computing the expectation of the running time of A by summing over $1 \leq i \leq \log(n)$ such that $2^i < \mu(x) \leq 2^{i+1}$, we have that the contribution to the expectation when B is $\mu(x)$-suitable is at most $(1/2 \cdot 2 + 1/4 \cdot 4 + \ldots)8\log(n)r(n) \leq 16(\log(n))^2 r(n)$. Thus, the total expectation is at most $16(\log(n))^2 r(n) + 1 = O(q(n)\mathsf{polylog}(n))$, as desired. ◀

Next, we turn to bounded-error average-case solvability. We show that the ϵ-HWE problem of approximating the Hamming weight of a string to within an additive term ϵ is solvable efficiently on average by bounded-error pseudo-deterministic query algorithms. We note that Goldreich, Goldwasser and Ron [13] showed an $\Omega(n)$ query lower bound for *worst-case* bounded-error pseudo-deterministic algorithms solving this problem.

▶ **Theorem 27.** *For any distribution D and any constant $\epsilon > 0$, ϵ-HWE is solvable on average over D by bounded-error pseudo-deterministic algorithms of complexity $O(\log(n)/\epsilon^2)$.*

Proof. We use the fact that, on any x, if we take a random sample of bits of size $q = O(\log(n)/\epsilon^2)$, the empirical average of ones of this sample differs from that of x by $\epsilon/4$ with probability at most $1/5n$, using a standard Chernoff-Hoeffding bound. By averaging, for every distribution D there must be some fixed such subset of bits with this property, when we take the expectation over random x from D. Call this subset A, and let $d_A(x)$ be the empirical estimate of the density of x based on the bits in A. Let B represent a uniform random subset of bits of size q, and let $d_B(x)$ represent the empirical estimate of the density of x based on the bits in B. Let $d(x)$ represent the actual density of x.

Let $p(x)$ be the function : $p(x) = d_A(x)$ if $Prob_B[|d_B(x) - d_A(x)| \leq \epsilon/2] > 1/5$, and $p(x) = d(x)$ otherwise. $p(x)$ is a fixed function of x, and it is always a good approximation to $d(x)$, since if it is not literally $d(x)$, it is $\epsilon/2$ close to $d_B(x)$ for most B, and a random B has $d_B(x)$ $\epsilon/2$ close to $d(x)$.

Consider the following algorithm for computing $p(x)$:

1. Compute $d_A(x)$.
2. Choose a random B of size q.
3. Compute $d_B(x)$
4. If $|d_B(x) - d_A(x)| \leq \epsilon/2$, return $d_A(x)$
5. Otherwise, query all bits of x, and compute $p(x)$. Return $p(x)$.

We claim that this algorithm returns $p(x)$ on any x except with probability at most $1/5$. Case 1: If $Prob_B[|d_B(x) - d_A(x)| \leq \epsilon/2] > 1/5$, then $p(x) = d_A(x)$. Then we either return the correct value in step 4, or we go on to compute the correct value in step 5. Either way, the algorithm is always correct.

Case 2: If $Prob_B[|d_B(x) - d_A(x)| \leq \epsilon/2] \leq 1/5$, then by definition, we return a value in step 4 with probability at most $1/5$. Thus, on such an input, with probability at least $4/5$, we go on to compute and return $p(x)$ by brute force in step 5.

Finally, we bound the expected number of bits queried by the algorithm over a random x from D. Over such random x, with probability $1 - 1/5n$, $|d_A(x) - d(x)| \leq \epsilon/4$, and for any x, with the same probability over B, $|d_B(x) - d(x)| \leq \epsilon/4$. If both of these happen, $|d_A(x) - d_B(x)| \leq \epsilon/2$ and the algorithm terminates in line 4 after making $2q$ queries. So the expected number of queries is at most $2q + 2/5n \cdot n = O(q)$. ◀

We observe that Theorem 27 generalizes to yield an efficient bounded-error pseudo-deterministic algorithm on average for any *approximation* problem with low bounded-error randomized query complexity. Given a metric Δ on a space $\mathcal{O}$ and a function $f : \mathcal{X} \to \mathcal{O}$, a search problem $\mathcal{S}$ with domain $\mathcal{X} \subset \{0,1\}^n$ and range $\mathcal{O}$ is said to be the ϵ-approximation problem for f if the solutions to $\mathcal{S}$ on input $x \in \mathcal{X}$ are all points $y \in \mathcal{O}$ for which $\Delta(y, f(x)) \leq \epsilon$.

▶ **Theorem 28.** *Let ϵ be a constant, $\mathcal{O}$ be a space with metric Δ and $f : \mathcal{X} \to \mathcal{O}$ be a function such that there is a randomized query algorithm with complexity q to $\epsilon/4$-approximate f. Then for any distribution D over $\mathcal{X}$ there is a pseudo-deterministic query algorithm A that ϵ-approximates f with query complexity $O(q\log(n))$ on average over D.*

The proof is a straightforward generalization of the proof of Theorem 27, and we therefore omit it.

We note that unlike with zero-error randomized query complexity, efficient bounded-error query algorithms are not in general efficiently simulated on average by deterministic query algorithms.

▶ **Proposition 29.** *Let D be any distribution assigning positive weight to every n-bit input. For any $\epsilon < 1/2$, ϵ-HWE has zero-error average-case query complexity $\Omega(n)$ over D.*

Proof. Let A be any zero-error query algorithm solving ϵ-HWE on average over D. A is a distribution over deterministic query algorithms. We note that for any deterministic query algorithm in the support of A, there is no path of length $< (1-2\epsilon)n$ with an output. If there were such a path, then the output would not be a correct ϵ-approximation either for the input on which all unqueried bits are 0 or for the input on which all unqueried bits are 1. Since both of these inputs have positive probability according to D, this would imply that A is not a correct zero-error algorithm for ϵ-HWE.

Now for any input x, since A is a correct zero-error query algorithm, it must return an output with probability at least $2/3$. By the previous paragraph, this means that the average number of queries over D is $\Omega(n)$. ◀

7 Open Problems

Here we record some open problems and directions that we leave open.

First, our lower bound is tight for pseudo-deterministic quantum query complexity. We conjecture that the bound for both FIND1 and $\mathcal{S}_C$ can be improved to $\Omega(n)$ for pseudo-determnistic query complexity. Such an improvement would have to bypass sensitivity (and approximate degree) since both incur a quadratic loss. Secondly, we leave open the question of proving superpolynomial or exponential lower bounds for pseudo-deterministic Resolution refutations.

More generally, it is very interesting to study pseudo-determinism in the realm of communication complexity. A pseudo-deterministic communication protocol for a search problem $\mathcal{S} = \{0,1\}^n \times \{0,1\}^n \times [m]$ is a distribution $\Pi = \{\pi_1, \ldots, \pi_q\}$ over deterministic protocols with the property that there exists a function $f_\Pi : \{0,1\}^n \times \{0,1\}^n \rightarrow [m]$ solving $\mathcal{S}$, where Π is a randomized protocol for f. That is, for every input $(x,y) \in \{0,1\}^n \times \{0,1\}^n$, $Pr_{i \in [q]}[\pi_i(x,y) = f(x,y)] \geq 3/4$.

Pseudo-deterministic communication complexity is interesting for several reasons. For Boolean functions an exciting body of work has culminated in what is now a nearly complete understanding of many query/degree measures and their pairwise relationships. In turn these query measures for Boolean functions have natural analogs in communication complexity, and lifting theorems give a way to lift query upper and lower bounds to their communication counterparts. However for search problems, we lack a good understanding of query measures and the relationships between them, and this in turn leads to a lack of clarity with respect to their communication analogs. For example, what is the analog of sensitivity and block-sensitivity for search problems? In [26] a notion called critical block sensitivity was defined, and used in [20, 18] to prove strong lower bounds on dynamic SOS and extended formulations on the exact compution of certain functions. Unfortunately critical block sensitivity is only defined for search problems containing inputs with a unique solution and therefore these tools cannot be used to prove inapproximability results. As a second example, extended formulation lower bounds have been proven by lifting semialgebraic degree lower bounds, but applying the lifting framework to prove *inapproximability* lower bounds is quite subtle, in large part due to a lack of relaxed/approximate/pseudo-deterministic notions of query complexity for search problems (e.g., approximate notions of Sherali-Adams (SA) and Sum-of-Squares (SOS) degree.) Since pseudo-deterministic algorithms are just randomized algorithms for computing *some* function solving the search problem, they are central to the study of relaxed query measures for search problems.

Secondly, the pseudo-determinism communication complexity of Karchmer-Wigderson search problems is particularly interesting. It is well known that deterministic communication complexity lower bounds on the KW search problems associated with a Boolean function is equivalent to formula size lower bounds (and dag-like communication lower bounds are equivalent to circuit lower bounds). This equivalence has been quite successful for proving lower bounds in monotone models of computation where lifting theorems in communication complexity have been applied to prove a variety of state-of-the art lower bounds for monotone formulas, monotone span programs, monotone circuits, as well as extended formulations (which are also a monotone model as they relate to nonnegative rank).

An exciting direction towards proving *nonmonotone* circuit/formula lower bounds is to further develop lower bound techniques for monotone models to apply to more functions – such as slice functions or all small "perturbations" of the function [24]. Related to this, we note that the communication complexity of monotone KW games is quite different than that of non-monotone KW games: whereas the (nonmonotone) KW game for *any* f has a trivial $O(\log n)$ pseudo-deterministic protocol, the *monotone* KW game (for monotone f) in general appears to be hard pseudo-deterministically.

A reasonable approach for separating pseudo-determininistic from randomized communication is lifting. We conjecture that the lifted/composed functions $\text{FIND1} \circ g^n$ and $\mathcal{S}_C \circ g^n$ require large pseudo-deterministic communication complexity for good choices of g (such as the index function). We note that standard lifting theorems won't work in a black-box way since the pseudo-deterministic protocol can have different canonical solutions for different inputs $(\vec{x},\vec{y}),(\vec{x}',\vec{y}')$ such that $g^n(\vec{x},\vec{y}) = g^n(\vec{x}'\vec{y}')$. Nonetheless, pseudo-deterministic communication lower bounds should be possible by combining lifting (in a non-blackbox way) with the right pseudo-deterministic query lower bound argument. In this respect we view our pseudo-deterministic query lower bounds as a first step towards obtaining a similar separation in communication complexity.

.

References

1 Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 863–876. ACM, 2016. doi:10.1145/2897518.2897644.

2 Scott Aaronson, Shalev Ben-David, Robin Kothari, and Avishay Tal. Quantum implications of huang's sensitivity theorem. *CoRR*, abs/2004.13231, 2020. arXiv:2004.13231.

3 Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 190–199, 2001.

4 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. doi:10.1145/502090.502097.

5 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

6 Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. Syst. Sci.*, 62(2):267–289, 2001.

7 Samuel R. Buss, Leszek Aleksander Kolodziejczyk, and Neil Thapen. Fragments of approximate counting. *J. Symb. Log.*, 79(2):496–525, 2014. doi:10.1017/jsl.2013.37.

8 Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. *J. ACM*, 63(4):34:1–34:22, 2016. doi:10.1145/2811255.

9 Richard Cleve. An introduction to quantum complexity theory. *Quantum Computation and Quantum Information Theory*, page 103–127, January 2001. doi:10.1142/9789810248185_0004.

10 Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Found. Trends Theor. Comput. Sci.*, 14(1-2):1–221, 2019. doi:10.1561/0400000086.

11 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 902–911. ACM, 2018. doi:10.1145/3188745.3188838.

12 Eran Gat and Shafi Goldwasser. Probabilistic search algorithms with unique answers and their cryptographic applications. *Electron. Colloquium Comput. Complex.*, 18:136, 2011. URL: http://eccc.hpi-web.de/report/2011/136.

13 Oded Goldreich, Shafi Goldwasser, and Dana Ron. On the possibilities and limitations of pseudodeterministic algorithms. In *4th Innovations in Theoretical Computer Science Conference, ITCS*, pages 127–138, 2013.

14 Shafi Goldwasser and Ofer Grossman. Bipartite perfect matching in pseudo-deterministic NC. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 87:1–87:13, 2017.

15 Shafi Goldwasser, Ofer Grossman, and Dhiraj Holden. Pseudo-deterministic proofs. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 17:1–17:18, 2018.

16 Shafi Goldwasser, Ofer Grossman, Sidhanth Mohanty, and David P. Woodruff. Pseudo-deterministic streaming. *CoRR*, abs/1911.11368, 2019. arXiv:1911.11368.

17 Shafi Goldwasser, Ofer Grossman, Sidhanth Mohanty, and David P. Woodruff. Pseudo-deterministic streaming. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 79:1–79:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.79.

18 Mika Göös, Rahul Jain, and Thomas Watson. Extension complexity of independent set polytopes. *SIAM J. Comput.*, 47(1):241–269, 2018. doi:10.1137/16M109884X.

19 Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, volume 124 of *LIPIcs*, pages 38:1–38:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.38.

20 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM J. Comput.*, 47(5):1778–1806, 2018. doi:10.1137/16M1082007.

21 Dima Grigoriev. Tseitin's tautologies and lower bounds for nullstellensatz proofs. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 648–652, 1998.

22 Ofer Grossman and Yang P. Liu. Reproducibility and pseudo-determinism in log-space. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 606–620, 2019.

23 Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996. doi:10.1145/237814.237866.

24 Pavel Hrubes. On ϵ-sensitive monotone computations. *Comput. Complex.*, 29(2):6, 2020. doi:10.1007/s00037-020-00196-6.

25 Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *CoRR*, abs/1907.00847, 2019. arXiv:1907.00847.

26 Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 233–248. ACM, 2012. doi:10.1145/2213977.2214000.

27 László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson. Search problems in the decision tree model. *SIAM J. Discret. Math.*, 8(1):119–132, 1995. doi:10.1137/S0895480192233867.

28 Noam Nisan. CREW PRAMs and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991.

29 Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994. doi:10.1007/BF01263419.

30 Igor Carboni Oliveira and Rahul Santhanam. Pseudodeterministic constructions in subexponential time. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 665–677. ACM, 2017. doi:10.1145/3055399.3055500.

31 A. A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya RAN. Ser. Mat.*, pages 201–224, 1995.

32 Dmitry Sokolov. Dag-like communication and its applications. In Pascal Weil, editor, *Computer Science - Theory and Applications - 12th International Computer Science Symposium in Russia, CSR 2017, Kazan, Russia, June 8-12, 2017, Proceedings*, volume 10304 of *Lecture Notes in Computer Science*, pages 294–307. Springer, 2017. doi:10.1007/978-3-319-58747-9_26.

A Simple Proof of a New Set Disjointness with Applications to Data Streams

Akshay Kamath
University of Texas at Austin, TX, USA

Eric Price
University of Texas at Austin, TX, USA

David P. Woodruff
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

The multiplayer promise set disjointness is one of the most widely used problems from communication complexity in applications. In this problem there are k players with subsets $S^1, \ldots, S^k$, each drawn from $\{1, 2, \ldots, n\}$, and we are promised that either the sets are (1) pairwise disjoint, or (2) there is a unique element j occurring in all the sets, which are otherwise pairwise disjoint. The total communication of solving this problem with constant probability in the blackboard model is $\Omega(n/k)$.

We observe for most applications, it instead suffices to look at what we call the "mostly" set disjointness problem, which changes case (2) to say there is a unique element j occurring in at least half of the sets, and the sets are otherwise disjoint. This change gives us a much simpler proof of an $\Omega(n/k)$ randomized total communication lower bound, avoiding Hellinger distance and Poincare inequalities. Our proof also gives strong lower bounds for high probability protocols, which are much larger than what is possible for the set disjointness problem. Using this we show several new results for data streams:

1. for ℓ_2-Heavy Hitters, any $O(1)$-pass streaming algorithm in the insertion-only model for detecting if an ε-ℓ_2-heavy hitter exists requires $\min(\frac{1}{\varepsilon^2} \log \frac{\varepsilon^2 n}{\delta}, \frac{1}{\varepsilon} n^{1/2})$ bits of memory, which is optimal up to a $\log n$ factor. For deterministic algorithms and constant ε, this gives an $\Omega(n^{1/2})$ lower bound, improving the prior $\Omega(\log n)$ lower bound. We also obtain lower bounds for Zipfian distributions.
2. for ℓ_p-Estimation, $p > 2$, we show an $O(1)$-pass $\Omega(n^{1-2/p} \log(1/\delta))$ bit lower bound for outputting an $O(1)$- approximation with probability $1 - \delta$, in the insertion-only model. This is optimal, and the best previous lower bound was $\Omega(n^{1-2/p} + \log(1/\delta))$.
3. for low rank approximation of a sparse matrix in $\mathbb{R}^{d \times n}$, if we see the rows of a matrix one at a time in the row-order model, each row having $O(1)$ non-zero entries, any deterministic algorithm requires $\Omega(\sqrt{d})$ memory to output an $O(1)$-approximate rank-1 approximation.

Finally, we consider strict and general turnstile streaming models, and show separations between sketching lower bounds and non-sketching upper bounds for the heavy hitters problem.

2012 ACM Subject Classification Theory of computation → Lower bounds and information complexity

Keywords and phrases Streaming algorithms, heavy hitters, communication complexity, information complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.37

Funding *Eric Price*: NSF grant CCF-1751040 (CAREER).
David P. Woodruff: NSF grant No. CCF-1815840 and a Simons Investigator Award.

Acknowledgements The authors would like to thank the anonymous reviewers of a previous version of this paper for helpful suggestions that significantly improved the presentation.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 37; pp. 37:1–37:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Communication complexity is a common technique for establishing lower bounds on the resources required of problems, such as the memory required of a streaming algorithm. The multiplayer promise set disjointness is one of the most widely used problems from communication complexity in applications, not only in data streams [3, 5, 18, 39, 48, 49, 19], but also in compressed sensing [67], distributed functional monitoring [77, 78], distributed learning [32, 52, 11], matrix-vector query models [71], voting [60, 61], and so on. We shall restrict ourselves to the study of set disjointness in the number-in-hand communication model, described below, which covers all of the above applications. Set disjointness is also well-studied in the number-on-forehead communication model, see, e.g., [38, 72, 7, 56, 21, 6, 69, 70], though we will not discuss that model here.

In the number-in-hand multiplayer promise set disjointness problem there are k players with subsets $S^1, \ldots, S^k$, each drawn from $\{1, 2, \ldots, n\}$, and we are promised that either:

1. the S^i are pairwise disjoint, or
2. there is a unique element j occurring in all the sets, which are otherwise pairwise disjoint.

The promise set disjointness problem was posed by Alon, Matias, and Szegedy [3], who showed an $\Omega(n/k^4)$ total communication bound in the blackboard communication model, where each player's message can be seen by all other players. This total communication bound was then improved to $\Omega(n/k^2)$ by Bar-Yossef, Jayram, Kumar, and Sivakumar [5], who further improved this bound to $\Omega(n/k^{1+\gamma})$ for an arbitrarily small constant $\gamma > 0$ in the one-way model of communication. These bounds were further improved by Chakrabarti, Khot, and Sun to $\Omega(n/(k \log k))$ in the general communication model and an optimal $\Omega(n/k)$ bound for 1-way communication. The optimal $\Omega(n/k)$ total communication bound for general communication was finally obtained in [39, 49].

To illustrate a simple example of how this problem can be used, consider the *streaming model*. The streaming model is one of the most important models for processing massive datasets. One can model a stream as a list of integers $i_1, \ldots, i_m \in [n] = \{1, 2, \ldots, n\}$, where each item $i \in [n]$ has a frequency x_i which denotes its number of occurrences in the stream. We refer the reader to [4, 66] for further background on the streaming model of computation.

An important problem in this model is computing the p-th frequency moment $F_p = \sum_{j=1}^n x_j^p$. To reduce from the promise set disjointness problem, the first player runs a streaming algorithm on the items in its set, passes the state of the algorithm to the next player, and so on. The total communication is $k \cdot s$, where s is the amount of memory of the streaming algorithm. Observe that in the first case of the promise we have $F_p \leq n$, while in the second case we have $F_p \geq k^p$. Setting $k = (2n)^{1/p}$ therefore implies an algorithm estimating F_p up to a factor better than 2 can solve promise set disjointness and therefore $k \cdot s = (2n)^{1/p} s = \Omega(n/(2n)^{1/p})$, that is, $s = \Omega(n^{1-2/p})$. For $p > 2$, this is known to be best possible up to a constant factor [14].

Notice that nothing substantial would change in this reduction if one were to change the second case in the promise to instead say: (2) there is a unique element j occurring in at least half of the sets, and the sets are otherwise disjoint. Indeed, in the above reduction, in one case we have $F_p \geq (k/2)^p$, while in the second case we have $F_p \leq n$. This recovers the same $\Omega(n^{1-2/p})$ lower bound, up to a constant factor. We call this new problem "mostly" set disjointness (MostlyDISJ).

While it is seemingly inconsequential to consider MostlyDISJ instead of promise set disjointness, there are some peculiarities about this problem that one cannot help but wonder about. In the promise set disjointness problem, there is a deterministic protocol solving the problem with $O(n/k \log k + k)$ bits of communication – we walk through the players one at a time, and each indicates if its set size is smaller than n/k. Eventually we must reach such a player, and when we do, that player posts its set to the blackboard. We then ask one other player to confirm an intersection. Notice that there always must exist a player with a set of size at most n/k by the pigeonhole principle. On the other hand, for the MostlyDISJ problem, it does not seem so easy to achieve a deterministic protocol with $O(n/k \log k + k)$ bits of communication. Indeed, in the worst case we could have up to $k/2$ players posting their entire set to the blackboard, and still be unsure if we are in Case (1) or Case (2).

More generally, is there a gap in the dependence on the error probability of algorithms for promise set disjointness versus MostlyDISJ? Even if one's main interest is in constant error probability protocols, is there anything that can be learned from this new problem?

1.1 Our Results

We more generally define MostlyDISJ so that in Case (2), there is an item occurring in $l = \Theta(k)$ of the sets, though it is still convenient to think of $l = k/2$. Our main theorem is that MostlyDISJ requires $\Omega(n)$ communication to solve deterministically, or even with failure probability e^{-k}.

▶ **Theorem 1.** MostlyDISJ *with n elements, k players, and $l = ck$ for an absolute constant $c \in (0, 1)$ requires $\Omega(\min(n, n\frac{\log(1/\delta)}{k}))$ bits of communication for failure probability δ.*

This result does not have any restriction on the order of communication, and is in the "blackboard model" where each message is visible to all other players. We note that as $c \to 1$, our lower bound goes to 0, but for any absolute constant $c \in (0, 1)$, we achieve the stated $\Omega(\min(n, n\frac{\log(1/\delta)}{k}))$ lower bound. We did not explicitly compute our lower bound as a function of c, as $c \to 1$.

Notice that for constant δ, Theorem 1 recovers the $\Omega(n/k)$ total communication bound for promise set disjointness, which was the result of a long sequence of work. Our proof of Theorem 1 gives a much simpler proof of an $\Omega(n/k)$ total communication lower bound, avoiding Hellinger distance and Poincare inequalities altogether, which were the main ingredients in obtaining the optimal $\Omega(n/k)$ lower bound for promise set disjointness in previous work. Moreover, as far as we are aware, an $\Omega(n/k)$ lower bound for the MostlyDISJ problem suffices to recover all of the lower bounds in applications that promise set disjointness has been applied to. Unlike our work, however, existing lower bounds for promise set disjointness do not give improved bounds for small error probability δ. Indeed, it is impossible for them to do so because of the deterministic protocol described above. We next use this bound in terms of δ to obtain the first lower bounds for deterministic streaming algorithms and randomized δ-error algorithms for a large number of problems.

We note that other work on deterministic communication lower bounds for streaming, e.g., the work of Chakrabarti and Kale [17], does not apply here. They study multi-party equality problems and it is not clear how to use their fooling set arguments to prove a lower bound for MostlyDISJ. One of the challenges in designing a fooling set is the promise, namely, that a single item occurs on a constant fraction of the players *and* all remaining items occur on at most one player. This promise is crucial for the applications of MostlyDISJ.

We now formally introduce notation for the data stream model. In the streaming model, an integer vector x is initialized to 0^n and undergoes a sequence of $L = \text{poly}(n)$ updates. The streaming algorithm is typically allowed one (or a few) passes over the stream, and the goal is to use a small amount of memory. We cannot afford to store the entire stream since n and L are typically very large. In this paper, we mostly restrict our focus to the *insertion-only* *model* where the updates to the vector are of the form $x \leftarrow x + \delta$ where $\delta \in \{e_1, \ldots, e_n\}$ is a standard basis vector. There are also the turnstile data stream models in which $x \leftarrow x + \delta$ where $\delta \in \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$. In the *strict turnstile model* it is promised that $x \geq 0^n$ at all times in the stream, whereas in the *general turnstile model* there are no restrictions on x. Therefore, an algorithm in the general turnstile model works also in the strict turnstile model and insertion-only models.

Finding Heavy Hitters

Finding the heavy hitters, or frequent items, is one of the most fundamental problems in data streams. These are useful in IP routers [29], in association rules and frequent itemsets [1, 68, 73, 44, 42] and databases [30, 9, 41]. Finding the heavy hitters is also frequently used as a subroutine in data stream algorithms for other problems, such as moment estimation [46], entropy estimation [16, 43], ℓ_p-sampling [65], finding duplicates [37], and so on. For surveys on algorithms for heavy hitters, see, e.g., [25, 76].

In the *ϵ-ℓ_p-heavy hitters problem*, for $p \geq 1$, the goal is to find a set S which contains all indices $i \in [n]$ for which $|x_i|^p \geq \epsilon^p \|x\|_p^p$, and contains no indices $i \in [n]$ for which $|x_i|^p \leq \frac{\epsilon^p}{2} \|x\|_p^p$.

The first heavy hitters algorithms were for $p = 1$, given by Misra and Gries [64], who achieved $O(\epsilon^{-1})$ words of memory, where a word consists of $O(\log n)$ bits of space. Interestingly, their algorithm is *deterministic*, i.e., the failure probability $\delta = 0$. This algorithm was rediscovered by Demaine, López-Ortiz, and Munro [27], and again by Karp, Shenker, and Papadimitriou [53]. Other than these algorithms, which are deterministic, there are a number of randomized algorithms, such as the Count-Min sketch [26], sticky sampling [62], lossy counting [62], space-saving [63], sample and hold [29], multi-stage bloom filters [15], and sketch-guided sampling [54]. One can also achieve stronger residual error guarantees [8].

An often much stronger notion than an ℓ_1-heavy hitter is an ℓ_2-heavy hitter. Consider an n-dimensional vector $x = (\sqrt{n}, 1, 1, \ldots, 1)$. The first coordinate is an ℓ_2-heavy hitter with parameter $\epsilon = 1/\sqrt{2}$, but it is only an ℓ_1 heavy hitter with parameter $\epsilon = 1/\sqrt{n}$. Thus, the algorithms above would require at least $\sqrt{n}$ words of memory to find this heavy hitter. In [20] this problem was solved by the CountSketch algorithm, which provides a solution to the ϵ-ℓ_2-heavy hitters problem, and more generally to the ℓ_p-heavy hitters problem for any $p \in (0, 2]$[1], in 1-pass and in the general turnstile model using $O\left(\frac{1}{\epsilon^p} \log(n/\delta)\right)$ words of memory. For insertion-only streams, this was recently improved [13, 12] to $O\left(\frac{1}{\epsilon^p}\right)$ words of memory for constant δ, and $O\left(\frac{1}{\epsilon^p} \log(1/\delta)\right)$ in general. See also work [55] on reducing the decoding time for finding the heavy hitters from the algorithm's memory contents, without sacrificing additional memory.

There is also work establishing lower bounds for heavy hitters. The works of [28, 50] establish an $\Omega\left(\frac{1}{\epsilon^p} \log n\right)$ word lower bound for any value of $p > 0$ and constant δ, for any algorithm in the strict turnstile model. This shows that the above algorithms are optimal for constant δ. Also for $p > 2$, it is known that solving the ϵ-ℓ_p-heavy hitters problem even with constant ϵ and δ requires $\Omega(n^{1-2/p})$ words of memory [5, 39, 49], and thus $p = 2$ is often considered the gold standard for space-efficient streaming algorithms since it is the largest value of p for which there is a $\text{poly}(\log n)$ space algorithm. For deterministic algorithms computing linear sketches, the work of [31] shows the sketch requires $\Omega(n^{2-2/p}/\epsilon^2)$ dimensions for $p \geq 1$ (also shown for $p = 2$ by [23]). This also implies a lower bound for general turnstile algorithms for streams with several important restrictions; see also [57, 51]. There is also work on the related compressed sensing problem which studies small δ [36].

[1] For $p < 1$ the quantity $\|x\|_p$ is not a norm, but it is still a well-defined quantity.

Despite the work above, for all we knew it could be entirely possible that, in the insertion-only model, an ϵ-ℓ_2-heavy hitters algorithm could achieve $O\left(\frac{1}{\epsilon^2}\right)$ words of memory and solve the problem *deterministically*, i.e., with $\delta = 0$. In fact, it is well-known that the above $\Omega(n)$ lower bound for ϵ-ℓ_2-heavy hitters for linear sketches does not hold in the insertion-only model. Indeed, by running a deterministic algorithm for ϵ-ℓ_1-heavy hitters, we have that if $x_i^2 \geq \epsilon^2 \|x\|_2^2$, then $x_i \geq \epsilon \|x\|_2 \geq \frac{\epsilon}{\sqrt{n}} \|x\|_1$, and consequently one can find all ℓ_2-heavy hitters using $O\left(\frac{\sqrt{n}}{\epsilon}\right)$ words of memory. Thus, for constant ϵ, there is a deterministic $O\left(\sqrt{n}\right)$ words of memory upper bound, but only a trivial $\Omega\left(1\right)$ word lower bound. Surprisingly, this factor $\sqrt{n}$ gap was left wide open, and the main question we ask about heavy hitters is:

Can one deterministically solve ϵ-ℓ_2-heavy hitters in insertion-only streams in constant memory?

One approach to solve MostlyDISJ would be for each player to insert their elements into a stream and apply a heavy hitters algorithm. For example, if $k = \sqrt{n}$, there will be a $\Theta(1)$-ℓ_2-heavy hitter if and only if the MostlyDISJ instance is a YES instance. For a space-S streaming algorithm, this uses Sk communication to pass the structure from player to player. Hence $S \gtrsim n/k = \sqrt{n}$. In general:

▶ **Theorem 21.** *Given $\varepsilon \in (\frac{1}{n^{1/p}}, \frac{1}{2})$ and $p \geq 1$, any δ-error r-pass insertion-only streaming algorithm for ε-ℓ_p-heavy hitters requires $\Omega(\min(\frac{n^{1-1/p}}{r\varepsilon}, n^{1-2/p}\frac{\log(1/\delta)}{r\varepsilon^2}))$ bits of space.*

Most notably, setting $\delta = 0$ and $p = 2$ and $r = O(1)$, this gives an $\Omega(\sqrt{n}/\varepsilon)$ bound for deterministic ℓ_2 heavy hitters. The FrequentElements algorithm [64] matches this up to a factor of $\log n$ (i.e., it uses this many words, not bits). For $n^{-.1} > \delta > 0$, the other term $(\frac{\log(1/\delta)}{r\varepsilon^2})$ is also achievable up to the bit/word distinction, this time by CountSketch. For larger δ, we note that it takes $\Omega(\frac{1}{\varepsilon^2}\log \varepsilon^2 n)$ bits already to encode the output size. As a result, we show that the existing algorithms are within a $\log n$ factor of optimal.

One common motivation for heavy hitters is that many distributions are power-law or Zipfian distributions. For such distributions, the i-th most frequent element has frequency approximately proportional to $i^{-\zeta}$ for some constant ζ, typically $\zeta \in (0.5, 1)$ [22]. Such distributions have significant $\ell_{1/\zeta}$-heavy hitters. Despite our lower bound for general heavy hitters, one might hope for more efficient deterministic/very high probability insertion-only algorithms in this special case. We rule this out as well, getting an $\Omega(\min(n^{1-\zeta}, n^{1-2\zeta}\log(1/\delta)))$ lower bound for finding the heavy hitters of these distributions (see Theorem 24). This again matches the upper bounds from FrequentElements or CountSketch up to a logarithmic factor.

To extend our lower bound to power-law distributions, we embed our hard instance as the single largest and $n/2$ smallest entries of a power-law distribution; we then insert the rest of the power-law distribution deterministically, so the overall distribution is power-law distributed. Solving heavy hitters will identify whether this single largest element exists or not, solving the communication problem.

Frequency Moments

We next turn to the problem of estimating the frequency moments F_p, which in our reduction from the MostlyDISJ problem, just corresponds to estimating $\|x\|_p^p = \sum_{i=1}^n |x_i|^p$. Our hard instance for MostlyDISJ immediately gives us the following theorem:

▶ **Theorem 2.** *For any constant $\epsilon \in (0,1)$ and $p \geq 2$, any δ-error r-pass insertion-only streaming algorithm for ε-F_p-estimation must have space complexity of $\Omega(\min(\frac{n^{1-1/p}}{r}, n^{1-2/p}\frac{\log(1/\delta)}{r}))$ bits.*

The proof of Theorem 2 follows immediately by setting the number of players in MostlyDISJ to be $\Theta((\epsilon n)^{1/p})$, and performing the reduction to F_p-estimation described before Section 1.1. This improves the previous $\Omega((n^{1-2/p} + \log(1/\delta))/r)$ lower bound, which follows from [5, 49], as well as a simple reduction from the Equality function [3], see also [17]. It matches an upper bound of [14] for constant ϵ, by repeating their algorithm independently $O(\log(1/\delta))$ times. Our lower bound instance shows that to approximate $\|x\|_\infty = \max_i |x_i|$ of an integer vector, with $O(\log n)$-bit coordinates in n dimensions, up to an additive $\Theta(\sqrt{\|x\|_2})$ deterministically, one needs $\Omega(\sqrt{n})$ memory. This follows from our hard instance. Approximating the ℓ_∞ norm is an important problem in streaming, and its complexity was asked about in Question 3 of [24].

Low Rank Approximation

Our ℓ_2-heavy hitters lower bound also has applications to *deterministic* low rank approximation in a stream, a topic of recent interest [59, 35, 75, 34, 33, 45]. Here we see rows $A_1, A_2, \ldots, A_n$ of an $n \times d$ matrix A one at a time. At the end of the stream we should output a projection P onto a rank-k space for which $\|A - AP\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$, where A_k is the best rank-k approximation to A. A natural question is if the deterministic FrequentDirections algorithm of [34] using $O(dk/\epsilon)$ words of memory can be improved when the rows of A are $O(1)$-sparse. The sparse setting was shown to have faster running times in [33, 45], and more efficient randomized communication protocols in [10]. Via a reduction from our MostlyDISJ problem, we show a polynomial dependence on d is necessary.

▶ **Theorem 25.** *Any 1-pass deterministic streaming algorithm outputting a rank-k projection matrix P providing a $(1+\epsilon)$-approximate rank-k low rank approximation requires $\Omega(\sqrt{d})$ bits of memory, even for $k = 1$, $\epsilon = \Theta(1)$, and when each row of A has only a single non-zero entry.*

Algorithms and Lower Bounds in Other Streaming Models

We saw above that deterministic insertion-only ℓ_2 heavy hitters requires $\widetilde{\Theta}(\sqrt{n})$ space for constant ε. We now consider turnstile streaming and linear sketching.

The work of [31, 23] shows that $\Omega(n)$ space is needed for general deterministic linear sketching, but the corresponding hard instances have negative entries. We extend this in two ways: when negative entries are allowed, an $\Omega(n)$ lower bound is easy even in turnstile streaming (for heavy hitters, but not the closely related ℓ_∞/ℓ_2 sparse recovery guarantee; see Remark 27). If negative entries are not allowed, we still get an $\Omega(n)$ bound on the number of linear measurements for deterministic linear sketching (see Theorem 20).

A question is if we can solve ℓ_2 heavy hitters deterministically in the strict turnstile model in $o(n)$ space. In some sense the answer is no, due to the near equivalence between turnstile streaming and linear sketching [31, 58, 2], but this equivalence has significant limitations. Recent work has shown that with relatively mild restrictions on the stream, such as a bound on the length L, significant improvements over linear sketching are possible [47, 51]. Can we get that here? We show that this is indeed possible: streams with $O(n)$ updates can be solved in $O(n^{2/3})$ space. While this does not reach the $\sqrt{n}$ lower bound from insertion-only streams (Theorem 22), it is significantly better than the $\Omega(n)$ for linear sketches. In general, we show:

▶ **Theorem 26.** *There is a deterministic ℓ_2 heavy hitters algorithm for length-L strict turnstile streams with ± 1 updates using $O((L/\varepsilon)^{2/3})$ words of space.*

Our algorithm for short strict turnstile streams is a combination of FrequentElements and exact sparse recovery. With space S, FrequentElements (modified to handle negative updates) gives estimation error L/S, which is good unless $\|x\|_2 \ll L/S$. But if it is not good, then $\|x\|_0 \leq \|x\|_2^2 \ll (L/S)^2$. Hence in that case $(L/S)^2$-sparse recovery will recover the vector (and hence the heavy hitters). Running both algorithms and combining the results takes $S + (L/S)^2$ space, which is optimized at $L^{2/3}$.

1.2 Our Techniques

Our key lemma is that solving MostlyDISJ on n elements, k items, and $l = ck$ with probability $1 - e^{-k}$ has $\Omega(n)$ conditional information complexity for any constant $c \in (0, 1)$. It is well-known that the conditional information complexity of a problem lower bounds its communication complexity (see, e.g., [5]).

This can then be extended to $\delta \gg e^{-\Theta(k)}$ using repetition, namely, we can amplify the success probability of the protocol to $1 - e^{-\Theta(k)}$ by independent repetition, apply our $\Omega(n)$ lower bound on the new protocol with $\delta = e^{-\Theta(k)}$, and then conclude a lower bound on the original protocol. Indeed, this is how we obtain our total communication lower bound of $\Omega(n/k)$ for constant δ, providing a much simpler proof than that of the $\Omega(n/k)$ total communication lower bound for promise set disjointness in prior work.

Our bound is tight up to a $\log k$ factor. It can be solved deterministically with $O(n \log k)$ communication (for each bit, the first player with that bit publishes it), and with probability $1 - (1 - \varepsilon)^{l-1}$ using $O(\varepsilon n \log k)$ communication (only publish the bit with probability ε). Setting $\varepsilon = o(1)$, any $e^{-o(k)}$ failure probability is possible with $o(n \log k)$ communication.

We lower bound MostlyDISJ using conditional information complexity. Using the direct sum property of conditional information cost, analogous to previous work (see, e.g., [5]), it suffices to get an $\Omega(1)$ conditional information cost bound for the $n = 1$ problem F_k: we have k players, each of whom receives one bit, and the players must distinguish (with probability $1 - e^{-k}$) between at most one player having a 1, and at least $\Omega(k)$ players having 1s. In particular, it suffices to show for correct protocols π that

$$\mathop{\mathbb{E}}_{i \in [t]} d_{\mathrm{TV}}(\pi_0, \pi_{e_i}) = \Omega(1) \tag{1}$$

where π_0 is the distribution of protocol transcripts if the players all receive 0, and π_{e_i} is the distribution if player i receives a 1. The main challenge is therefore in bounding this expression.

Consider any protocol that does not satisfy (1). We show that, when $d_{\mathrm{TV}}(\pi_0, \pi_{e_i}) \ll 1$, player i can be implemented with an equivalent protocol for which the player usually does not even observe its input bit. That is, if every other player receives a 0, player i will only observe its bit with probability $d_{\mathrm{TV}}(\pi_0, \pi_{e_i})$. This means that most players only have a small probability of observing their bit. The probability that any two players i, i' observe their bits may be correlated; still, we show that this implies the existence of a large set S of ck players such that the probability – if every player receives a zero – that *no* player $i \in S$ observes their bit throughout the protocol is above e^{-k}. But then $d_{\mathrm{TV}}(\pi_0, \pi_{e_S}) < 1 - e^{-k}$, so the protocol cannot distinguish these cases with the desired probability. We now give the full proof.

2 Preliminaries

We use the following measures of distance between distributions in our proofs.

▶ **Definition 3.** *Let P and Q be probability distributions over the same countable universe $\mathcal{U}$. The total variation distance between P and Q is defined as: $d_{TV}(P,Q) = \frac{1}{2}\left\|P-Q\right\|_1$.*

In our proof we also use the Jensen-Shannon divergence and Kullback-Liebler divergence. We define these notions of divergence here:

▶ **Definition 4.** *Let P and Q be probability distributions over the same discrete universe $\mathcal{U}$. The Kullback-Liebler divergence or KL-divergence from Q to P is defined as: $D_{KL}(P,Q) = \sum_{x\in\mathcal{U}} P(x)\log(\frac{P(x)}{Q(x)})$. This is an asymmetric notion of divergence. The Jensen-Shannon divergence between two distributions P and Q is the symmetrized version of the KL divergence, defined as: $D_{JS}(P,Q) = \frac{1}{2}(D_{KL}(P,Q) + D_{KL}(Q,P))$.*

From Pinsker's inequality, for any two distributions P and Q, $D_{KL}(P,Q) \geq \frac{1}{2}d^2_{\mathrm{TV}}(P,Q)$.

In the multiparty communication model we consider k-ary functions $F : \mathcal{L} \to \mathcal{Z}$ where $\mathcal{L} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_k$. There are k parties(or players) who receive inputs $X_1,\ldots,X_k$ which are jointly distributed according to some distribution μ. We consider protocols in the blackboard model where in any protocol π players speak in any order and each player broadcasts their message to all other players. So, the message of player i is a function of the messages they receive, their input and randomness i.e., $m_i = M_i(X_i, m_{i-1}, R_i)$. The final player's message is the output of the protocol.

The communication cost of a multiparty protocol π is the sum of the lengths of the individual messages $\|\pi\| = \sum|M_j|$. A protocol π is a δ-error protocol for the function f if for every input $x \in \mathcal{L}$, the output of the protocol equals $f(x)$ with probability $1-\delta$. The randomized communication complexity of f, denoted $R_\delta(f)$, is the cost of the cheapest randomized protocol that computes f correctly on every input with error at most δ over the randomness of the protocol.

The distributional communication complexity of the function f for error parameter δ is denoted as $D^\delta_\mu(f)$. This is the communication cost of the cheapest deterministic protocol which computes the function f with error at most δ under the input distribution μ. By Yao's minimax theorem, $R_\delta(f) = \max_\mu D^\delta_\mu(f)$ and hence it suffices to prove a lower bound for a hard distribution μ. In our proofs, we bound the conditional information complexity of a function in order to prove lower bounds on $R_\delta(f)$. We define this notion below.

▶ **Definition 5.** *Let π be a randomized protocol whose inputs belong to $\mathcal{K} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \ldots \times \mathcal{X}_k$. Suppose $((X_1,X_2,\ldots,X_k),D) \sim \eta$ where η is a distribution over $\mathcal{K}\times\mathcal{D}$ for some set $\mathcal{D}$. The **conditional information cost** of π with respect to η is defined as: $cCost_\eta(\pi) = I(X_1,\ldots,X_k;\pi(X_1,\ldots,X_k) \mid D)$.*

▶ **Definition 6.** *The δ-error **conditional information complexity** of f with respect to η, denoted $CIC_{\eta,\delta}(f)$ is defined as the minimum conditional information cost of a δ-error protocol for f with respect to η.*

In [5] it was shown that the randomized communication complexity of a function is at least the conditional information complexity of the function f with respect to any input distribution η.

▶ **Proposition 7** (Corollary 4.7 of [5]). *Let $f : \mathcal{K} \to \{0,1\}$, and let η be a distribution over $\mathcal{K}\times\mathcal{D}$ for some set $\mathcal{D}$. Then, $R_\delta(f) \geq CIC_{\eta,\delta}(f)$.*

Direct Sum

Per [5], conditional information complexity obeys a Direct Sum Theorem condition under various conditions. The Direct Sum Theorem of [5] allows us to reduce a t-player conditional information complexity problem with an n-dimensional input to each player to a t-player conditional information complexity with a 1-dimensional input to each player. This theorem applies when the function is "decomposable" and the input distribution is "collapsing". We define both these notions here.

▶ **Definition 8.** *Suppose $\mathcal{L} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_t$ and $\mathcal{L}_n \subseteq \mathcal{L}^n$. A function $f : \mathcal{L}_n \to \{0,1\}$ is g-**decomposable** with primitive $h : \mathcal{L} \to \{0,1\}$ if it can be written as:*

$$f(X_1, \ldots, X_t) = g(h(X_{1,1}, \ldots, X_{1,t}), \ldots, h(X_{n,1}, \ldots, X_{n,t}))$$

for $g : \{0,1\}^n \to \{0,1\}$.

▶ **Definition 9.** *Suppose $\mathcal{L} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_t$ and $\mathcal{L}_n \subseteq \mathcal{L}^n$. A distribution η over $\mathcal{L}_n$ is a **collapsing distribution** for $f : \mathcal{L}_n \to \{0,1\}$ with respect to $h : \mathcal{L} \to \{0,1\}$ if for all $Y_1, \ldots, Y_n$ in the support of η, for all $y \in \mathcal{L}$ and for all $i \in [n]$, $f(Y_1, \ldots, Y_{i-1}, y, Y_{i+1}, \ldots, Y_n) = h(y)$.*

We state the Direct Sum Theorem for conditional information complexity below. The proof of this theorem in [5] applies to the blackboard model of multiparty communication. We state this in the most general form here and then show that it may be applied to the hard distribution η_0 which we choose in Section 3.

▶ **Theorem 10** (Multiparty version of Theorem 5.6 of [5]). *Let $\mathcal{L} \subseteq \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \mathcal{X}_t$ and let $\mathcal{L}_n \subseteq \mathcal{L}^n$. Suppose that the following conditions hold:*

(i) *$f : \mathcal{L}_n \to \{0,1\}$ is a decomposable function with primitive $h : \mathcal{L} \to \{0,1\}$,*

(ii) *ζ is a distribution over $\mathcal{L} \times \mathcal{D}$, such that for any $d \in \mathcal{D}$ the distribution $(\zeta \mid D = d)$ is a product distribution,*

(iii) *$\eta = \zeta^n$ is supported on $\mathcal{L}_n \times \mathcal{D}^n$, and*

(iv) *the marginal probability distribution of η over $\mathcal{L}_n$ is a collapsing distribution for f with respect to h.*

Then $CIC_{\eta,\delta}(f) \geq n \cdot CIC_{\zeta,\delta}(h)$.

3 Communication Lower Bound for Mostly Set Disjointness

Let $[n] = \{1, 2, \ldots, n\}$. We let $H(X)$ denote the entropy of a random variable X, and $I(X; Y) = H(X) - H(X|Y)$ be the mutual information.

3.1 The Hard Distribution

▶ **Definition 11.** *Denote by* MostlyDISJ$_{n,l,t}$, *the multiparty Mostly Set-Disjointness problem in which each player $j \in [t]$ receives an n-dimensional input vector $X_j = (X_{j,1}, \ldots, X_{j,n})$ where $X_{j,i} \in \{0,1\}$ and the input to the protocol falls into either of the following cases:*

- ***NO:** For all $i \in [n]$, $\sum_{j\in[t]} X_{j,i} \leq 1$*
- ***YES:** There exists a unique $i \in [n]$ such that $\sum_{j\in[t]} X_{j,i} = l$ and for all other $i' \neq i, \sum_{j\in[t]} X_{j,i'} \leq 1$.*

The final player must output 1 if the input is in the YES case and 0 in the NO case.

Let $\mathcal{L} \subset \{0,1\}^t$ be the set of valid inputs along one index in $[n]$ for $\mathsf{MostlyDISJ}_{n,l,t}$, i.e., the set of elements in $x \in \{0,1\}^t$ with $\sum_{j\in[t]} x_j \le 1$ or $\sum_{j\in[t]} x_j = l$. Let $\mathcal{L}_n \subset \mathcal{L}^n$ denote the set of valid inputs to the $\mathsf{MostlyDISJ}_{n,l,t}$ function.

Then $\mathsf{MostlyDISJ}_{n,l,t} : \mathcal{L}_n \to \{0,1\}$ is defined as: $\mathsf{MostlyDISJ}_{n,l,t}(X_1,\ldots,X_t) = \bigvee_{i\in[n]} F_{l,t}(X_{1,i},\ldots,X_{t,i})$ for the function $F_{l,t} : \mathcal{L} \to \{0,1\}$ defined as: $F_{l,t}(x_1,\ldots,x_t) = \bigvee_{\substack{S\subseteq[t]\\|S|=l}} \bigwedge_{j\in S} x_j$. This means that $\mathsf{MostlyDISJ}_{n,l,t}$ is OR-decomposable into n copies of $F_{l,t}$ and we may hope to apply a direct sum theorem with an appropriate distribution over the inputs.

In order to prove a lower bound on the conditional information complexity, we need to define a "hard" distribution over the inputs to $\mathsf{MostlyDISJ}_{n,l,t}$. We define the distribution η over $\mathcal{L}_n \times \mathcal{D}^n$ where $\mathcal{D} = [t]$ as follows:

- For each $i \in [n]$ pick $D_i \in [t]$ uniformly at random and sample $X_{D_i,i}$ uniformly from $\{0,1\}$ and for all $j' \neq D_i$ set $X_{j',i} = 0$.
- Pick $I \in [n]$ uniformly at random and $Z \in \{0,1\}$
- if $Z = 1$, pick a set $S \subseteq [t]$ such that $|S| = l$ uniformly at random and for all $j \in S$ set $X_{j,I} = 1$ and for all $j \notin S$, set $X_{j,I} = 0$

Let μ_0 denote the distribution for each $i \in [n]$ conditioned on $Z = 0$. For any $d \in [t]$, when $D = d$, the conditional distribution over $\mathcal{L}$ is the uniform distribution over $\{0, e_d\}$ and hence a product distribution. Let η_0 be the distribution η conditioned on $Z = 0$. Clearly, $\eta_0 = \mu_0^n$.

This definition of $\mathsf{MostlyDISJ}_{n,l,t}$ and the hard distribution η_0 allows us to apply the Direct Sum theorem (Theorem 10) of [5]. Note that: (i) $\mathsf{MostlyDISJ}_{n,l,t}$ is OR-decomposable by $F_{l,t}$, (ii) μ_0 is a distribution over $\mathcal{L} \times [t]$ such that the marginal distribution $(\mu_0 \mid D = d)$ over $\mathcal{L}$ is uniform over $\{0, e_d\}$(and hence a product distribution), (iii) $\eta_0 = \mu_0^n$, and (iv) since $\mathsf{MostlyDISJ}_{n,l,t}$ is OR-decomposable and η_0 has support only on inputs in the NO case, η_0 is a collapsing distribution for $\mathsf{MostlyDISJ}_{n,l,t}$ with respect to $F_{l,t}$. Hence:

$$CIC_{\eta_0,\delta}(\mathsf{MostlyDISJ}_{n,l,t}) \ge n \cdot CIC_{\mu_0,\delta}(F_{l,t}) \tag{2}$$

3.2 Information Cost for a Single Bit

A key lemma for our argument is that the players can be implemented so that they only "observe" their input bits with small probability. The model here is that each player's input starts out hidden, but they can at any time choose to observe their input. Before they observe their input, however, all their decisions (including messages sent and choice of whether to observe) depend on the transcript and randomness, but not the player's input.

In this section we use π to denote the protocol in consideration and abuse notation slightly by using π_x to denote the distribution of the transcript of the protocol π on input x.

▶ **Definition 12.** *Any (possibly multi-round) communication protocol involving n players, where each player receives one input bit, is defined to be a "clean" protocol with respect to player i if, in each round,*

1. *if player i has previously not "observed" his input bit, he "observes" his input bit with some probability that is a function only of the previous messages in the protocol,*
2. *if player i has not observed his input bit in this round or any previous round, then his message distribution depends only on the previous messages in the protocol but not his input bit, and*

3. *if player i* has *observed his input bit in this round or any previous round, then – for a fixed value of the previous messages in the protocol – his distribution of messages on input* 0 *and on input* 1 *are* disjoint.

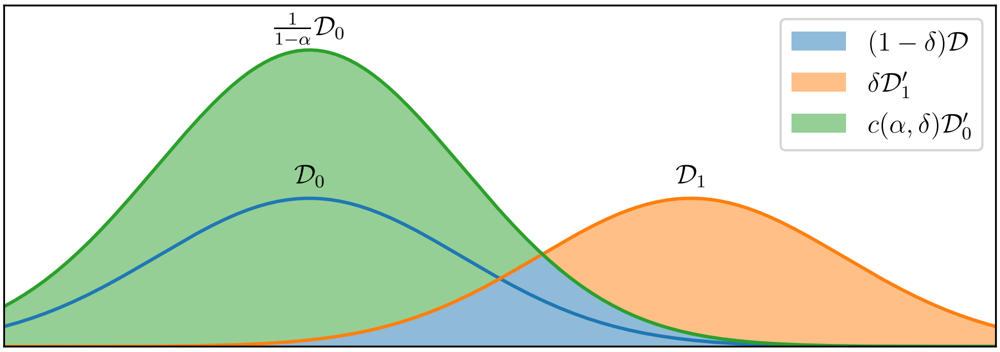


Figure 1 An illustration of Lemma 13, given a parameter α and pair of distributions $(\mathcal{D}_0, \mathcal{D}_1)$. We set $(1-\delta)\mathcal{D}$ to be the overlap between $\mathcal{D}_1$ and $\frac{1}{1-\alpha}\mathcal{D}_0$, then $\mathcal{D}'_0$ and $\mathcal{D}'_1$ to be proportional to the remainder of $\frac{1}{1-\alpha}\mathcal{D}_0$ and $\mathcal{D}_1$, respectively. These $\mathcal{D}'_0$ and $\mathcal{D}'_1$ are disjoint.

We start off by proving a lemma about decomposing any two arbitrary distributions into one "common" distribution and two disjoint different distributions. This lemma will enable us to show that any communication protocol can be simulated in a clean manner.

▶ **Lemma 13.** *Let $\mathcal{D}_0, \mathcal{D}_1$ be two distributions, and $\alpha \in [0,1]$. There exist three distributions $\mathcal{D}, \mathcal{D}'_0, \mathcal{D}'_1$ and a parameter $\delta \in (0,1)$ such that: $\mathcal{D}_0 = (1-\alpha)(1-\delta)\mathcal{D} + (1-(1-\alpha)(1-\delta))\mathcal{D}'_0$, $\mathcal{D}_1 = (1-\delta)\mathcal{D} + \delta\mathcal{D}'_1$, and $\mathcal{D}'_0$ has a disjoint support from $\mathcal{D}'_1$.*

We refer the reader to Figure 1 for an illustration corresponding to Lemma 13.

Proof. We begin with two special cases. If $\alpha = 1$, then setting $\delta = 0$ allows us to set $\mathcal{D}'_0 = \mathcal{D}_0$, $\mathcal{D} = \mathcal{D}_1$. $\mathcal{D}'_1$ may be any arbitrary distribution that has disjoint support from $\mathcal{D}'_0$. If $\operatorname{supp}(\mathcal{D}_0) \cap \operatorname{supp}(\mathcal{D}_1) = \varnothing$, we may set $\delta = 1$, $\mathcal{D}'_0 = \mathcal{D}_0$ and $\mathcal{D}'_1 = \mathcal{D}_1$.

So it suffices to consider the case where $\alpha < 1$ and $\operatorname{supp}(\mathcal{D}_0) \cap \operatorname{supp}(\mathcal{D}_1) \neq \varnothing$. Let $\mathcal{D}$ and δ be such that $\mathcal{D}(x) = \frac{1}{1-\delta}\min(\frac{1}{1-\alpha}\mathcal{D}_0(x), \mathcal{D}_1(x))$ is a distribution over the support of $\mathcal{D}_0$. Then, it suffices to define:

$$\mathcal{D}'_0(x) = \begin{cases} 0 & \text{if } \frac{1}{1-\alpha}\mathcal{D}_0(x) \leq \mathcal{D}_1(x) \\ \frac{1}{1-(1-\alpha)(1-\delta)}(\mathcal{D}_0(x) - (1-\alpha)\mathcal{D}_1(x)) & \text{otherwise} \end{cases}$$

and we define:

$$\mathcal{D}'_1(x) = \begin{cases} 0 & \text{if } \frac{1}{1-\alpha}\mathcal{D}_0(x) \geq \mathcal{D}_1(x) \\ \frac{1}{\delta}(\mathcal{D}_1(x) - \frac{\mathcal{D}_0(x)}{1-\alpha}) & \text{otherwise} \end{cases}$$

If $\alpha = 1$, we set $\mathcal{D}'_0 = \mathcal{D}_0$, $\mathcal{D}(x) = \frac{1}{1-\delta}\min(\mathcal{D}_1(x), \mathcal{D}_0(x))$ where δ is a scaling term which ensures that $\mathcal{D}(x)$ is a valid distribution. ◀

▶ **Lemma 14.** *Consider any (possibly multi-round) communication protocol π where each player receives one input bit. Then for any player i, the protocol can simulated in a manner that is "clean" with respect to that player.*

Proof. Let b denote player i's bit. We use "round r" to refer to the rth time that player i is asked to speak. Let m_r be the transcript of the protocol just before player i speaks in round r, and let m_r^+ denote the transcript immediately after player i speaks in round r. Let $\mathcal{D}_{m_r}^b$ be the distribution of player i's message the rth time he is asked to speak, conditioned on the transcript so far being m_r and on player i having the bit b. We will describe an implementation of player i that produces outputs with the correct distribution $\mathcal{D}_{m_r}^b$ such that the implementation only looks at b with relatively small probability.

In the first round, given m_1, player i looks at b with probability $d_{\mathrm{TV}}(\mathcal{D}_{m_1}^0, \mathcal{D}_{m_1}^1)$. If he does not look at the bit, he outputs each message m with probability proportional to $\min(\mathcal{D}_{m_1}^0(m), \mathcal{D}_{m_1}^1(m))$; if he sees the bit b, he outputs each message m with probability proportional to $\max(0, \mathcal{D}_{m_1}^b(m) - \mathcal{D}_{m_1}^{1-b}(m))$. His output is then distributed according to $\mathcal{D}_{m_r}^b$. Note also that, for any message m, it is not possible that the player can send m both after reading a 0 and after reading a 1.

In subsequent rounds r, given m_r, player i needs to output a message with distribution $\mathcal{D}_{m_r}^b$. Let p_0 denote the probability that the player has already observed his bit in a previous round, conditioned on m_r and $b = 0$; let p_1 be analogous for $b = 1$. We will show by induction that $\min(p_0, p_1) = 0$ for all m_r. That is, any given transcript may be compatible with having already observed a 0 or a 1 but not both. As noted above, this is true for $r = 2$.

Without loss of generality, suppose $p_1 = 0$. We apply Lemma 13 to $\mathcal{D}_{m_r}^0$ and $\mathcal{D}_{m_r}^1$ with $\alpha = p_0$, obtaining three distributions $(\mathcal{D}, \mathcal{D}_0, \mathcal{D}_1)$ such that $\mathcal{D}_{m_r}^0 = (1 - p_0)(1 - \delta)\mathcal{D} + (1 - (1 - p_0)(1 - \delta))\mathcal{D}_0$ and $\mathcal{D}_{m_r}^1 = (1 - \delta)\mathcal{D} + \delta\mathcal{D}_1$, and $\mathcal{D}_0$ is disjoint from $\mathcal{D}_1$.

Player i behaves as follows: if he has not observed his bit already, he does so with probability δ. After this, if he still has not observed his bit, he outputs a message according to $\mathcal{D}$; if he has observed his bit b, he outputs according to $\mathcal{D}_b$.

The resulting distribution is $\mathcal{D}_{m_r}^b$ regardless of b, and the set of possible transcripts where a 1 has been observed is disjoint from those possible where a 0 has been observed. By induction, this holds for all rounds r. Thus, this is a simulation of the original protocol that is "clean" with respect to player i. ◀

▶ **Lemma 15.** *Consider any (possibly multiround) communication protocol π where each player receives one bit. Each player i can be implemented such that, if every other player receives a 0 input, player i only observes his input with probability $d_{TV}(\pi_{e_i}, \pi_0)$.*

Proof. Using Lemma 14, we know that player i can be implemented such that the protocol is clean with respect to that player.

We may now analyze the probability p^* that player i ever observes his bit, assuming that all other players receive the input zero. For every possible transcript m let $p_0(m)$ denote the probability, conditioned on the transcript being m and player i's bit being 0, that player i observes his bit at any point during the protocol; let $p_1(m)$ be analogous for the bit being 1. Because the choice of player i to observe his input bit in a clean protocol is independent of the bit, we have that $p^* = \sum_m \Pr_{\pi_0}[m]p_0(m) = \sum_m \Pr_{\pi_{e_i}}[m]p_1(m)$. Moreover, because the protocol is independent of the bit if it is not observed,

$$(1 - p_0(m)) \Pr_{\pi_0}[m] = (1 - p_1(m)) \Pr_{\pi_{e_i}}[m]$$

for all m. By the definition of a clean protocol, the last message player i sends can be consistent with him observing a 0 or a 1 but not both; therefore $p_0(m) = 0$ or $p_1(m) = 0$ for all m. Now, define $S := \{m \mid p_0(m) > 0\} = \{m \mid \Pr_{\pi_0}(m) > \Pr_{\pi_{e_i}}(m)\}$. Therefore

$$d_{\mathrm{TV}}(\pi_0, \pi_{e_i}) = \sum_{m \in S} \Pr_{\pi_0}[m] - \Pr_{\pi_{e_i}}[m] = \sum_{m \in S} p_0(m) \Pr_{\pi_0}[m] = p^*$$

as desired. ◀

Lemma 15 will be used to show that each player has a decent chance of not reading their input. But to get a lower bound for MostlyDISJ, we need a large *set* of players that have a nontrivial chance of all ignoring their input at the same time. We show the existence of such a set, despite the players not being independent. For any $c \in (0,1)$, define

$$\gamma_c := \frac{1}{c\log(e/c)} \tag{3}$$

We have

▶ **Lemma 16.** *Let $c \in (0,1)$, $p \in (0, \frac{1-c}{2})$, and γ_c as in (3). For a set of 0-1 random variables $Y_1, \dots, Y_k$ such that $\mathbb{E}[\sum_i Y_i] = pk$, there exists $S \subset \{1, 2, \dots, n\}$ of size ck such that $\Pr[\forall j \in S, Y_j = 0] > e^{-k/\gamma_c - 1}$.*

Proof. We wish to show that there exists a set S such that $Y_i = 0$ for all $i \in S$ with nontrivial probability. Observe that if S were chosen uniformly at random,

$$\mathop{\mathbb{E}}_{S:|S|=ck} \Pr[\forall j \in S, Y_j = 0] \geq \frac{1}{\binom{k}{ck}} \Pr[wt(Y) \leq k - ck] \geq \left(\frac{c}{e}\right)^{ck} \cdot \left(1 - \frac{p}{1-c}\right) \geq e^{-1-kc\log(e/c)}.$$

where the first inequality considers the existence of such a set, the second inequality uses $\binom{a}{b} \leq (\frac{e \cdot a}{b})^b$ and Markov's inequality, and $wt(Y)$ denotes the Hamming weight of Y, i.e., number of non-zero entries of the vector Y. Therefore there exists a set S of size $\Omega(ck)$ such that $\Pr[Y_S = 0] \geq e^{-1-kc\log(e/c)}$. ◀

We can now bound the 1-bit communication cost of our problem.

▶ **Lemma 17.** *Given $0 < \delta, c < 1$, γ_c as in (3), and $k \leq \gamma_c \log(\frac{1}{2e\delta})$, for any δ-error protocol for $F_{ck,k}$ we have that $cCost_{\mu_0,\delta}(\pi) = \Omega((1-c)^2)$.*

Proof. Let π be a protocol for $F_{ck,k}$. Let π_x is the distribution of the transcript of the protocol on input x. We start by establishing a connection between conditional information cost and total variation distances. First observe that due to the choice of distribution μ_0, we may write the conditional mutual information as:

$$cCost_{\mu_0,\delta}(\pi) = I(\pi(X_1, \dots, X_k); X_1, \dots, X_k \mid D) = \mathop{\mathbb{E}}_{i \in [k]}[I(X_i; \pi_{0,0,0,\dots X_i,\dots 0,0,0})].$$

Since X_i is uniformly picked from $\{0,1\}$, this mutual information is a Jensen-Shannon divergence (see, for example, Wikipedia [74] or Proposition A.6 of [5]):

$$I(X_i; \pi_{0,0,0,\dots X_i,\dots 0,0,0}) = D_{JS}(\pi_0, \pi_{e_i}) = \frac{1}{2}\left(D_{KL}(\pi_0, \frac{1}{2}(\pi_0 + \pi_{e_i})) + D_{KL}(\pi_{e_i}, \frac{1}{2}(\pi_0 + \pi_{e_i}))\right)$$

From Pinsker's inequality, $D_{KL}(P,Q) \geq \frac{1}{2} d^2_{\mathrm{TV}}(P,Q)$, so:

$$cCost_{\mu_0,\delta}(\pi) \geq \frac{1}{4} \mathop{\mathbb{E}}_{i \in [k]}[d^2_{\mathrm{TV}}(\pi_0, \frac{1}{2}(\pi_0 + \pi_{e_i})) + d^2_{\mathrm{TV}}(\frac{1}{2}(\pi_0 + \pi_{e_i}), \pi_{e_i})] = \frac{1}{8} \mathop{\mathbb{E}}_{i \in [k]}[d^2_{\mathrm{TV}}(\pi_0, \pi_{e_i})]. \tag{4}$$

This is similar to the connection established in Lemma 6.2 of [5] between conditional information cost and squared Hellinger distance (it is weaker but simpler to show).

Suppose, for the sake of contradiction, that $\sum_i d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) = kp$ where $p < \frac{1-c}{2}$. Suppose for each player $i \in [k]$, that $d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) = p_i$. By Lemma 15, this implies that each player in the protocol can be equivalently implemented in a manner such that – if everyone else receives a 0 – player i only looks at their input with probability p_i. If a player does not look at his bit, it means the player's messages are independent of his input. Let Y_i denote the indicator random variable for the event that player i looks at his input in this equivalent protocol.

For the input $X = 0$, we have $\mathbb{E}[\sum_i Y_i] = \sum p_i = kp$. Observe, that for any set S, if $Y_i = 0$ for all $i \in S$, the players do not see their input. So if E_S denotes the event that $\forall i \in S, Y_i = 0$, then

$$d_{\mathrm{TV}}(\pi_{e_S}, \pi_0) = \Pr[E_S] \cdot d_{\mathrm{TV}}(\pi_{e_S} \mid E_S, \pi_0 \mid E_S) + \Pr[\overline{E_S}] d_{\mathrm{TV}}(\pi_{e_S} \mid \overline{E_S}, \pi_0 \mid \overline{E_S}) \leq \Pr[\overline{E_S}]$$

Since $\mathbb{E}[\sum_i Y_i] = kp$ for $p < \frac{1-c}{2}$, this means by Lemma 16 that there exists a set S with $|S| = ck$ such that $\Pr[E_S] \geq e^{-k/\gamma_c - 1}$. Since $k \leq \gamma_c \log(\frac{1}{2e\delta})$, we have $\Pr[E_S] > 2\delta$. For this S, we have that $d_{\mathrm{TV}}(\pi_{e_S}, \pi_0) < 1 - 2\delta$ and this means that the protocol errs with probability $> \delta$. This is a contradiction. So, we must have $\sum_i d_{\mathrm{TV}}(\pi_{e_i}, \pi_0) > \frac{1-c}{2} k$. By (4) and Jensen's inequality, this gives

$$cCost_{\mu_0,\delta}(\pi) \geq \frac{1}{8} \mathop{\mathbb{E}}_{i \in [k]} [d^2_{\mathrm{TV}}(\pi_{e_i}, \pi_0)] \geq \frac{1}{8} \mathop{\mathbb{E}}_{i \in [k]} [d_{\mathrm{TV}}(\pi_{e_i}, \pi_0)]^2 \geq \frac{(1-c)^2}{32}. \qquad ◀$$

3.3 Finishing it Off

We prove a lower bound on the randomized communication complexity of MostlyDISJ.

▶ **Theorem 18.** *Given $0 < \delta, c < 1$ and $k \leq \gamma_c \log(\frac{1}{2e\delta})$ for γ_c as in (3),*

$$R_\delta(\mathsf{MostlyDISJ}_{n,ck,k}) = \Omega((1-c)^2 n).$$

To prove this, it suffices to prove Lemma 17 where we show a lower bound on the conditional information cost of δ-error protocols for $F_{ck,k}$. This implies a lower bound on the conditional information complexity of $F_{ck,k}$ which together with (2) implies the desired result.

Proof. Combining Proposition 7, Equation (2), and Lemma 17 gives:

$$R_\delta(\mathsf{MostlyDISJ}_{n,ck,k}) \geq CIC_{\eta_0,\delta}(\mathsf{MostlyDISJ}_{n,ck,k}) \geq n \cdot CIC_{\mu_0,\delta}(F_{ck,k}) \gtrsim n(1-c)^2$$

as desired. ◀

In the Lemma 17 we showed that for any protocol for $F_{ck,k}$ with input drawn from μ_0, if the conditional information cost is $o(1)$, there exists an input on which it errs with probability $> \delta$. This implies a lower bound on the conditional information complexity of $F_{ck,k}$.

For algorithms that have large error probability, the success probability can be amplified by using independent copies of the algorithm and taking the majority vote. We use this observation to obtain a lower bound for algorithms with error probability larger than e^{-k}.

▶ **Theorem 1.** MostlyDISJ *with n elements, k players, and $l = ck$ for an absolute constant $c \in (0, 1)$ requires $\Omega(\min(n, n\frac{\log(1/\delta)}{k}))$ bits of communication for failure probability δ.*

Proof. For the absolute constant γ_c, when $k < \gamma_c \log(1/\delta)$ (or $\delta < e^{-k/\gamma_c}$), Theorem 18 gives us a lower bound of $\Omega(n)$. Now, consider the case where $\delta > e^{-k/\gamma_c}$. Suppose π is a protocol whose communication cost is C. Then, we may amplify the success probability of this protocol. We create a new protocol π' which runs r independent copies of π in parallel and outputs the majority vote across these copies. The probability of failure for this new protocol is: $\Pr[\geq r/2$ copies of π fail$] \leq \binom{r}{r/2}\delta^{r/2} \leq (4\delta)^{r/2}$. This achieves failure probability e^{-k/γ_c} for $r = O_c(\frac{k}{\log(1/\delta)})$. The lower bound of $\Omega_c(n)$ on the communication complexity of e^{-k/γ_c}-error protocols implies that the communication cost of π is lower bounded by $\Omega(n\frac{\log(1/\delta)}{k})$ in this case. ◀

4 Lower Bounds for ℓ_2-Heavy Hitters

In this section, we will prove lower bounds for certain variants of the ℓ_2 heavy hitters problem in the insertion-only model. Our first lower bound follows from some simple observations and the lower bounds that follow use reductions from the Mostly Set Disjointness problem and the lower bound proved in the previous section.

▶ **Definition 19.** *Given $p > 1$, in the ε-ℓ_p-heavy hitters problem, we are given $\varepsilon \in (0, 1)$ and a stream of items $a_1, \ldots, a_m$ where $a_i \in [n]$. If f_i denotes the frequency of item i in the stream, the algorithm should output all the elements $j \in [n]$ such that:*

$$|f_j| \geq \varepsilon \left\| f \right\|_p$$

▶ **Theorem 20.** *Given $\varepsilon \in (0, \frac{1}{4}]$, any deterministic linear sketching algorithm for the ε-ℓ_2-heavy hitters problem must use at least $\Omega(n)$ bits of space even for nonnegative vectors.*

Proof. Assume for the sake of contradiction that $r = o(n)$ and $M \in \mathbb{R}^{r \times n}$ is the sketching matrix which is associated with a deterministic algorithm for $1/4$-ℓ_2 heavy hitters. We may assume that M has orthonormal rows (else there is an orthonormal $r \times n$ matrix whose sketch is linearly related to the sketch in the algorithm and we consider that matrix).

Since M is orthonormal we have $\sum_{i\in[n]} \left\| M^T M e_i \right\|_2^2 \leq r$. So, there must exists an $i^* \in [n]$ such that $\left\| M^T M e_{i^*} \right\|_2^2 \leq r/n$. Consider the vector $v = e_{i^*} - M^T M e_{i^*}$ which lies in the kernel of M. Observe that $v_{i^*}^2 \geq (1 - r/n)^2 \geq 1/2$ and $\left\| v \right\|_2^2 \leq 1$ since $I - M^T M$ is a projection.

Now, let us define $w \in \mathbb{R}^n$ such that for all $j \neq i^*$, $w_j = |v_j|$ and $w_{i^*} = 0$. Observe that $w + v$ is a non-negative vector and that i^* is a heavy hitter in $(w + v)$ because $(w + v)_{i^*}^2 \geq 1/2$ and $\left\| w + v \right\|_2^2 \leq (2 \left\| v \right\|_2)^2 \leq 4$. Since v is in the kernel of M , $M(w + v) = Mw$ and the algorithm must give the same output for both $(w + v)$ and w. However, i^* is a heavy hitter in $(w + v)$ and is not a heavy hitter in w. Hence, by contradiction, $r = \Omega(n)$. ◀

In Theorem 21, we prove a lower bound on the space complexity of δ-error r-pass streaming algorithm for ε-ℓ_p-heavy hitters through a reduction from Mostly Set Disjointness.

▶ **Theorem 21.** *Given $\varepsilon \in (\frac{1}{n^{1/p}}, \frac{1}{2})$ and $p \geq 1$, any δ-error r-pass insertion-only streaming algorithm for ε-ℓ_p-heavy hitters requires $\Omega(\min(\frac{n^{1-1/p}}{r\varepsilon}, n^{1-2/p} \frac{\log(1/\delta)}{r\varepsilon^2}))$ bits of space.*

Proof. Let $\mathcal{A}$ be a δ-error r-pass streaming algorithm for ε-ℓ_p-heavy hitters in the insertion-only model. We describe a multiparty protocol to deterministically solve the Mostly Set Disjointness problem i.e., $\mathsf{MostlyDISJ}_{n,\varepsilon(4n)^{\frac{1}{p}},2\varepsilon(4n)^{\frac{1}{p}}}$ that uses the $\mathcal{A}$. The players simulate a stream which updates a vector $x \in \mathbb{R}^{2n}$. Instead of starting with 0^{2n} (as is the case with most streaming algorithms), the protocol starts off with a vector

$$f_0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}0\\ \vdots\\ 0\end{matrix}}\right\} n \\ \left.\vphantom{\begin{matrix}1\\ \vdots\\ 1\end{matrix}}\right\} n \end{matrix}$$

Each player performs an update $f \leftarrow f + \delta_i$ to the vector and passes the state of $\mathcal{A}$ to the next player. The update vector δ_i that is processed by player i is just their input x_i padded to length $2n$.

$$\delta = \begin{pmatrix} \\ x_i \\ \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Observe that if the input to the players is a NO-instance of $\mathsf{MostlyDISJ}_{n,\varepsilon(4n)^{1/p},2\varepsilon(4n)^{1/p}}$, then the final vector f' in the turnstile stream consists of 0-1 entries with at least n 1-s. So, $\|f'\|_p^p \geq n$ and since $\varepsilon \geq n^{1/p}$, no element is a ε-ℓ_p heavy hitter.

If the input is a YES-instance, then the final vector f' consists of $\leq 2n-1$ entries that are 1 and one entry at which is $\varepsilon(4n)^{\frac{1}{p}}$. Since $4\varepsilon^p n \geq \varepsilon^p(2n + 4\varepsilon^p n)$, that entry is a ε-heavy hitter. Using the lower bound of Theorem 1, we know that the total communication in the protocol is $\Omega(\min(n, n\frac{\log(1/\delta)}{\varepsilon n^{1/p}}))$. Since the number of messages sent over r rounds in the protocol is $r \cdot 2\varepsilon(4n)^{1/p}$, there exists at least one player whose communication is:

$$\Omega\Big(\min\big(\frac{n^{1-\frac{1}{p}}}{r\varepsilon}, \frac{n^{1-\frac{2}{p}}\log(1/\delta)}{r\varepsilon^2}\big)\Big)$$

bits and this is a lower bound on the space complexity of $\mathcal{A}$. ◀

A deterministic lower bound follows as a consequence of this lower bound.

▶ **Theorem 22.** *For any $\varepsilon \in (\frac{1}{n^{1/p}}, \frac{1}{2})$ and $p \geq 1$, any r-pass deterministic insertion-only streaming algorithm for ε-ℓ_p-heavy hitters must have a space complexity of $\Omega(\frac{n^{1-1/p}}{r\varepsilon})$ bits.*

In real world applications, one is concerned with lower bounds for naturally occurring frequency vectors. One such naturally occurring frequency distribution is a power law frequency distribution where the i^{th} most frequent element has frequency $\propto \frac{1}{i^\zeta}$ where ζ typically lies in $(0.5, 1]$. Formally:

▶ **Definition 23.** *Let $f \in \mathbb{R}^n$ be a vector such that $|f_{(1)}| \geq |f_{(2)}| \geq \dots |f_{(n)}|$. We say that this vector is power law distributed with parameter ζ if for all $i \in [n]$,*

$$|f_{(i)}| = \Theta(f_{(1)} \cdot i^{-\zeta}) + O(1)$$

In the next theorem, we prove a lower bound on the space complexity of streaming algorithms for ℓ_p-heavy hitters when the frequency vector is power law distributed. We denote $H_m = \sum_{i=1}^{\infty} i^{-m}$ which is finite when $m > 1$.

▶ **Theorem 24.** *Given $p \geq 1$, $\zeta \in (\frac{1}{p}, 1]$ and $\varepsilon \in (\frac{1}{n^\zeta}, \frac{1}{(2+2\cdot H_{p\zeta})^{1/p}})$, any δ-error r-pass streaming algorithm for the ε-ℓ_p-heavy hitters problem where the frequency vector is power law distributed with parameter ζ must have space complexity of $\Omega(\min(n^{1-\zeta}, n^{1-2\zeta}\log(1/\delta)))$.*

Proof. Let $\mathcal{A}$ be a one-pass deterministic streaming algorithm for ℓ_2 heavy hitters when the frequency vector is power law distributed with parameter ζ. We will use a reduction similar to the Theorem 21 to deterministically solve $\mathsf{MostlyDISJ}_{n,n^\zeta,2n^\zeta}$ using $\mathcal{A}$.

Instead of padding the initial vector f_0 with 1's as in Theorem 22, we pad with $\frac{2n^\zeta}{i^\zeta}$ for $i \in [2, n]$.

$$f_0 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 2n^{\zeta} \cdot 2^{-\zeta} \\ 2n^{\zeta} \cdot 3^{-\zeta} \\ \vdots \\ 2 \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix}}\right\} n \\ \left.\vphantom{\begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix}}\right\} n \end{matrix}$$

Now, suppose the players pass this frequency vector and successively perform updates to obtain the final frequency vector f'. In the YES instance, there exists one index $i \in [n]$ such that $|f'_i|^p = n^{p\zeta}$ and in the NO instance for all $i \in [n]$, we have $|f'_i| \leq 1$. In the NO case, we have $\|f'\|_p^p \geq \sum_{i\in[2,n+1]} 2n^{p\zeta} \cdot i^{-p\zeta} \geq n^{p\zeta}$ and in the YES case

$$\begin{aligned} \|f'\|_p^p &= \sum_{i\in[2n]} (f'_i)^p \\ &\leq n^{p\zeta} + n + \sum_{i\in[2,n+1]} 2n^{p\zeta} \cdot i^{-p\zeta} \\ &\leq n^{p\zeta} + n + H_{p\zeta} 2n^{p\zeta} \\ &< (2 + 2H_{p\zeta})n^{p\zeta}. \end{aligned}$$

So, in the YES instance, the heavy element is a ε-ℓ_p-heavy hitter since $\varepsilon^p < \frac{1}{2(1+H_{p\zeta})}$ and in the NO instance all the ℓ_p-heavy hitters are indices in $[n+1, 2n]$. Now, the final player runs the ℓ_p-heavy hitter algorithm and if any element from $[1, n]$ is a heavy hitter they output YES and they output NO otherwise.

So, we have described a reduction from ℓ_p heavy hitters for power law distributed vectors to Mostly Set Disjointness. Using Theorem 1, the total communication here is lower bounded by $\Omega(n, n^{1-\zeta}\log(1/\delta))$. Since there are n^{ζ} players, the space complexity lower bound for the streaming algorithm is $\Omega(n^{1-\zeta}, n^{1-2\zeta}\log(1/\delta))$. ◀

5 Application to Low Rank Approximation

As an application of our deterministic ℓ_2-heavy hitters lower bound in insertion streams, we prove a lower bound for the low rank approximation problem in the standard row-arrival model in insertion streams: we see rows $A_1, A_2, \ldots, A_n$ each in $\mathbb{R}^d$, one at a time. At the end of the stream we should output a projection P onto a rank-k space for which $\|A - AP\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$, where A_k is the best rank-k approximation to A. The FrequentDirections algorithm provides a deterministic upper bound of $O(dk/\epsilon)$ words of memory (assuming entries of A are $O(\log(nd))$ bits and a word is $O(\log(nd))$ bits) was shown in [59, 35], and a matching lower bound of $\Omega(dk/\epsilon)$ words of memory was shown in [75]. See also [34] where the upper and lower bounds were combined and additional results for deterministic algorithms were shown.

A natural question is if FrequentDirections can be improved when the rows of your matrix are sparse. Indeed, the sparse setting was shown to have faster running times in [33, 45]. Assuming there are n rows and each row has s non-zero entries, the running time was shown to be $O(sn(k + \log n) + nk^3 + d(k/\epsilon)^3)$, significantly improving the nd time required for dense matrices. Another question is if one can improve the memory required in the sparse setting. The above lower bound has an $\Omega(d)$ term in its complexity because of the need to store directions in $\mathbb{R}^d$. However, it is well-known [40] that any matrix A contains $O(k/\epsilon)$ rows whose row-span contains a rank-k projection P for which $\|A - AP\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2$. Consequently, it is conceivable in the stream one could use $O(sk/\epsilon)$ words of memory in the sparse setting, which would be a significant improvement if $s \ll d$. Indeed, in the related communication setting, this was shown to be possible in [10], whereby assuming the rows have at most s non-zero entries it is possible to find such a P with communication only $O(sk/\epsilon)$ words per server, improving upon the $O(dk/\epsilon)$ words per server bound for general protocols, at least in the randomized case. It was left open if the analogous improvement was possible in the streaming setting, even for deterministic algorithms such as FrequentDirections.

Here we use our deterministic lower bound to show it is not possible to remove a polynomial dependence on d in the memory required in streaming setting for deterministic algorithms.

▶ **Theorem 25.** *Any 1-pass deterministic streaming algorithm outputting a rank-k projection matrix P providing a $(1+\epsilon)$-approximate rank-k low rank approximation requires $\Omega(\sqrt{d})$ bits of memory, even for $k = 1$, $\epsilon = \Theta(1)$, and when each row of A has only a single non-zero entry.*

Proof. Recall in one instantiation of our hard communication problem, the players have sets $S_1, \ldots, S_{\sqrt{d}} \subseteq \{1, 2, \ldots, d\}$ each of size $\sqrt{d}/2$ and either the sets are pairwise disjoint or there exists a unique element i^* occurring in at least $2/3$ fraction of the sets. We associate each element i in each set S_ℓ with a row of A which the standard unit vector e_i which is 1 in position i and 0 in all remaining positions. The stream is defined by seeing all the rows corresponding to elements in S_1, then in S_2, and so on.

Suppose we have seen the first $1/2$ fraction of sets in the stream. In this case, the row i^* must have occurred in at least $1/2 - 1/3 = 1/6$ fraction of sets. Thus, at this point in the stream, the top singular value of A is $\sqrt{d}/6$ and all remaining singular values of A equal 1. Now, the algorithm outputs a rank-1 projection P from its internal memory state. Suppose $P = vv^T$ for a unit vector v. Then

$$\|A - Avv^T\|_F^2 = \|A\|_F^2 - \|Av\|_2^2 \geq \|A\|_F^2 - 1 + (d/36)v_{i^*}^2.$$

Consequently, to obtain a C-approximation for a sufficiently small constant $C > 1$, we must have $v_{i^*}^2 = \Omega(1)$. Since $\|v\|_2^2 = 1$, there is a set T of size $O(1)$ which contains all indices j for which $v_j^2 = \Omega(1)$.

Now, since we have only observed a $1/2$ fraction of sets in the stream, the element i^* must occur in at least $2/3 - 1/2 = 1/6$ fraction of sets in the remaining half of the stream. Thus, for each element in the set T, we can check if it occurs at all in the second half of the stream. However, if there is such an element i^*, it must be the only element in T occurring in the second half of the stream. In case the sets in our hard instance are pairwise disjoint, no element in T will occur in the second half of the stream. Thus, we can deterministically distinguish which of the two cases we are in.

Note that the maximum communication of this reduction is the memory size of the streaming algorithm, together with an additional additive $O(\log d)$ bits of memory to store T. Thus, we get that the memory required of our streaming algorithm is at least $\Omega(\sqrt{d}) - O(\log d) = \Omega(\sqrt{d})$ bits. ◀

6 Algorithm for bounded-length turnstile streams

In this section we show that ℓ_2 heavy hitters on turnstile streams of length $O(n)$ can be solved in $O(n^{2/3})$ space. This is intermediate between the $O(\sqrt{n})$ possible in the insertion-only model and the $\Omega(n)$ necessary in linear sketching.

▶ **Theorem 26.** *There is a deterministic ℓ_2 heavy hitters algorithm for length-L strict turnstile streams with ± 1 updates using $O((L/\varepsilon)^{2/3})$ words of space.*

Proof. Let S be a parameter to be determined later. We run three algorithms in parallel: space-$O(S)$ FrequentElements on the positive updates to x; space-$O(S)$ FrequentElements on the negative updates to x (with sign flipped to be positive); and a linear sketching algorithm for exact S-sparse recovery (e.g., Reed-Solomon syndrome decoding).

Let P, N be the number of positive/negative updates, respectively, so $L = P + N$. Let x^+ and x^- be the sum of positive/negative updates, so $x = x^+ - x^-$. The two FrequentElements sketches give us estimates $\widehat{x}^+$ and $\widehat{x}^-$, respectively, such that for each i:

$$x_i^+ - P/S \le \widehat{x}_i^+ \le x_i^+ \qquad\qquad x_i^- - N/S \le \widehat{x}_i^- \le x_i^-$$

Therefore $\widehat{x} := \widehat{x}^+ - \widehat{x}^-$ satisfies

$$\|\widehat{x} - x\|_\infty \le \max(P/S, N/S) \le L/S.$$

Second, the S-sparse recovery algorithm gives us a $\widehat{y}$ such that, if $\|x\|_0 \le S$, $\widehat{y}_i = x_i$ for all i.

For a strict turnstile stream, we can compute $\|x\|_1 = P - N$. Our algorithm outputs the ε-heavy hitters of $\widehat{y}$ if $\|x\|_1 \le S$, and otherwise outputs the entries of $\widehat{x}$ larger than $3L/S$.

Since $\|x\|_0 \le \|x\|_1$, the output is exactly correct when $\|x\|_1 \le S$. Otherwise, $\|x\|_2 \ge \sqrt{\|x\|_1} \ge \sqrt{S}$, so for $S \ge (L/\varepsilon)^{2/3}$,

$$\|\widehat{x} - x\|_\infty \le L/S \le \varepsilon\sqrt{S} \le \varepsilon\|x\|_2.$$

Therefore the algorithm will output all 4ε-heavy hitters and only 2ε-heavy hitters. Rescaling ε by 4 gives the standard ℓ_2 heavy hitters guarantee. ◀

▶ Remark 27. For non-strict turnstile streams, one can still achieve the ℓ_∞/ℓ_2 guarantee

$$\|\widehat{z} - x\|_\infty \le \varepsilon\|x\|_2$$

with the same space, but the ℓ_2 heavy hitters guarantee (of outputting all ε-heavy hitters and only $\varepsilon/2$-heavy hitters) requires $\Omega(\min(n, L))$ space.

Proof. To achieve the ℓ_∞/ℓ_2 guarantee, we combine $\widehat{x}$ and $\widehat{y}$ in the above algorithm slightly differently: if $\|\widehat{y} - \widehat{x}\|_\infty \le L/S$, output $\widehat{y}$; else, output $\widehat{x}$. Call this output $\widehat{z}$. We have that $\|\widehat{z} - x\|_\infty \le \|\widehat{z} - \widehat{x}\|_\infty + \|\widehat{x} - x\|_\infty \le 2L/S$ unconditionally, and $\widehat{z} = x$ if $\|x\|_0 \le S$. The algorithm outputs $\widehat{z}$.

So when $\|x\|_0 \le S$, this algorithm recovers x exactly and certainly finds the heavy hitters. On the other hand, when $\|x\|_0 \ge S$, we have $\|x\|_2 \ge \sqrt{S}$. Therefore for $S \ge 2(L/\varepsilon)^{2/3}$,

$$\|\widehat{z} - x\|_\infty \le 2L/S \le \varepsilon\sqrt{S} \le \varepsilon\|x\|_2$$

as desired.

For the lower bound, it suffices to consider $L = \Theta(n)$ [otherwise, restrict to the first $\Theta(L)$ coordinates/do nothing interesting after the first $O(n)$ updates]. We can solve Equality on $b = n/10$ bits as follows: using a constant-distance, constant-rate code, associate each input $y \in \{0,1\}^b$ with a codeword $C_y \in \{0,1\}^{n-1}$, such that $\|C_y - C_{y'}\|_1 > n/10$ for all $y \neq y'$. Alice, given the input y, inserts $x_1 = 1$, then inserts C_y on the remaining coordinates. She sends the sketch of the result to Bob, who subtracts his $C_{y'}$ from coordinates $2, \ldots, n$ and asks for the ε-heavy hitters of the result. For any $1 > \varepsilon > 10/\sqrt{n}$, this list will contain coordinate 1 if and only if $y = y'$, solving equality, giving the desired $\Omega(n)$ bound. [And since ε-heavy hitters exactly reconstructs binary vectors on $1/\varepsilon^2$ coordinates, an $\Omega(n)$ bound for $\varepsilon \leq O(1/\sqrt{n})$ is trivial.] ◀

References

1 Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499, 1994.

2 Yuqing Ai, Wei Hu, Yi Li, and David P Woodruff. New characterizations in turnstile streams with applications. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

3 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

4 Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*, pages 1–16, 2002.

5 Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004. doi:10.1016/j.jcss.2003.11.006.

6 Paul Beame and Trinh Huynh. Multiparty communication complexity and threshold circuit size of \sfac^0. *SIAM Journal on Computing*, 41(3):484–518, 2012.

7 Paul Beame, Toniann Pitassi, Nathan Segerlind, and Avi Wigderson. A strong direct product theorem for corruption and the multiparty communication complexity of disjointness. *Computational Complexity*, 15(4):391–432, 2006.

8 Radu Berinde, Piotr Indyk, Graham Cormode, and Martin J. Strauss. Space-optimal heavy hitters with strong error bounds. *ACM Trans. Database Syst.*, 35(4):26, 2010. doi:10.1145/1862919.1862923.

9 Kevin S. Beyer and Raghu Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA.*, pages 359–370, 1999.

10 Christos Boutsidis, David P Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 236–249, 2016.

11 Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1011–1020, 2016.

12 Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, Jelani Nelson, Zhengyu Wang, and David P. Woodruff. Bptree: an ℓ_2 heavy hitters algorithm using constant memory. *CoRR*, abs/1603.00759, 2016.

13 Vladimir Braverman, Stephen R. Chestnut, Nikita Ivkin, and David P. Woodruff. Beating countsketch for heavy hitters in insertion streams. *STOC*, 2016.

14 Vladimir Braverman, Jonathan Katzman, Charles Seidell, and Gregory Vorsanger. An optimal algorithm for large frequency moments using o(n^(1-2/k)) bits. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pages 531–544, 2014.

15 Yousra Chabchoub, Christine Fricker, and Hanene Mohamed. Analysis of a bloom filter algorithm via the supermarket model. In *21st International Teletraffic Congress, ITC 2009, Paris, France, September 15-17, 2009*, pages 1–8, 2009.

16 Amit Chakrabarti, Graham Cormode, and Andrew McGregor. A near-optimal algorithm for estimating the entropy of a stream. *ACM Transactions on Algorithms*, 6(3), 2010.

17 Amit Chakrabarti and Sagar Kale. Strong fooling sets for multi-player communication with applications to deterministic estimation of stream statistics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 41–50, 2016.

18 Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 107–117. IEEE, 2003.

19 Ho-Leung Chan, Tak-Wah Lam, Lap-Kei Lee, Jiangwei Pan, Hing-Fung Ting, and Qin Zhang. Edit distance to monotonicity in sliding windows. In *International Symposium on Algorithms and Computation*, pages 564–573. Springer, 2011.

20 Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004.

21 Arkadev Chattopadhyay and Anil Ada. Multiparty communication complexity of disjointness. *arXiv preprint*, 2008. `arXiv:0801.3624`.

22 Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

23 A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, 22(1):211–231, 2009.

24 Graham Cormode. Open problem in data streams and related topics. *IITK Workshop on Algorithms for Data Streams*, 2006.

25 Graham Cormode and Marios Hadjieleftheriou. Finding frequent items in data streams. *PVLDB*, 1(2):1530–1541, 2008.

26 Graham Cormode and S Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

27 Erik D Demaine, Alejandro López-Ortiz, and J Ian Munro. Frequency estimation of internet packet streams with limited space. In *Algorithms—ESA 2002*, pages 348–360. Springer, 2002.

28 Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. Lower bounds for sparse recovery. *CoRR*, abs/1106.0365, 2011.

29 Cristian Estan and George Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Trans. Comput. Syst.*, 21(3):270–313, 2003.

30 Min Fang, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, and Jeffrey D. Ullman. Computing iceberg queries efficiently. In *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 299–310, 1998.

31 Sumit Ganguly. Deterministically estimating data stream frequencies. In Ding-Zhu Du, Xiaodong Hu, and Panos M. Pardalos, editors, *Combinatorial Optimization and Applications, Third International Conference, COCOA 2009, Huangshan, China, June 10-12, 2009. Proceedings*, volume 5573 of *Lecture Notes in Computer Science*, pages 301–312. Springer, 2009.

32 Ankit Garg, Tengyu Ma, and Huy Nguyen. On communication cost of distributed statistical estimation and dimensionality. In *Advances in Neural Information Processing Systems*, pages 2726–2734, 2014.

33 Mina Ghashami, Edo Liberty, and Jeff M Phillips. Efficient frequent directions algorithm for sparse matrices. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 845–854, 2016.

34 Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

35 Mina Ghashami and Jeff M Phillips. Relative errors for deterministic low-rank matrix approximations. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 707–717. SIAM, 2014.

36 Anna C Gilbert, Hung Q Ngo, Ely Porat, Atri Rudra, and Martin J Strauss. L2/l2-foreach sparse recovery with low risk. *arXiv preprint*, 2013. `arXiv:1304.6232`.

37 Parikshit Gopalan and Jaikumar Radhakrishnan. Finding duplicates in a data stream. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 402–411, 2009.

38 Vince Grolmusz. The bns lower-bound for multiparty protocols is nearly optimal. *Information and computation*, 112(1):51–54, 1994.

39 André Gronemeier. Asymptotically optimal lower bounds on the nih-multi-party information complexity of the and-function and disjointness. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

40 Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1207–1214. SIAM, 2012.

41 Jiawei Han, Jian Pei, Guozhu Dong, and Ke Wang. Efficient computation of iceberg cubes with complex measures. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data, Santa Barbara, CA, USA, May 21-24, 2001*, pages 1–12, 2001.

42 Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 1–12, 2000.

43 Nicholas J. A. Harvey, Jelani Nelson, and Krzysztof Onak. Sketching and streaming entropy via approximation theory. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 489–498, 2008.

44 Christian Hidber. Online association rule mining. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA.*, pages 145–156, 1999.

45 Zengfeng Huang. Near optimal frequent directions for sketching dense and sparse matrices. In *International Conference on Machine Learning*, pages 2048–2057, 2018.

46 Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 202–208, 2005.

47 Rajesh Jayaram and David P Woodruff. Data streams with bounded deletions. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Prin ciples of Database Systems*, pages 341–354. ACM, 2018.

48 Thathachar S Jayram and David P Woodruff. The data stream space complexity of cascaded norms. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 765–774. IEEE, 2009.

49 TS Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 562–573. Springer, 2009.

50 Hossein Jowhari, Mert Saglam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In Maurizio Lenzerini and Thomas Schwentick, editors, *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece*, pages 49–58. ACM, 2011.

51 John Kallaugher and Eric Price. Separations and equivalences between turnstile streaming and linear sketching. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1223–1236, 2020.

52 Ravi Kannan, Santosh Vempala, and David Woodruff. Principal component analysis and higher correlations for distributed data. In *Conference on Learning Theory*, pages 1040–1057, 2014.

53 Richard M Karp, Scott Shenker, and Christos H Papadimitriou. A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems (TODS)*, 28(1):51–55, 2003.

54 Abhishek Kumar and Jun (Jim) Xu. Sketch guided sampling - using on-line estimates of flow size for adaptive data collection. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 23-29 April 2006, Barcelona, Catalunya, Spain*, 2006.

55 Kasper Green Larsen, Jelani Nelson, Huy L. Nguyen, and Mikkel Thorup. Heavy hitters via cluster-preserving clustering. *Commun. ACM*, 62(8):95–100, 2019.

56 Troy Lee and Adi Shraibman. Disjointness is hard in the multiparty number-on-the-forehead model. *Computational Complexity*, 18(2):309–336, 2009.

57 Yi Li, Huy L Nguyen, and David P Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 174–183, 2014.

58 Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 174–183, 2014. doi:10.1145/2591796.2591812.

59 Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588, 2013.

60 Debmalya Mandal, Ariel D. Procaccia, Nisarg Shah, and David P. Woodruff. Efficient and thrifty voting by any means necessary. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7178–7189, 2019.

61 Debmalya Mandal, Nisarg Shah, and David P. Woodruff. Optimal communication-distortion tradeoff in voting. In *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, pages 795–813, 2020.

62 Gurmeet Singh Manku and Rajeev Motwani. Approximate frequency counts over data streams. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 346–357. VLDB Endowment, 2002.

63 Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *Proceedings of the 10th International Conference on Database Theory*, ICDT'05, pages 398–412, Berlin, Heidelberg, 2005. Springer-Verlag. doi:10.1007/978-3-540-30570-5_27.

64 Jayadev Misra and David Gries. Finding repeated elements. *Sci. Comput. Program.*, 2(2):143–152, 1982.

65 Morteza Monemizadeh and David P. Woodruff. 1-pass relative-error L_p-sampling with applications. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1143–1160, 2010.

66 Shanmugavelayutham Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

67 Eric Price and David P Woodruff. Lower bounds for adaptive sparse recovery. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 652–663. SIAM, 2013.

68 Ashok Savasere, Edward Omiecinski, and Shamkant B. Navathe. An efficient algorithm for mining association rules in large databases. In *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland.*, pages 432–444, 1995.

69 Alexander A Sherstov. The multiparty communication complexity of set disjointness. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 525–548, 2012.

70 Alexander A Sherstov. Communication lower bounds using directional derivatives. *Journal of the ACM (JACM)*, 61(6):1–71, 2014.

71 Xiaoming Sun, David P. Woodruff, Guang Yang, and Jialin Zhang. Querying a matrix through matrix-vector products. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 94:1–94:16, 2019.

72 Pascal Tesson. Computational complexity questions related to finite monoids and semigroups, 2003.

73 Hannu Toivonen. Sampling large databases for association rules. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3-6, 1996, Mumbai (Bombay), India*, pages 134–145, 1996.

74 Wikipedia contributors. Jensen–Shannon divergence – Wikipedia, the free encyclopedia, 2020. [Online; accessed 06-November-2020]. URL: `https://en.wikipedia.org/w/index.php?title=Jensen%E2%80%93Shannon_divergence&oldid=980081721`.

75 David Woodruff. Low rank approximation lower bounds in row-update streams. In *Advances in Neural Information Processing Systems*, pages 1781–1789, 2014.

76 David P Woodruff. New algorithms for heavy hitters in data streams. *arXiv preprint*, 2016. `arXiv:1603.01733`.

77 David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 941–960, 2012.

78 David P Woodruff and Qin Zhang. An optimal lower bound for distinct elements in the message passing model. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 718–733. SIAM, 2014.

Toward Better Depth Lower Bounds: The XOR-KRW Conjecture

Ivan Mihajlin
St. Petersburg Department of Steklov Mathematical Institute of
Russian Academy of Sciences, Russia

Alexander Smal
St. Petersburg Department of Steklov Mathematical Institute of
Russian Academy of Sciences, Russia

Abstract

In this paper, we propose a new conjecture, the XOR-KRW conjecture, which is a relaxation of the Karchmer-Raz-Wigderson conjecture [10]. This relaxation is still strong enough to imply $\mathbf{P} \not\subseteq \mathbf{NC}^1$ if proven. We also present a weaker version of this conjecture that might be used for breaking n^3 lower bound for De Morgan formulas. Our study of this conjecture allows us to partially answer an open question stated in [5] regarding the composition of the universal relation with a function. To be more precise, we prove that there exists a function g such that the composition of the universal relation with g is significantly harder than just a universal relation. The fact that we can only prove the existence of g is an inherent feature of our approach.

The paper's main technical contribution is a new approach to lower bounds for multiplexer-type relations based on the non-deterministic hardness of non-equality and a new method of converting lower bounds for multiplexer-type relations into lower bounds against some function. In order to do this, we develop techniques to lower bound communication complexity in half-duplex and partially half-duplex communication models.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases communication complexity, KRW conjecture, circuit complexity, half-duplex communication complexity, Karchmer-Wigderson games, multiplexer relation, universal relation

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.38

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2020/116/

Acknowledgements We would like to thank the anonymous reviewers who have done a tremendous job carefully reading our paper and whose detailed comments helped us significantly improve the text of the paper and make it more readable.

1 Introduction

1.1 Background

Proving lower bounds on the Boolean formula complexity is one of the classical problems of computational complexity theory. For over 40 years, the researchers had been developing the methods for proving lower bounds – starting with the works of Subbotovskaya [17] and Khrapchenko [12] all the way to the celebrated work of Håstad [6]. As a result, the researchers managed to achieve a cubic lower bound on the formula complexity of an explicit Boolean function (Andreev's function). This lower bound has been unbeaten for over 20 years (up to lower order terms, see. [18] for more information).

Karchmer, Raz, and Wigderson [10] suggested an approach for proving superpolynomial formula size lower bound for Boolean functions from class **P**. The suggested approach is to prove lower bounds on the formula depth of *the block-composition* of two arbitrary Boolean functions.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 38; pp. 38:1–38:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

▶ **Definition 1.** *Let* $f : \{0,1\}^m \to \{0,1\}$ *and* $g : \{0,1\}^n \to \{0,1\}$ *be Boolean functions.* The block-composition $f \diamond g : (\{0,1\}^n)^m \to \{0,1\}$ *is defined by*

$$(f \diamond g)(x_1, \dots, x_m) = f(g(x_1), \dots, g(x_m)),$$

where $x_1, \dots, x_m \in \{0,1\}^n$.

Let $\mathrm{D}(f)$ denotes the minimal depth of De Morgan formula for function f. It is easy to show that $\mathrm{D}(f \diamond g) \leq \mathrm{D}(f) + \mathrm{D}(g)$ by constructing a formula for $f \diamond g$ by substituting every variable in a formula for f with a copy of the formula for g. Karchmer, Raz, and Wigderson [10] conjectured that this upper bound is roughly optimal.

▶ **Conjecture 2** (The KRW conjecture). *Let* $f : \{0,1\}^m \to \{0,1\}$ *and* $g : \{0,1\}^n \to \{0,1\}$ *be non-constant functions. Then*

$$\mathrm{D}(f \diamond g) \approx \mathrm{D}(f) + \mathrm{D}(g).$$

If the conjecture is true then there is a polynomial-time computable function that does not have De Morgan formula of polynomial size, and hence $\mathbf{P} \not\subseteq \mathbf{NC}^1$. Consider the function $h : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, which interprets its first input as a truth table of a function $f : \{0,1\}^{\log n} \to \{0,1\}$ and computes the value of the block-composition of $\log n / \log\log n$ functions f on its second input:

$$h(f, x) = (\underbrace{f \diamond \dots \diamond f}_{\log n/\log\log n})(x).$$

It is not hard to see that $h \in \mathbf{P}$. To show that $h \notin \mathbf{NC}^1$, let $\tilde{f}$ be a function with maximal depth complexity. By Shannon's counting argument $\tilde{f}$ has depth complexity roughly $\log n$. Assuming the KRW conjecture, the function $\tilde{f} \diamond \dots \diamond \tilde{f}$ has depth complexity roughly $\log n \cdot (\log n / \log\log n) = \omega(\log n)$, and hence $\tilde{f} \diamond \dots \diamond \tilde{f} \notin \mathbf{NC}^1$. Any formula for h must compute $\tilde{f} \diamond \dots \diamond \tilde{f}$ if we hard-wire $f = \tilde{f}$ in it, so $h \notin \mathbf{NC}^1$. This argument is especially attractive since it does not seem to break any known meta mathematical barriers such as the concept of "natural proofs" by Razborov and Rudich [16] (the function h is very special, so the argument does not satisfy "largeness" property). It worth noting that the proof would work even assuming some weaker version of the KRW conjecture, like $\mathrm{D}(f \diamond g) \geq \mathrm{D}(f) + \epsilon \cdot \mathrm{D}(g)$ or $\mathrm{D}(f \diamond g) \geq \epsilon \cdot \mathrm{D}(f) + \mathrm{D}(g)$ for some $\epsilon > 0$.

The seminal work of Karchmer and Wigderson [11] established a correspondence between De Morgan formulas for non-constant Boolean function f and communication protocols for the Karchmer-Wigderson game for f.

▶ **Definition 3.** The Karchmer-Wigderson game (KW game) *for Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *is the following communication problem: Alice gets an input* $x \in \{0,1\}^n$ *such that* $f(x) = 0$*, and Bob gets as input* $y \in \{0,1\}^n$ *such that* $f(y) = 1$*. Their goal is to find a coordinate* $i \in [n]$ *such that* $x_i \neq y_i$*. The KW game can be considered as a communication problem for* the Karchmer-Wigderson relation for f*:*

$$\mathrm{KW}_f = \{(x, y, i) \mid x, y \in \{0,1\}^n, i \in [n], f(x) = 0, f(y) = 1, x_i \neq y_i\}.$$

Karchmer and Wigderson showed that the communication complexity of KW_f is exactly equal to the depth formula complexity of f. This correspondence allows us to use communication complexity methods for proving formula depth lower bounds. In fact, Conjecture 2 can be reformulated in terms of communication complexity of the Karchmer-Wigderson game for the block-composition of two arbitrary Boolean functions. Let $\mathrm{CC}(R)$ denotes deterministic communication complexity of a relation R. For convenience, we also define *a block-composition for KW relations*, so that the following equality holds: $\mathrm{KW}_{f \diamond g} = \mathrm{KW}_f \diamond \mathrm{KW}_g$. This leads to the following reformulation of the KRW conjecture.

▶ **Conjecture 4** (The KRW conjecture (reformulation)). *Let $f : \{0,1\}^m \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}$ be non-constant functions. Then*

$$\mathrm{CC}(\mathrm{KW}_f \diamond \mathrm{KW}_g) \approx \mathrm{CC}(\mathrm{KW}_f) + \mathrm{CC}(\mathrm{KW}_g).$$

The study of Karchmer-Wigderson games had already been shown to be a potent tool in the monotone setting – the monotone KW games were used to separate the monotone counterparts of classes $\mathbf{NC}^1$ and $\mathbf{NC}^2$ [11]. Therefore, there is a reason to believe that the communication complexity perspective might help to prove new lower bounds in the non-monotone setting.

In a series of works [4, 7, 5, 3, 13, 1] several steps were taken towards proving the KRW conjecture. In the first two works [4, 7] the authors proved the similar bound for the block-composition of two *universal relations*.

▶ **Definition 5.** The universal relation *of length n,*

$$\mathrm{U}_n = \{(x, y, i) \mid x, y \in \{0,1\}^n, i \in [n], x_i \neq y_i\} \cup \{(x, x, \perp) \mid x \in \{0,1\}^n\}.$$

A communication problem for the universal relation is a generalization of the Karchmer-Wigderson games: Alice and Bob are given n-bit distinct strings and their goal is to find a coordinate $i \in [n]$ such that $x_i \neq y_i$. In contrast to KW games, in this game Alice and Bob can be given the same input string – in that case, they have to output a special symbol $\perp$ to indicate that the promise is broken. Intuitively, the universal relation is a more complex communication problem than KW game because the players do not have proof that their inputs are different. For any non-constant $f : \{0,1\}^n \to \{0,1\}$, there is a natural reduction from KW_f *to* U_n*: given inputs (x, y) for* KW_f *the players follow a protocol for* U_n*, the protocol outputs some i such that $x_i \neq y_i$, the players output i as it is a correct output for* KW_f*. The block-composition of the universal relations generalizes the block-composition of KW games in the same manner. A similar reduction uses a protocol for the block-composition of the universal relations to solve the block-composition KW games. Thus, proving lower bounds for the universal relations seems to be a natural first step.*

In the subsequent works [5, 13], the authors proved a lower bound on the block-composition of the Karchmer-Wigderson relation for an arbitrary function and the universal relation. This result is presented in terms of the number of leaves rather than formula depth. In [3], the authors presented an alternative proof for the block-composition of an arbitrary function with the parity function in the framework of the Karchmer-Wigderson games (this result was originally proved in [6] using an entirely different approach). Their result gives an alternative proof of the cubic lower bound for Andreev's function [6]. In the most recent paper [1] of the series, the authors extended the range of inner functions that can be handled in the monotone version of the KRW conjecture to all functions whose depth complexity can be lower bounded via query-to-communication lifting theorem. They also introduce an intermediate semi-monotone setting where only inner function is monotone and show a lower bound on the composition of the (non-monotone) universal relation with every monotone inner function for which a lower bound can be proved using a lifting theorem.

In the last section of [4], the authors introduced *the same function multiplexer communication game*, that is very similar to the Karchmer-Wigderson game for *the multiplexer function.*

▶ **Definition 6.** The multiplexer function *of size n is a function* $\mathsf{M}_n : \{0,1\}^{2^n} \times \{0,1\}^n \to \{0,1\}$ *with two arguments, such that* $\mathsf{M}_n(f,x) = f_x$*. It is convenient to interpret the string f as a truthtable of some function $f : \{0,1\}^n \to \{0,1\}$, so we can say that* $\mathsf{M}_n(f,x) = f(x)$*.*

In the KW game for M_n, Alice gets a function $f : \{0,1\}^n \to \{0,1\}$ and $x \in \{0,1\}^n$, such that $f(x) = 0$, Bob gets a function $g : \{0,1\}^n \to \{0,1\}$ and $y \in \{0,1\}^n$, such that $g(y) = 1$. Their goal is to find a coordinate $i \in [2^n + n]$ such that $(f,x)_i \neq (g,y)_i$. The authors of [4] suggest to consider a version of this game where players are given the same function, i.e., $f = g$, so they only need to find the differing coordinate between x and y.

▶ **Definition 7.** *In* the same function multiplexer communication game (the multiplexer game) MUX_n*, Alice gets a function $f : \{0,1\}^n \to \{0,1\}$ and $x \in \{0,1\}^n$ such that $f(x) = 0$, Bob gets the same function $g : \{0,1\}^n \to \{0,1\}$ and $y \in \{0,1\}^n$ such that $g(y) = 1$. Their goal is to find a coordinate $i \in [n]$ such that $x_i \neq y_i$, or output $\perp$ if $f \neq g$ (if $x \neq y$ and $f \neq g$ then both outputs are possible).*

The same function multiplexer communication game can be considered as a generalization of the Karchmer-Wigderson games for Boolean functions on n bits. Indeed, solving the KW game for any $g : \{0,1\}^n \to \{0,1\}$ can be reduced to the same function multiplexer game: Alice and Bob are given g and the corresponding x and y. Given that we already have a lower bound on $f \diamond \mathrm{U}_n$ [5, 13], it looks natural to study the block-composition of the KW game for an arbitrary function and the same function multiplexer game. The detailed explanation how a lower bound on the block-composition of the KW game for an arbitrary function and the same function multiplexer might be used to separate $\mathbf{P}$ and $\mathbf{NC}^1$, see [15] for details (to the best of our knowledge, this result was independently proved by Russell Impagliazzo).

▶ Remark 8. The KW game for M_n can also be considered as a generalization of KW games using the same reduction. On the other hand, it is unclear whether lower bounds on the block-composition with it implies any new results. Moreover, the following lower bound applies. Let $\mathrm{L}(f)$ denotes the minimal size of De Morgan formula computing f.

▶ **Theorem 9.** *For any $m, n \in \mathbb{N}$ with $n \geq 6 \log m$, and any non-constant function $f : \{0,1\}^m \to \{0,1\}$,*

$$\mathrm{CC}(\mathrm{KW}_{f \diamond \mathsf{M}_n}) \geq \log L(f) + n - O(\log^* n).$$

The proof is given in Appendix B.

1.2 The XOR-KRW conjecture

As an alternative to the block-composition, we define a new composition operation.

▶ **Definition 10.** *For any $n, m, k \in \mathbb{N}$ with $k \mid n$, and functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^k \to \{0,1\}^k$* the XOR-composition $f \boxplus_m g : (\{0,1\}^n)^m \to \{0,1\}$ *is defined by*

$$(f \boxplus_m g)(x_{1,1}, \ldots, x_{n/k,m}) = f\left(g(x_{1,1}) \oplus \cdots \oplus g(x_{1,m}), \ldots, g(x_{n/k,1}) \oplus \cdots \oplus g(x_{n/k,m})\right),$$

where $x_{i,j} \in \{0,1\}^k$ for all $i \in [n/k]$ and $j \in [m]$, and $\oplus$ denotes bit-wise XOR.

This composition becomes a stronger version of the block composition if we consider case of $n = m = k$. In this case, both compositions are mapping an $n \times n$ matrix into a vector and then applying a function to it. But in the XOR-composition every bit of the vector depends on the entire matrix rather than just one row. However we will focus on the case of constant m as we believe it might be sufficient for our goals.

We suggest the following generalization of the KRW conjecture.

▶ **Conjecture 11** (The XOR-KRW conjecture)**.** *There exist $m \in \mathbb{N}$ and $\epsilon > 0$, such that for all natural $n, k \in \mathbb{N}$ with $k \mid n$, and every non-constant $f : \{0,1\}^n \to \{0,1\}$, there exists $g : \{0,1\}^k \to \{0,1\}^k$,*

$$\mathrm{D}(f \boxplus_m g) \geq \mathrm{D}(f) + \epsilon k.$$

Using the ideas from [10] one can show that XOR-KRW implies $\mathbf{P} \neq \mathbf{NC}^1$.

▶ **Theorem 12.** *If Conjecture 11 is true then $\mathbf{P} \neq \mathbf{NC}^1$.*

Proof. Suppose Conjecture 11 is true. Let f be any non-constant function from $\{0,1\}^{\log n}$ to $\{0,1\}$, and let $m \in \mathbb{N}$ be provided by Conjecture 11. For every $t \in \mathbb{N}$, consider a function h_t defined by:

$$h_t(x, g_1, g_2, \ldots g_t) = (f \boxplus_m g_1 \boxplus_m g_2 \boxplus_m \cdots \boxplus_m g_t)(x),$$

where $x \in \{0,1\}^{m^t \log n}$ and $g_i : \{0,1\}^{\log n} \to \{0,1\}^{\log n}$ for all $i \in [t]$. Conjecture 11 implies that there exist $m \in \mathbb{N}$ and $g_1, \ldots, g_t : \{0,1\}^{\log n} \to \{0,1\}^{\log n}$, such that $\mathrm{D}(f \boxplus_m g_1 \boxplus_m g_2 \boxplus_m \cdots \boxplus_m g_t) = \mathrm{D}(h_t) \geq \epsilon t \log n - O(t)$. For $t = \log n$ that gives us

$$\mathrm{D}(h_{\log n}) \geq \epsilon \log^2 n - O(\log n).$$

Now lets estimate the size of the input to $h_{\log n}$. Each g_i requires $n \log n$ bits of description, x requires $m^{\log n} \log n = n^{\log m} \log n = n^{O(1)}$. So, the size of the input to $h_{\log n}$ is $N = n^{O(1)}$ bits, and $\mathrm{D}(h_{\log n}) \geq \epsilon \log^2 n - O(\log n) = \Omega(\log^2 N)$. Thus, $h_{\log n} \notin \mathbf{NC}^1$. On the other hand, we can compute $h_{\log n}$ in a natural way in $\mathbf{P}$. ◀

The idea behind the XOR-KRW conjecture is influenced by the constructions used in the areas of pseudorandomness and cryptography, where bit-wise xor is used to achieve better results. The proof of hardness of the composition of the universal relations is based on the idea that any protocol that makes progress solving the top relation of the composition is leaking very little information about the actual inputs of the composition. We hope that the additional entanglement provided by taking entry-wise xor of multiple copies of a gadget function g will make it possible to use the same kind of argument about the composition of functions.

In this paper we will focus on specific case of $k = n$. In this case, $f \boxplus_m g$ has the same number of inputs as f. This is not the regime we need for the KRW conjecture in order to separate $\mathbf{P}$ and $\mathbf{NC}^1$, as the proof of the Theorem 12 uses KRW for the case of $k \ll n$. But let us scale our ambitions down a bit. One of the current major challenges of circuit complexity is to beat the $\Omega(n^3)$ lower bound for a specific formula. As we already have mentioned, this bound was proved by Håstad in [6] and was not improved rather than by lower terms since then. If we only aim to prove a supercubic lower bound for a specific formula then we can only focus on the case $k = n$. For $k = n$, the definition of the XOR-composition a bit simpler.

▶ **Definition 13** (A special case of Definition 10 for $k = n$)**.** *For $n, m \in \mathbb{N}$ and functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}^n$ the* XOR-composition *$f \boxplus_m g : (\{0,1\}^n)^m \to \{0,1\}$ is defined by*

$$(f \boxplus_m g)(x_1, \ldots, x_m) = f\left(g(x_1) \oplus \cdots \oplus g(x_m)\right),$$

where $x_i \in \{0,1\}^n$ for all $i \in [m]$.

This definition allows us to formulate a weak version of the XOR-KRW conjecture.

▶ **Conjecture 14** (The weak XOR-KRW conjecture). *There exists $m \in \mathbb{N}$ and $\epsilon > 0$, such that for all $n \in \mathbb{N}$, for any non-constant functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}^n$:*

$$\mathrm{D}(f \boxplus_m g) \geq \mathrm{D}(f) + \epsilon n.$$

We also introduce a version of this conjecture for a formula size rather than depth. Proving that this conjecture is true would allow us to beat $\Omega(n^3)$ formula size lower bound.

▶ **Conjecture 15** (The weak XOR-KRW conjecture for formula size). *There exists $m \in \mathbb{N}$ and $\epsilon > 0$, such that for all $n \in \mathbb{N}$, for any non-constant function $f : \{0,1\}^n \to \{0,1\}$ there exists a non-constant function $g : \{0,1\}^n \to \{0,1\}^n$:*

$$\mathrm{L}(f \boxplus_m g) \geq 2^{\epsilon n} \cdot \mathrm{L}(f).$$

The weak XOR-KRW conjecture implies the existence of a function $h = f \boxplus_m g$ for some $f : \{0,1\}^{\log n} \to \{0,1\}$, $g : \{0,1\}^{\log n} \to \{0,1\}^{\log n}$ and $m \in \mathbb{N}$, such that $\mathrm{CC}(\mathrm{KW}_h) \geq (1+\epsilon)\log n$. In order to prove a cubic lower bound for the Andreev's function one needs to hardwire a hard function into it's description. We define a modified Andreev's function that takes the XOR-composition of functions instead. Note that there are $n^{\log n + 1}$ pairs of functions $f : \{0,1\}^{\log n} \to \{0,1\}$ and $g : \{0,1\}^{\log n} \to \{0,1\}^{\log n}$. That means that one can encode h with $\theta(n \log n)$ bit.

▶ **Definition 16.** *For $n \in \mathbb{N}$ that is a power of two, any $m \in \mathbb{N}$, and any functions $f : \{0,1\}^{\log n} \to \{0,1\}$ and $g : \{0,1\}^{\log n} \to \{0,1\}^{\log n}$* the XOR-composed Andreev's function $\mathrm{Andr}_{\boxplus m}$ *is defined by*

$$\mathrm{Andr}_{\boxplus m}(f, g, x_1, \ldots, x_{m \log n}) = (f \boxplus_m g)\big(\oplus_n(x_1), \cdots, \oplus_n(x_{m \log n})\big),$$

where $x_i \in \{0,1\}^n$ for $i \in [m \log n]$, and $\oplus_n(x)$ denotes the sum of all bits of x modulo 2.

The input size of $\mathrm{Andr}_{\boxplus m}$ is $\Theta(n \log n)$. It is also important that there is a natural polynomial time algorithm for $\mathrm{Andr}_{\boxplus m}$.

▶ **Theorem 17.** *Conjecture 15 implies that $\mathrm{L}(\mathrm{Andr}_{\boxplus m}) = \Omega(n^{3+\epsilon})$ for some $m \in \mathbb{N}$.*

The proof of this theorem is identical to the original proof of Håstad with only difference that we can now hardwire functions f and g for some hard f and g provided by the conjecture.

As the main result of this paper we show that some form of XOR-KRW conjecture holds for XOR-composition of the universal relation and the KW game for some hard function. It would be interesting to see if our techniques could be extended to handle the case of $k < n$. It feels that this setting is significantly more sensitive and would require more intricate proof. In this paper, we focused on the regime of $k = n$ since this is the regime that is useful for super-cubic formula lower bounds, but the regime of smaller k's would be useful for other applications.

1.3 Techniques and Results

The paper's main technical contribution is a new approach to lower bounds for multiplexer-type relations based on the non-deterministic hardness of non-equality and a new method of converting lower bounds for multiplexer-type relations into lower bounds against some function. We define two communication problems based on the XOR-composition and prove lower bounds on it: a XOR-composition of the universal relation with the KW game for some function g, we denote it $\mathrm{U}_n \boxplus \mathrm{KW}_g$, and the XOR-composition of the universal relation with the multiplexer relation, we denote it $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. Both communication problems are based on the XOR-composition for $m = 2$. Our proofs also allow to get a lower bound for the standard block composition of the universal relation and a function (see Appendix A).

Further in this section we discuss a special case of Definition 10 for $m = 2$, which is sufficient for our purposes. Then we will discuss the problem $\mathrm{U}_n \boxplus \mathrm{KW}_g$, which is a relaxed version of the weak KRW-conjecture, and describe our main result, which is a lower bound for this problem. Next, we discuss an even more relaxed version of the problem $\mathrm{U}_n \boxplus \mathrm{MUX}_n$, and describe our second result, which is a lower bound to that problem. Finally, we describe how the second result is proved, and how we use it to derive the first result.

▶ **Definition 18** (Special case of Definition 13 for $m = 2$). *For functions* $f : \{0,1\}^n \to \{0,1\}$ *and* $g : \{0,1\}^n \to \{0,1\}^n$ the XOR-composition $f \boxplus g$ *is defined by*

$$(f \boxplus g)(x, y) = f(g(x) \oplus g(y)),$$

where $x, y \in \{0,1\}^n$*.*

In the definitions of the problems below, we are going to use a communication problem that is a generalization of the Karchmer-Wigderson game for non-Boolean functions. So, it is convenient to extend the definition of the KW game to handle the case of multioutput functions.

▶ **Definition 19.** The Karchmer-Wigderson game *for function* $g : \{0,1\}^n \to \{0,1\}^k$ *is the following communication problem: Alice gets an input* $x \in \{0,1\}^n$*, Bob gets as input* $y \in \{0,1\}^n$*. Their goal is to find a coordinate* $i \in [n]$ *such that* $x_i \neq y_i$*. If* $g(x) = g(y)$ *then the players are allowed to output* $\perp$*.*

Recall that our ultimate goal is to prove a lower bound for $f \boxplus g$. As an intermediate problem, we consider a version of this game f replaced with the universal relation. In a communication game for $\mathrm{KW}_{f \boxplus g}$, Alice is given $x_a, y_a \in \{0,1\}^n$, such that $(f \boxplus g)(x_a, y_a) = 0$, and Bob is given $x_b, y_b \in \{0,1\}^n$, such that $(f \boxplus g)(x_b, y_b) = 1$. Their goal is to find $i \in [2n]$ such that $(x_a \circ y_a)_i \neq (x_b \circ y_b)_i$. We now replace f with U_n, so the players only know that $g(x_a) \oplus g(y_a) \neq g(x_b) \oplus g(y_b)$.

▶ **Definition 20.** *Let* $g : \{0,1\}^n \to \{0,1\}^n$*. A communication game* $\mathrm{U}_n \boxplus \mathrm{KW}_g$ *is the XOR-composition of* U_n *and* KW_g *in the following way: Alice is given* $x_a, y_a \in \{0,1\}^n$ *and Bob is given* $x_b, y_b \in \{0,1\}^n$*. Their goal is to find* $i \in [2n]$ *such that* $(x_a \circ y_a)_i \neq (x_b \circ y_b)_i$*. If* $g(x_a) \oplus g(y_a) = g(x_b) \oplus g(y_b)$ *they can output* $\perp$*.*

The trivial upper bound for $\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g)$ is $\mathrm{CC}(\mathrm{KW}_g) + n + O(\log n) \leq 2n + O(\log n)$: Alice sends x_a to Bob, and Bob compares it with x_b. If he finds a difference then he sends the answer to Alice using $O(\log n)$ bits of communication. Otherwise, they simulate the shortest protocol for KW_g on y_a and y_b that outputs some index j. If $(y_a)_j \neq (y_b)_j$ then they output $n + j$, otherwise they output $\perp$. We are going to prove that there exists a function $g : \{0,1\}^n \to \{0,1\}^n$ such that $\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g) \geq 1.5n - O(\log n)$.

▶ **Theorem 21.** *For all* $n \in \mathbb{N}$*, there exists* $g : \{0,1\}^n \to \{0,1\}^n$ *such that*

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g) \geq 1.5n - O(\log n).$$

This theorem partially answer an open question from [5] showing a lower bound for the XOR-composition of the universal relation with a function. The answer is partial because the original open question was to prove a composition result for $\mathrm{U} \diamond \mathrm{KW}_g$ for every function g, and we prove that there exists some hard function g for which a composition result holds. We also only focus on the case where both U and g have the same input length. A corresponding result for the block-composition follows from our proof. See Appendix A for details.

In order to prove the result on $\mathrm{U}_n \boxplus \mathrm{KW}_g$, we will consider a similar communication problem where the function g is given to the players as a part of the input rather than being hardwired into definition of the problem.

▶ **Definition 22.** *In a* communication problem $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ *Alice is given* $x_a, y_a \in \{0,1\}^n$ *and* $g_a : \{0,1\}^n \to \{0,1\}^n$*, Bob is given* $x_b, y_b \in \{0,1\}^n$ *and* $g_b : \{0,1\}^n \to \{0,1\}^n$*. Their goal is to find* $i \in [2n]$ *such that* $(x_a \circ y_a)_i \neq (x_b \circ y_b)_i$*. If* $g_a(x_a) \oplus g_a(y_a) = g_b(x_b) \oplus g_b(y_b)$ *or* $g_a \neq g_b$ *they can output* $\perp$*.*

In some sense, the communication problem $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ contains an instance of $\mathrm{U}_n \boxplus \mathrm{KW}_g$ for every g as a special case where Alice and Bob receive g as a part of the input. So, on the one hand, it might be easier to prove a lower bound for it as it is a more complex problem. On the other hand, it seems that there is a natural way of arguing that a lower bound on $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ implies a lower bound on $\mathrm{U}_n \boxplus \mathrm{KW}_g$ for some g: if the problem is hard in common, then it has to be hard in some of the special cases.

The trivial upper bound for $\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{MUX}_n)$ is $2n + O(\log n)$: Alice sends x_a and y_a to Bob, he compares it with x_b and y_b, and then he either finds a difference or realizes that they are allowed to output $\perp$. At the end, Bob sends the answer to Alice using $O(\log n)$ bits of communication. We prove the following lower bound using a reductions from non-deterministic communication complexity.

▶ **Theorem 23.** *For all* $n \in \mathbb{N}$*,* $\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{MUX}_n) \geq 1.5n - o(n)$*.*

After we prove this lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$, we will translate it to a lower bound on $\mathrm{U}_n \boxplus \mathrm{KW}_g$ for some g. The problem $\mathrm{U}_n \boxplus \mathrm{KW}_g$ is a special case of $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ for fixed g. The intuition suggests that if $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ is hard then there should be some function g such that $\mathrm{U}_n \boxplus \mathrm{KW}_g$ is hard. Thus, to get a lower bound for $\mathrm{U}_n \boxplus \mathrm{KW}_g$ for some g from a lower bound on $\mathrm{U}_n \boxplus \mathrm{MUX}_n$, we need to show that $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ is at most as hard as $\mathrm{U}_n \boxplus \mathrm{KW}_g$ for the "hardest" function that we can feed to the players, and hence we can hard-wire this "hardest" function in $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ to get $\mathrm{U}_n \boxplus \mathrm{KW}_g$. Let's forget about the outer U_n for a bit, and consider MUX_n. It seems almost obvious that the complexity of MUX_n is equal to the complexity of the hardest function: given some function g in the MUX_n game the players can use the optimal protocol for KW_g, hence the complexity of MUX_n is upper bounded by the complexity of the hardest KW_g. The same idea should work for the composed problems like $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. However, this argument is incorrect. In the argument we assume that the players choose a protocol depending on the function g they have got as a part of the input. This is not possible in the classical model of communication complexity. Suppose that in the best protocol for KW_{g_1} Alice sends the first message, while in the best protocol for KW_{g_2} for $g_2 \neq g_2$ the first message is sent by Bob. Then it is not clear who sends first in the protocol for MUX_n. There is a natural workaround – we can consider only alternating protocols where Alice sends every odd message and Bob sends every even message [15]. The drawback of this approach is that all the lower bounds in this setting have to be multiplied by $1/2$ when translated to the unrestricted case, that might make them useless for proving non-trivial bounds. This obstacle motivated the study of half-duplex communications models [9, 2]. In half-duplex communication models, every player can send messages in every round, but if both players send simultaneously, then their messages get lost. Thus, if we use half-duplex communication model instead of the classical one, then the described problem will not arise, and we can show that the complexity of $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ is at most the complexity of $\mathrm{U}_n \boxplus \mathrm{KW}_g$ for some function g. Using a technique that employs half-duplex communication, we translate the lower bound of Theorem 23 to $\mathrm{U}_n \boxplus \mathrm{KW}_g$.

1.4 Organization of this paper

In Section 2, we review the required preliminaries. In Section 3, we prove a lower bound for the XOR-composition of the universal relation with the multiplexer relation using a reduction from non-deterministic communication complexity (Theorem 23). In Section 4, we prove a lower bound for the XOR-composition of the universal relation with the KW game for some function using the same ideas together with the results from half-duplex communication complexity (Theorem 21). Section 5 contains a conclusion and open problems. In Appendix A, we show the block-composition analogue of Theorem 21. In Appendix B, we prove Theorem 9.

2 Preliminaries

2.1 Notation

Let us mention the notation used in this paper. We use $[k]$ as a shortcut for $\{1, \dots, k\}$, $\mathbb{B}$ as a shortcut for $\{0, 1\}$ and $\circ$ to denote concatenation of binary strings. Working with binary strings we use $\oplus$ for entry-wise xor: $\forall u, v \in \mathbb{B}^k : \ (v \oplus u)_i = v_i \oplus u_i$. For a set of tuples S we use $\pi_i(S)$ to denote the projection of S on the ith coordinate: $\pi_i(S) = \{e_i \mid (e_1, e_2, \dots, e_i, \dots) \in S\}$.

2.2 Communication complexity

We expect that the reader is familiar with the standard definitions of communication complexity that can be found in [14]. It will be important to understand how the nodes of communication protocol relate to combinatorial rectangles of the input matrix. Throughout the paper whenever we discuss rectangles we always mean the rectangles of the input matrix of the communication problem under consideration. If some rectangle has equal sides, i.e., it is equal to $A \times A$ for some set A, then we call it *a square*.

We are going to use the following simple theorem that is a generalization of the well-known lower bound for the equality function. For any non-empty finite set S, *the equality on* S is a function $\mathrm{EQ}_S : S \times S \to \mathbb{B}$, such that for all $a, b \in S$, $\mathrm{EQ}_S(a, b) = 1 \iff a = b$.

▶ **Theorem 24.** *For any non-empty finite set S, $\mathrm{CC}(\mathrm{EQ}_S) \geq \log |S|$.*

Proof. For any $a, b \in S$, $a \neq b$, a communication transcript on input (a, a) must be different from a transcript on input (b, b), otherwise the same transcript would correspond to (a, b) and (b, a). Thus, the length of the longest transcript is at least $\log |S|$. ◀

For convenience, we are going to use some basic results from non-deterministic communication complexity. Let X and Y be non-empty finite sets.

▶ **Definition 25.** *We say that a function $f : X \times Y \to \mathbb{B}$ has* non-deterministic communication protocol *of complexity d if there are two functions $A : X \times \mathbb{B}^d \to \mathbb{B}$ and $B : Y \times \mathbb{B}^d \to \mathbb{B}$ such that*

- $\forall (x, y) \in f^{-1}(1)\ \exists w \in \mathbb{B}^d : A(x, w) = B(y, w) = 1,$
- $\forall (x, y) \in f^{-1}(0)\ \forall w \in \mathbb{B}^d : A(x, w) \neq 1 \vee B(y, w) \neq 1.$

The non-deterministic communication complexity *of f, denoted* $\mathrm{NCC}(f)$*, is the minimal complexity of a non-deterministic communication protocol for f.*

In contrast to deterministic case, the definition of non-deterministic complexity is asymmetric and hence the complexity of a function and its negation might be different. We will use the following lower bound for the negation of the equality function. For any non-empty finite set S *the non-equality on* S is a function $\mathrm{NEQ}_S : S \times S \to \mathbb{B}$, such that

$$\mathrm{NEQ}_S(a, b) = 1 - \mathrm{EQ}_S(a, b).$$

▶ **Theorem 26.** *For any non-empty finite set* S*,* $\mathrm{NCC}(\mathrm{NEQ}_S) \geq \log\log|S|$*.*

Proof. Assume, for the sake of contradiction, that for some S, $\mathrm{NCC}(\mathrm{NEQ}_S) = d \leq \log\log|S| - 1$. Then the following deterministic protocol solves EQ_S: Alice sends $A(x, w)$ for all possible $w \in \mathbb{B}^d$, Bob replies with 1 if and only if there is some $w \in \mathbb{B}^d : A(x, w) = B(x, w) = 1$. The complexity of this protocol is

$$2^d + 1 \leq 2^{\log\log|S|-1} + 1 = \frac{1}{2}\log|S| + 1 < \log|S|$$

that contradicts Theorem 24. ◀

Notable property of non-deterministic communication complexity is that it does not involve any communication at all. For our purposes it will be easier for us to think about the following alternative definition of non-deterministic communication, which is implicitly mentioned in the classical book by Nisan and Kushilevich [14].

▶ **Definition 27.** *We say that a function* $f : X \times Y \to \mathbb{B}$ *has* privately non-deterministic communication protocol *of complexity d if there is a function* $\hat{f} : (X \times \mathbb{B}^*) \times (Y \times \mathbb{B}^*) \to \mathbb{B}$ *of (deterministic) communication complexity at most d such that*

- $\forall (x, y) \in f^{-1}(1)\ \exists w_x, w_y \in \mathbb{B}^* : \hat{f}((x, w_x), (y, w_y)) = 1,$
- $\forall (x, y) \in f^{-1}(0)\ \forall w_x, w_y \in \mathbb{B}^* : \hat{f}((x, w_x), (y, w_y)) = 0.$

The privately non-deterministic communication complexity of f, *denoted* $\mathrm{NCC}'(f)$*, is the minimal depth of a privately non-deterministic communication protocol for* f*.*

This alternative definition of non-deterministic communication uses private witnesses instead of a public one, and hence the players need to communicate. Let us prove the equivalence of these definitions.

▶ **Theorem 28.** *For any function* $f : X \times Y \to \mathbb{B}$*,*

$$\mathrm{NCC}(f) + 2 \geq \mathrm{NCC}'(f) \geq \mathrm{NCC}(f).$$

Proof. To prove the first inequality, we suppose that there is a non-deterministic protocol of complexity d for f defined by functions A and B. Lets show that there is a privately non-deterministic protocol for f of complexity $d+2$. We define a function $\hat{f} : (X \times \mathbb{B}^*) \times (Y \times \mathbb{B}^*) \to \mathbb{B}$ such that

$$\hat{f}((x, w_x), (y, w_y)) = 1 \iff |w_x| = |w_y| = d \wedge A(x, w_x) = B(y, w_y) = 1 \wedge w_x = w_y.$$

This function has a deterministic protocol with $d + 2$ bits of communication: given some x Alice privately guesses $w_x \in \mathbb{B}^d$ and sends $w_x \circ A(x, w_x)$ to Bob, Bob privately guesses $w_y \in \mathbb{B}^d$ and replies with 1 if and only if $A(x, w_x) = B(y, w_y) = 1$ and $w_x = w_y$, otherwise he replies with 0.

Now we show the second inequality by constructing a non-deterministic protocol of complexity d given a privately non-deterministic protocol of complexity d. Let $\hat{f}$ defines the privately non-deterministic protocol for f, and let Π is a (deterministic) protocol for

$\hat{f}$ of depth d. In the non-deterministic protocol for f Alice and Bob interpret the public non-deterministic witness w as a transcript of Π on $((x, w_x), (y, w_y))$ for some (unknown) w_x and w_y. We define a function $A(x, w)$ such that $A(x, w) = 1$ if and only if there exists $w_x \in \mathbb{B}^*$ such that w is a valid transcript for (x, w_x) leading to output 1. Similarly, we define function $B(y, w)$ such that $B(y, w) = 1$ if and only if there exists $w_y \in \mathbb{B}^*$ such that w is a valid transcript for (y, w_y) leading to output 1. The resulting non-deterministic protocol for f defined by A and B has complexity d. ◀

▶ **Corollary 29.** *For any non-empty finite set S, $\mathrm{NCC}'(\mathrm{NEQ}_S) \geq \log\log|S|$.*

2.3 Half-duplex communication complexity

The essential property of the classical model of communication complexity proposed by Yao is that in every round of communication one player sends some bit and the other one receives it. In [9], the authors suggest a generalization of the classical communication model, *the half-duplex model*, where the players are allowed to speak simultaneously. Lets assume that the players have some synchronising mechanism, e.g., synchronised clock, that allows then understand when each round begins. Every round each player chooses one of three actions: send 0, send 1, or receive. There are three different types of rounds.

- If one player sends some bit and the other one receives then communication works like in the classical case, we call such rounds *normal* or *classical*.
- If both players send bits during the round then these bits get lost (the same happens if two persons try to speak via a "walkie-talkie" simultaneously), these rounds are called *wasted*.
- If both players receive, these rounds are called *silent*.

In [9], the authors consider three variations of this model based on what happens in silent rounds. We are going to focus on one of the models – *half-duplex communication with adversary*, where in silent round both players receive *some* bits. In order to solve a communication problem in half-duplex communication model with adversary the players have to devise a protocol that is correct for any bits that were received in silent rounds (the protocol must give a correct answer even if these bits were chosen by an adversary).

In the classical case, a protocol is a binary rooted tree that describes the communication of players on all possible inputs: every internal node corresponds to a state of communication and defines which of the players sends in this round. Unlike the classical case in half-duplex communication player does not always know what the other's player action was – the information about it can be "lost", i.e., in wasted rounds a player do not know what the other player's action was. It means that a player might not know what node of the protocol corresponds to the current state of communication. The protocol for half-duplex communication can be described by a pair of rooted trees of arity 4 that describe how Alice and Bob communicate on all possible inputs and for any bits they receive in silent rounds. The arity 4 stands for four possible events: send 0, send 1, receive 0, and receive 1. However, in this paper, it will be convenient for us to talk about the half-duplex protocol being a single tree that describes all the actions of players from the point of view of an external observer.

We can also think about half-duplex communication in a following way. In the classical communication protocol player's action (send or receive) is always defined by the previous communication. In half-duplex communication player's action can also depend on the input. We will also consider an intermediate model where player's action depends on the previous communication and a part of the input. We call such a model *partially half-duplex communication model*. In partially half-duplex communication problems the players receive inputs divided in two parts: Alice receives (f, x), Bob receives (g, y). They can use half-duplex protocols but with a restriction: if $f = g$ then the communication must have no non-classical rounds.

Let P be a communication problem with classical communication complexity k. It is not hard to see that half-duplex communication complexity is bounded between $k/2$ and k – classical protocol can be used in the half-duplex model and every half-duplex protocol can be simulated by a classical protocol of double depth where Alice sends only in even rounds and Bob sends only in odd rounds. In [9, 2], a series of non-trivial bounds were proved for various functions and KW relations.

We use CC^{hd} to denote the half-duplex communication complexity a communication problem with adversary.

▶ **Theorem 30** ([9]). *For any non-empty finite set S, $\mathrm{CC}^{hd}(\mathrm{EQ}_S) \geq \log|S|/\log 2.5$.*

The main motivation to study half-duplex communication comes from the following lemma.

▶ **Lemma 31.** *For all $n \in \mathbb{N}$, there exist a function $f : \mathbb{B}^n \to \mathbb{B}$ such that*

$$\mathrm{CC}(\mathrm{KW}_f) \geq \mathrm{CC}^{hd}(\mathrm{MUX}_n) - O(\log n).$$

The statement of this lemma seems almost trivial since it is easy to prove that there exists a function f such that $\mathrm{CC}(\mathrm{KW}_f) \geq n - O(\log n)$, and at the same time $\mathrm{CC}^{hd}(\mathrm{MUX}_n) \leq n + O(\log n)$. Nevertheless, we are going to prove it as a warm-up toward the proof of the main result to demonstrate how the half-duplex complexity comes into play. In the proof, Alice and Bob use the shortest protocols for given functions, and hence the lower bound on MUX_n would imply the existence of a hard function. Later when we will consider a multiplexer as a part of a XOR-composition with the universal relation, we will still be able to use the same argument to show the existence of a hard function.

Proof. Suppose that $\mathrm{CC}(\mathrm{KW}_f) \leq d$ for all $f : \mathbb{B}^n \to \mathbb{B}$. Consider the following half-duplex protocol for MUX_n. For every $f : \mathbb{B}^n \to \mathbb{B}$ let Π_f be the shortest (classical) protocol for KW_f. Alice, who is given f and x, follows the protocol Π_f using x as her input. Meanwhile Bob, who is given g and y, follows the protocol Π_g using y as his input. If f is different from g they might use different protocols, which is fine because we are in the half-duplex communication model.

When Alice reaches some leaf of Π_f she starts listening until the end of round d. Bob does the same. After d rounds of communication Alice has a candidate i for $x_i \neq y_i$, which is a valid output if $f = g$. Bob has a candidate j for $x_j \neq y_j$, that is equal to i if $f = g$. Now Alice and Bob just need to check that indeed $x_i \neq y_j$ and $i = j$, which can be done in $O(\log n)$. They output i if both conditions are true and $\perp$ otherwise. The total number of rounds of this half-duplex protocol for MUX_n is $d + O(\log n)$. ◀

This lemma shows that if we had a good understanding of half-duplex complexity we could translate lower bounds for multiplexer into the existence of a hard function. Unfortunately we will need to use a couple more tricks. Let CC^{phd} denotes partially half-duplex communication complexity of a communication problem with adversary.

▶ **Lemma 32.** *For all $n \in \mathbb{N}$, there exists a function $f : \mathbb{B}^n \to \mathbb{B}$ such that*

$$\mathrm{CC}(\mathrm{KW}_f) \geq \mathrm{CC}^{phd}(\mathrm{MUX}_n) - O(\log n).$$

Proof. The proof follows from proof of Lemma 31 by observing that the protocol for MUX_n in there is partially half-duplex: if $f = g$ the the players in fact follow the same classical protocol for KW_f. ◀

Now we are going to demonstrate how to prove lower bounds for partially half-duplex protocols.

▶ **Lemma 33.** *For all $n \in \mathbb{N}$, $\mathrm{CC}^{phd}(\mathrm{MUX}_n) \geq n - O(\log n)$.*

Proof. Let NEQ_{2^n} be a shortcut for non-equality on $\mathbb{B}^{2^n}$. We will show that $\mathrm{CC}^{phd}(\mathrm{MUX}_n) = d$ implies $\mathrm{NCC}(\mathrm{NEQ}_{2^n}) \leq d + O(\log n)$. Let Π be a partially half-duplex protocol for MUX_n. The main idea is that in partially half-duplex protocols for MUX_n any non-classical round indicates that the given functions are different. The non-deterministic protocol for NEQ_{2^n} goes as follows: the players guess a number $t \leq d$, a bit string $T \in \mathbb{B}^t$, and two bits $b_1, b_2 \in \mathbb{B}$. The players interpret T as a transcript of the first t rounds of Π such that it has only classical rounds (so, the communication can be described by t bits). Then they check that this transcript leads to a leaf marked with $\perp$ or to a non-classical round. To be more more precise, suppose Alice and Bob are given $f \in \mathbb{B}^{2^n}$ and $g \in \mathbb{B}^{2^n}$, respectively, as inputs for NEQ_{2^n}. The players guess a quadruple (t, T, b_1, b_2) as described. They have to check that

1. there exist $x \in f^{-1}(0)$ and $y \in g^{-1}(1)$ such that T is a valid transcript of the first t rounds of the protocol for MUX_n on input $((f, x), (g, y))$ assuming that all rounds are classical,
2. if $b_1 = 0$ then T is a transcript that ends up at a leaf labeled with $\perp$,
3. if $b_1 = 1$ and $b_2 = 0$ then both players were supposed to receive in round $t + 1$,
4. if $b_1 = 1$ and $b_2 = 1$ then both players were supposed to send in round $t + 1$.

Alice verifies that there exists x such that $f(x) = 0$ and T correctly describes first t rounds of communication on input (f, x). In addition, Alice checks the second condition and partially checks the last two conditions (i.e., if the third condition applies then Alice checks that she was supposed to receive in round $t + 1$, and if the fourth condition applies then she checks that she was supposed to send). Bob does the symmetric thing for y such that $g(y) = 1$. If there exist x and y that pass all the checks then the protocol for MUX_n on $((f, x), (g, y))$ either returns $\perp$ or contains a non-classical round. In both cases this is sufficient proof that $f \neq g$. Moreover, such a witness exists if and only if $f \neq g$. The size of the witness is $d + \log d + 2 = d + O(\log n)$.

The described protocol can be used to non-deterministically solve non-equality on binary strings of length 2^n. Theorem 26 implies $\mathrm{NCC}(\mathrm{NEQ}_{2^n}) \geq n$, so we can conclude that $d \geq n - O(\log n)$. ◀

The proof of this Lemma illustrates the important idea of reducing an instance of NEQ to the problem under consideration. Further in the paper, we will repeatedly use similar reductions.

3 Lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$

Let $\mathcal{P}$ be a set of all permutations of $\mathbb{B}^n$, and $N = 2^n$. Consider the following domain

$$\mathcal{X} = \mathcal{P} \times \mathbb{B}^n \times \mathbb{B}^n.$$

We are going to prove the following lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ on the rectangle $\mathcal{R} = \mathcal{X} \times \mathcal{X}$

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{MUX}_n) \geq \mathrm{CC}_{\mathcal{R}}(\mathrm{U}_n \boxplus \mathrm{MUX}_n) \geq 1.5n - O(\log n),$$

and hence get the desired lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$.

▶ **Theorem 23.** *For all $n \in \mathbb{N}$, $\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{MUX}_n) \geq 1.5n - o(n)$.*

To simplify our life a bit more we will stop applying g to one of the arguments inside $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. Consider the following communication problem (where $g(x) \oplus g(y)$ is replaced with $x \oplus g(y)$).

▶ **Definition 34.** *In a* communication problem $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ *Alice is given $x_a, y_a \in \mathbb{B}^n$ and $g_a : \mathbb{B}^n \to \mathbb{B}^n$, Bob is given $x_b, y_b \in \mathbb{B}^n$ and $g_b : \mathbb{B}^n \to \mathbb{B}^n$. Their goal is to find $i \in [2n]$ such that $(x_a \circ y_a)_i \neq (x_b \circ y_b)_i$. If $x_a \oplus g_a(y_a) = x_b \oplus g_b(y_b)$ or $g_a \neq g_b$ they can output $\perp$.*

If we can prove a lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ for it will also imply a lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. The same argument works for classical communication, for half-duplex communication and for partially half-duplex communication.

▶ **Lemma 35.** *For all $n \in \mathbb{N}$,*

$$\mathrm{CC}^*(\mathrm{U}_n \boxplus \mathrm{MUX}_n) \geq \mathrm{CC}^*(\mathrm{U}_n \boxplus \mathrm{MUX}'_n) - O(1),$$

where CC^* is one of CC, CC^{hd}, or CC^{phd}.

Proof. Suppose that $\mathrm{CC}^*(\mathrm{U}_n \boxplus \mathrm{MUX}_n) \leq h(n)$. Consider the following protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$. Alice is given x_a, y_a and g_a. Alice defines $x'_a = 0 \circ x_a$, $y'_a = 1 \circ y_a$, and

$$g'_a(b \circ z) = \begin{cases} 0 \circ z, & b = 0, \\ 0 \circ g_a(z), & b = 1. \end{cases}$$

Bob is given x_b, y_b and g_b. He defines x'_b, y'_b and g'_b in the same manner. Now the players can simulate the best protocol for $\mathrm{U}_{n+1} \circ \mathrm{MUX}_{n+1}$ of complexity at most $h(n+1) \leq h(n) + O(1)$ (this inequality is due to the linear upper bound on the complexity of $\mathrm{U}_n \boxplus \mathrm{MUX}_n$). Hence, $\mathrm{CC}^*(\mathrm{U}_n \boxplus \mathrm{MUX}'_n) \leq h(n) + O(1)$. ◀

The proof consists of two stages. At the first stage we go down the protocol tree and find a node at depth almost n (more precisely at depth $n - 3$) such that its rectangle contains many inputs that could be given to both to Alice and to Bob. Then we show that solving the problem on any large square requires depth about $\frac{n}{2}$. For the first stage we will use the following general lemma.

▶ **Lemma 36.** *Let P be a communication problem such that on a square $S \times S$ every monochromatic rectangle $A \times B$ has $|A \cap B| \leq \frac{|S|}{2^r}$ for some $r \geq 1$. Then for every $d \leq r$, every protocol that solves P on $S \times S$ has a node at depth d with rectangle $A \times B$ such that $|A \cap B| \geq \frac{|S|}{2^d}$.*

Proof. Proof by induction: the base case $d = 0$ is obvious. Now suppose that there exists a node at depth $d - 1$ with a rectangle $A' \times B'$ such that $|A' \cap B'| \geq \frac{|S|}{2^{d-1}}$. As $d - 1 < r$ we know that $A' \times B'$ is not monochromatic, and hence this node is not a leaf. W.l.o.g, assume that this node corresponds to Alice speaking. Let $A_0 \times B'$ and $A_1 \times B'$ be the children's rectangles, where $A' = A_0 \cup A_1$ and $A_0 \cap A_1 = \emptyset$. So, for some $i \in \{0, 1\}$ we have $|A_i \cap B'| \geq \frac{1}{2}|A' \cap B'| \geq \frac{|S|}{2^d}$. Which concludes the proof. ◀

We derive the following lemma from Lemma 36.

▶ **Lemma 37.** *For all natural $d \leq n$, any protocol tree that solves $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $\mathcal{R}$ has a node at depth d with a corresponding rectangle $A \times B$ such that $|A \cap B| \geq |\mathcal{X}|/2^d = N^2 \cdot |\mathcal{P}|/2^d$.*

Proof. Every monochromatic rectangle $A \times B$ of $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ is labeled with either an index or $\bot$. In the first case, $|A \cap B| = 0$. In the second case, for any $a = (g_a, x_a, y_a) \in A$ and $b = (g_b, x_b, y_b) \in B$ we have $g_a \neq g_b$ or $x_a \oplus g_a(y_a) = x_b \oplus g_b(y_b)$. We can subdivide all the elements of $C = A \cap B$ into 2^n disjoint groups $C = \bigcup_{z \in \mathbb{B}^n} C_z$, such that $(g, x, y) \in C_z$ if and only if $x \oplus g(y) = z$. For every two distinct $z_1, z_2 \in \mathbb{B}^n$ and inputs $(g_1, x_1, y_1) \in C_{z_1}$, $(g_2, x_2, y_2) \in C_{z_2}$, the permutations g_1 and g_2 are different (otherwise, $\bot$ would not be the correct output on this pair of inputs). Therefore, every permutation $g \in \mathcal{P}$ appear in at most one group. For fixed $g \in \mathcal{P}$ and $z \in \mathbb{B}^n$, there are only 2^n pairs $(x, y) : x \oplus g(y) = z$. That gives an upper bound on the number of elements in C, $|C| \leq 2^n \cdot |\mathcal{P}| = |\mathcal{X}|/2^n$. Application of Lemma 36 for $d \leq n$ concludes the proof. ◀

For the second lemma it is convenient to define the following combinatorial object that helps to understand the structure of a subset of inputs.

▶ **Definition 38.** *For a subset of inputs $S \subseteq \mathcal{X}$ we define* a domain graph *to be a bipartite graph $G_S = (U_S, V_S, E_S)$, such that $U_S \subseteq \mathcal{P}$, $V_S \subseteq \mathbb{B}^n \times \mathbb{B}^n$, and $(g, (x, y)) \in E_S \iff (g, x, y) \in S$.*

The statement of the next lemma seems to be very technical. The high-level idea is the following. We consider a large enough subset of inputs $S \subseteq \mathcal{X}$ with two additional properties saying that every function in S is defined on sufficiently many inputs and that for fixed $g \in \mathcal{P}$ and $y \in \mathbb{B}^n$ there are only a few $x \in \mathbb{B}^n$ such that $(g, x, y) \in S$. The first property is easy to achieve and the second comes from the proof of Theorem 23. The lemma shows that from such S we can extract a large set of functions H that will allow us reduce solving non-deterministic communication problem NEQ_H to solving (deterministic) communication problem $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S \times S$. So, we will be able to translate a lower bound of $\log\log |H|$ on the non-deterministic complexity of NEQ_H to a lower bound on deterministic complexity of $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S \times S$.

▶ **Lemma 39.** *Let $S \subseteq \mathcal{X}$ be a subset of inputs such that $|S| \geq N \cdot N!$, and let $G_S = (U_S, V_S, E_S)$ be a domain graph of S. If $\min_{g \in U_S}\{\deg_{G_S}(g)\} \geq 4N$ and*

$$\forall g \in \mathcal{P},\ \forall y \in \mathbb{B}^n,\ \left|\{x \mid (g, (x, y)) \in E_S\}\right| \leq \sqrt{N}, \tag{1}$$

then there is a set $H \subseteq U_S$ of size $2^{\Omega(\sqrt{N})}$ such that for all distinct $g_1, g_2 \in H$, there exist (x, y): $(g_1, x, y), (g_2, x, y) \in S$, and $g_1(y) \neq g_2(y)$.

Before we prove this lemma, lets look how it is used in the proof of Theorem 23.

Proof of Theorem 23. We start with applying Lemma 37 for $d = n - 3$ to find a rectangle $A \times B$ such that $|A \cap B| \geq 8NN!$. Let $S = A \cap B$ and $G_S = (U_S, V_S, E_S)$ be a domain graph of S. Average degree of the vertices in U_S is at least $8NN!/N! = 8N$. To increase the minimum degree we throw out all the vertices of low degree. Let $S' = S \setminus \{(g, x, y) \mid \deg_{G_S}(g) < 4N\}$. The size of $|S'| > |S| - 4N \cdot |\mathcal{P}| = 4NN!$. Let $G_{S'} = (U_{S'}, V_{S'}, E_{S'})$ be a domain graph of S'.

If there is $g \in \mathcal{P}$ and $y \in \mathbb{B}^n$ such that $\left|\{x \mid (g, (x, y)) \in E_{S'}\}\right| > \sqrt{N}$ then the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S' \times S'$ can be used to solve the equality problem on a set $W_{g,y} = \{x \mid (g, (x, y)) \in E_{S'}\}$. Given inputs $x_a, x_b \in W_{g,y}$, Alice and Bob simulate the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S' \times S'$ for inputs (g, x_a, y) and (g, x_b, y). If the protocol outputs $\bot$ then the players output 1, otherwise they output 0. For for inputs (g, x_a, y) and (g, x_b, y), the protocol outputs $\bot$ if and only if $x_a = x_b$, so this reduction gives a correct protocol for $\mathrm{EQ}_{W_{g,y}}$ of the same depth. By Theorem 24 any protocol for $\mathrm{EQ}_{W_{g,y}}$ has depth at least $\log |W_{g,y}| \geq \log(\sqrt{N}) = n/2$. By the reduction, the same lower bound applies for the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S' \times S'$.

Otherwise we apply Lemma 39 to construct a set H of size $2^{\Omega(\sqrt{N})}$. We are going to show that the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S' \times S'$ can be used to non-deterministically solve NEQ_H. Suppose that Alice and Bob are given $g_1 \in H$ and $g_2 \in H$ respectively, and they want to non-deterministically verify that $g_1 \neq g_2$ using a privately non-deterministic protocol. Alice privately guesses (x_a, y_a) such that $(g_1, x_a, y_a) \in S'$ and $k \in [n]$, Bob privately guesses (x_b, y_b) such that $(g_2, x_b, y_b) \in S'$. At first, the players verify that $x_a \oplus g_a(y_a) \neq x_b \oplus g_b(x_b)$: Alice sends k together with the k-th bit of $x_a \oplus g_a(y_a)$ and Bob compares it with the k-th bit of $x_b \oplus g_b(y_b)$. If the bits are equal then they reject (i.e., the function defining the privately non-deterministic protocol on these inputs equals 0). Otherwise, the players run the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S' \times S'$. If the protocol outputs $\perp$ then the private guesses give a valid proof of $g_1 \neq g_2$. Otherwise, if the protocol outputs some $i \in [2n]$ such that $(x_a, y_a)_i \neq (x_b, y_b)_i$ then the players reject. By Lemma 39, such private guesses exist for all distinct $g_1, g_2 \in H$. On the other hand, the statement of the problem $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ guarantees that if $x_a \oplus g_a(y_a) \neq x_b \oplus g_b(x_b)$ then the protocol can output $\perp$ only if $g_1 \neq g_2$. Thus, the depth of the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S' \times S'$ is at least

$$\mathrm{NCC}'(\mathrm{NEQ}_H) - O(\log n) = \log\log|H| - O(\log n) \geq n/2 - O(\log n).$$

Finally, we use Lemma 35 to translate the lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ to $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. ◀

Now it is time to prove Lemma 39.

Proof of Lemma 39. We are going to construct a rooted tree $T(S)$ such that

- each leaf ℓ is labeled with a set of functions $F_\ell \subseteq U_S$,
- each internal node v is labeled with a pair $(x_v, y_v) \in V_S$,
- for every leaf ℓ labeled with F_ℓ and every it's ancestor labeled with (x, y) there exists $a \in \mathbb{B}^n$ such that $\forall g \in F_\ell$, $g(y) = a$ and $(g, x, y) \in S$.
- for every two leaves labeled with F_1 and F_2, and their lowest common ancestor labeled with (x, y): $F_1 \cap F_2 = \emptyset$ and for all $g_1 \in F_1$, $g_2 \in F_2$, such that $g_1(y) \neq g_2(y)$,
- the number of leaves is a least $\frac{3^{\sqrt{N}}}{N}$.

Having such a tree, the set H is constructed by taking one function from every leaf. Indeed, the structure of the tree guarantees that for every $g_1, g_2 \in H$, $g_1 \neq g_2$, there exist (x, y), the label of the least common ancestor of corresponding leaves, such that $(g_1, x, y), (g_2, x, y) \in S$, and $g_1(y) \neq g_2(y)$.

The tree is defined recursively. For a set $Z \subseteq S$, let $T(Z)$ be a (non-empty) rooted tree. Let $G_Z = (U_Z, V_Z, E_Z)$ be a domain graph of Z. If $\min_{g \in U_Z}\{\deg_{G_Z}(g)\} \geq 2N$ then the rooted tree $T(Z)$ consists of a root node labelled with (x_Z, y_Z), where (x_Z, y_Z) is a vertex of maximal degree in V_Z, and a set of subtrees – for every $a \in \mathbb{B}^n$ such that $\exists g \in U_Z : (g, x_Z, y_Z) \in Z, g(y_Z) = a$ there is a subtree $T(Z_a)$ attached to the root node, where

$$Z_a = \{(g, x, y) \mid (g, x, y) \in Z, y \neq y_Z, g(y_Z) = a\}$$

Otherwise $T(Z)$ consists of one leaf node labeled with U_Z.

We are going to lower bound the number of leaves in $T(S)$ by lower bounding the number of nodes at depth $\sqrt{N}+1$. Let z be some node of $T(S)$ at depth $d \leq \sqrt{N}$ labeled with (x_Z, y_Z) corresponding to a root node of a subtree $T(Z)$ for some $Z \subseteq S$. Let $G_Z = (U_Z, V_Z, E_Z)$ be a domain graph of Z. Due to the condition (1) the minimal degree of vertices in U_Z can be lower bounded by $4N - d\sqrt{N} \geq 3N$. At the same time $|V_Z| \leq N(N - d)$. Let $T(Z_{a_1}), \ldots, T(Z_{a_k})$ – be the subtrees attached to z. Note that $\pi_1(Z_{a_i}) \cap \pi_1(Z_{a_j}) = \emptyset$ for all $i \neq j$, so the number of functions appearing in $Z_{a_1}, \dots, Z_{a_k}$ is exactly the number of functions in Z defined on (x_Z, y_Z). Given that (x_Z, y_Z) is a vertex of maximal degree in V_Z, the number of functions in the subtrees can be lower bounded as follows,

$$\left|\pi_1(Z_{a_1}) \sqcup \dots \sqcup \pi_1(Z_{a_k})\right| \geq \frac{|E_Z|}{|V_Z|} \geq \frac{3N|U_Z|}{N(N-d)} = \frac{3|U_Z|}{N-d}.$$

Thus by induction the total number of functions that appear in the sets at depth $d+1$ is at least

$$\frac{3^d \cdot |U_S|}{N(N-1)\cdots(N-d)} = \frac{3^d \cdot |U_S| \cdot (N-d-1)!}{N!},$$

where the size of U_S is at least $|S|/N^2 \geq N!/N$. Now we are ready to lower bound the number of nodes at depth $d+1$. Note that the number of permutations with k values fixed is $(N-k)!$, and hence a node at depth $d+1$ has at most $(N-d-1)!$ functions in its set. The number of nodes at depth $d+1$ is at least the total number of functions at depth $d+1$ divided by the upper bound on the number of functions in one node, that is

$$\frac{3^d \cdot |U_S| \cdot (N-d-1)!}{N!} / (N-d-1)! \geq \frac{3^d}{N}.$$

For $d = \sqrt{N}+1$ we get the desired lower bound $\frac{3^{\sqrt{N}}}{N} = 2^{\Omega(\sqrt{N})}$ on the number of leaves. ◀

4 Lower bound for $\mathrm{U}_n \boxplus \mathrm{KW}_g$

Our final goal is to show hardness of $\mathrm{U}_n \boxplus g$ for some function $g : \mathbb{B}^n \to \mathbb{B}^n$. Showing the lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ was the first step in this direction. As we discussed it earlier, it might be tempting to try to show that that hardness of multiplexer implies existence of a hard function. Unfortunately, the question whether that is true has remained open for decades. To get around this issue we will gradually extend the lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ using results from half-duplex communication complexity.

We start with extending the lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ to the half-duplex model.

▶ **Theorem 40.** *For all $n \in \mathbb{N}$,*

$$\mathrm{CC}^{hd}(\mathrm{U}_n \boxplus \mathrm{MUX}_n) \geq \left(\frac{1}{\log \frac{5}{2}} + \frac{1}{4}\right) n - O(1) \geq 1.006n - O(1).$$

The proof of this theorem mimics the proof for the classical case (Theorem 23). During the first stage, given a protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ we will find a large enough square $S \times S$, such that it is significantly easier to solve $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on this square. Then we will show that on every big square the problem is still hard. Finally, we apply Lemma 35 to get a result for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. The following lemma lower bounds the size of a square for the first stage.

▶ **Lemma 41.** *Let Π be a half-duplex protocol of length d that solves a communication problem on a rectangle $U \times U$. For every $t \leq d$ there exist a subset $S \subset U$ of size at least $(\frac{2}{5})^t \cdot |U|$, and a half-duplex protocol Π' of length $d - t$ that gives the same output as Π for all inputs from $S \times S$.*

Proof. In [8, Theorem 22], it is shown for $t = 1$. The general case follows by induction. ◀

Now we are ready to proof Theorem 40.

Proof of Theorem 40. Suppose $\mathrm{CC}^{hd}(\mathrm{U}_n \boxplus \mathrm{MUX}'_n) = d$ and let $t = \frac{n-3}{\log 2.5}$. According to Lemma 41 there is a subset $S \subset \mathcal{X}$ of size

$$|S| \geq \left(\frac{2}{5}\right)^t \cdot |\mathcal{X}| = \frac{8}{N} \cdot N^2 N! = 8NN!,$$

and a half-duplex protocol length $d - \frac{n-3}{\log 2.5}$ that can solve $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S \times S$. Any half-duplex protocol can be transformed into a classical one while at most doubling the length [9]. Then there is a length $2(d - \frac{n-3}{\log 2.5})$ classical protocol that solves $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S \times S$.

We apply the same argument as in the proof of Theorem 23 where we used Lemma 39 to solve NEQ_H using privately non-deterministic protocol, and we get

$$2\left(d - \frac{n-3}{\log 2.5}\right) \geq \frac{n}{2}.$$

Which gives us the following lower bound

$$d \geq \left(\frac{1}{\log 2.5} + \frac{1}{4}\right) n - O(1) > 1.006n - O(1).$$

Finally, we use Lemma 35 to translate this lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ to $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. ◀

Out next step is to relate the complexities of problems $\mathrm{U}_n \boxplus \mathrm{KW}_g$ and $\mathrm{U}_n \boxplus \mathrm{MUX}_n$.

▶ **Lemma 42.** *There exists $g : \mathbb{B}^n \to \mathbb{B}^n$ such that*

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g) \geq \mathrm{CC}^{hd}(\mathrm{U}_n \boxplus \mathrm{MUX}_n) - O(\log n).$$

The proof is almost identical to the proof of Lemma 31. Note that, in contrast to Lemma 31, the statement of this Lemma does not seem to be trivial.

Proof. Suppose that $\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g) \leq d$ for all $g : \mathbb{B}^n \to \mathbb{B}^n$. Consider the following half-duplex protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. For every $g : \mathbb{B}^n \to \mathbb{B}^n$ let Π_g be the shortest (classical) protocol for $\mathrm{U}_n \boxplus \mathrm{KW}_g$. Alice, who is given x_a, y_a and g_a, follows protocol Π_{g_a} on input (x_a, y_a). Meanwhile Bob, who is given x_b, y_b and g_b, follows protocol Π_{g_b} on input (x_b, y_b). If g_a is different from g_b they might use different protocols, which is fine because we are in the half-duplex communication model.

When Alice reaches some leaf of Π_{g_a} she starts listening until the end of round d. Bob does the same. After d rounds of communication Alice has a candidate i for $(x_a \circ y_a)_i \neq (x_b \circ y_b)_i$, which is a valid output if $g_a = g_b$. Bob has a candidate j for $(x_a \circ y_a)_j \neq (x_b \circ y_b)_j$, that is equal to i if $g_a = g_b$. Now Alice and Bob need to check that indeed $(x_a \circ y_a)_i \neq (x_b \circ y_b)_j$ and $i = j$, which can be done in $O(\log n)$. They output i if both conditions are true and $\perp$ otherwise. The total number of rounds of this half-duplex protocol for $\mathrm{U}_n \boxplus \mathrm{KW}_g$ is $d + O(\log n)$. ◀

Immediately we get the following theorem.

▶ **Theorem 43.** *There exists $g : \mathbb{B}^n \to \mathbb{B}^n$ such that*

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g) \geq 1.006n.$$

To improve this bound we will have to look deeper into the protocol structure and use the fact that it is partially half-duplex.

▶ **Definition 44.** *A half-duplex protocol for* $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ *is called* partially half-duplex *if it has the following property: whenever Alice and Bob are given the same function they are not allowed to perform non-classical communication. In other words, in a partially half-duplex protocol Alice and Bob never send or listen simultaneously if* $g_a = g_b$.

We are going to need the following analogue of Lemma 42.

▶ **Lemma 45.** *There exists* $g : \mathbb{B}^n \to \mathbb{B}^n$ *such that*

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g) \geq \mathrm{CC}^{phd}(\mathrm{U}_n \boxplus \mathrm{MUX}_n) - O(\log n).$$

Proof. Note that the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ in the proof of Lemma 42 is partially half-duplex (i.e., it has only classical rounds unless $g_a \neq g_b$). The rest of the proof is identical to the proof of Lemma 42. ◀

Next Lemma proves a lower bound on the partially half-duplex complexity of $\mathrm{U}_n \boxplus \mathrm{MUX}_n$.

▶ **Lemma 46.** *Any partially half-duplex protocol for* $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ *has depth at least* $\frac{3}{2}n - O(\log n)$.

Together with Lemma 42, this lemma immediately implies our main result that the XOR-KRW holds for a composition of the universal relation with the KW-game for some function.

▶ **Theorem 21.** *For all* $n \in \mathbb{N}$, *there exists* $g : \mathbb{B}^n \to \mathbb{B}^n$ *such that*

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}_g) \geq 1.5n - O(\log n).$$

Once again we are going to split the proof of Lemma 46 in two parts. In the first part, instead of finding one large subrectangle we will find a collection of subrectangles. All the nodes corresponding to these subrectangles will have equal *partial transcripts*. In the classical communication model, *a partial transcript* of a node of the protocol is a bit string consisting of all the messages that are sent on the path from the root to this node. For a partially half-duplex protocol we can also define a partial transcript of a node in the same way if all the preceding communication of the node is classical. An important difference is that in the classical model a partial transcript uniquely defines a node. In the half-duplex model the same partial transcript of length d can correspond to at most 2^d nodes of the protocol, e.g. a partial transcript "00" can correspond to 4 different nodes: a node where both messages were sent by Alice, a node where both messages were send by Bob, and two nodes where both players sent messages in different order.

▶ **Lemma 47.** *For any partially half-duplex protocol* Π *for* $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$, *there exists a subset of inputs* $S \subset \mathcal{X}$, $|S| \geq 8NN!$, *and a string* $T \in \mathbb{B}^{n-3}$, *such that if Alice and Bob are given the same input from* S *then the transcript of the first* $n-3$ *rounds is equal to* T.

Proof. Let $D = \{((g,x,y),(g,x,y)) \mid (g,x,y) \in \mathcal{X}\}$ be a subset of inputs where Alice's and Bob's inputs are identical. First, we need to notice that if Alice and Bob are given inputs from D, then they perform only classical communication. Consider the first $n-3$ rounds of communication. There are at most 2^{n-3} different transcripts of length $n-3$, so there is a transcript T that corresponds to at least $|D|/2^{n-3} = 8NN!$ inputs from D. Let S be the set of all these inputs. ◀

The difference from what we have seen before is that the set S constructed here is not consolidated in a single node of the protocol. All the elements of S have the same transcript of the first $n-3$ rounds but these transcripts do not include the information who sends each of the messages, so in fact the same transcripts can correspond to different nodes of the protocol. Note that any two inputs from S with the same function g necessarily belong to the same node of the protocol as all the rounds are classical.

Now we will prove Lemma 46 by showing that if $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ has a short protocol then we can use it to solve either equality or non-equality more efficiently than it is possible using a dichotomy similar to one from the proof of Theorem 23.

Proof of Lemma 46. Suppose that Π is a partially half-duplex protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ of depth d. Let S be the set provided by Lemma 47. Let $S' = S \setminus \{(g,x,y) \mid \deg_{G_S}(g) < 4N\}$, so $|S'| > 4NN!$. Let $G_{S'} = (U_{S'}, V_{S'}, E_{S'})$ be a domain graph of S'. The minimal degree of the vertices in $U_{S'}$ is at least $4N$.

Suppose that there is $g \in \mathcal{P}$ and $y \in \mathbb{B}^n$ such that $\left|\{x \mid (g,(x,y)) \in E_{S'}\}\right| > \sqrt{N}$. Let $S_{g,y} = \{(g,x,y) \mid (g,(x,y)) \in E_{S'}\}$. We can extract from Π a classical protocol of depth at most $d-n-3$ that solves $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S_{g,y} \times S_{g,y}$. This follows from the fact that Π s partially half-duplex, so it has only classical rounds for inputs from $S_{g,y} \times S_{g,y}$. To solve $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S_{g,y} \times S_{g,y}$ the players would have to solve the equality problem for $W_{g,y} = \{x \mid (g,(x,y)) \in E_{S'}\}$ that requires at least $\log|W_{g,u}| \geq \log(\sqrt{N}) = n/2$. The reduction is the same as in the proof of Theorem 23. Thus, we have $d \geq 1.5n - 3$.

Otherwise we apply Lemma 39 to construct a set H of size at least $2^{\Omega(2^{n/2})}$. Then the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ on $S' \times S'$ can be used to non-deterministically solve NEQ_H with additive overhead of $O(\log n)$. The reduction from NEQ_H to $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ is similar to the one we have seen in the proof of Theorem 23 with just a few twists.

Let's first see what are the necessary and sufficient conditions for $g_a, g_b \in H$ to be not equal. Let $R_{g_a,g_b} = \{((g_a,x_a,y_a),(g_b,x_b,y_b)) \in S' \times S' \mid x_a \oplus g_a(y_a) \neq x_b \oplus g_b(y_b)\}$.

- On elements of $D = \{((g,x,y),(g,x,y)) \mid (g,x,y) \in S'\}$ that contain g_a and g_b, the protocol Π performs differently during the first $n-3$ rounds. The partial transcript T of the first $n-3$ rounds of Π on elements of D is fixed by Lemma 47, but it does not include an information about who sends each message, so the same transcript can be produced by different rounds. Such a difference can only exists if $g_a \neq g_b$ – for every fixed $g_a = g_b$ the protocol has only classical rounds, and hence a partial transcript uniquely defines who sends in each round.
- The protocol Π performs a non-classical round on some input from R_{g_a,g_b}. If $g_a = g_b$ then Π can only perform classical rounds by the definition of partially half-duplex communication.
- Π performs classically on some input from R_{g_a,g_b} and returns $\perp$.

We can argue that one of this conditions is satisfied iff $g_a \neq g_b$. Indeed, suppose that $g_a \neq g_b$. If the first or the second condition is satisfied we are done, so let's assume that it is not. The first $n-3$ rounds of Π on inputs from R_{g_a,g_b} are already known, so we can skip them and only consider the rounds of Π after that. We also know that all the next rounds are going to be classical. By construction of H there exists x, y, such that (g_a,x,y) and (g_b,x,y) belong to S', and also $x \oplus g_a(y) \neq x \oplus g_b(y)$. By the definition of $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ the protocol Π has to output $\perp$, and hence satisfy the third condition.

Now suppose that $g_a = g_b$. Then neither of the conditions could be satisfied. The first condition fails as in this case a partial transcript uniquely defines who sends in each round. The second condition fails by the definition of partially half-duplex protocol. The third one fails by definition of the $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$.

Now we can use this property to solve NEQ_H. Alice and Bob guess which of the condition is satisfied, guess a proof of it, and then verify it.

- To prove the first condition the players guess the difference in the first $n-3$ rounds. Verification requires only $\log n$ bits of communication.
- For the second condition the players guess a number $t \in [d-n+3]$, a string $s \in \mathbb{B}^t$, a number $k \in [n]$, and bits b, p. Then they verify that there exist pairs (x_a, y_a) and (x_b, y_b) such that:
  - $p = (x_a \oplus g_a(y_a))_k \neq (x_b \oplus g_b(y_b))_k = 1-p$,
  - both players are consisted with s being an extension of the partial transcript T on inputs $((g_a, x_a, y_a), (g_b, x_b, y_b))$, meaning that if a player wants to send a bit in some round, this bit is equal to corresponding bit in s,
  - in the next round after the rounds described in s, the protocol Π performs a non-classical round: either both send (in case $b=1$) or both receive (in case $b=0$).

  All together the size of the witness in this case is $d-n+O(\log n)$.
- For the third condition the players guess a string $s \in \mathbb{B}^{d-n+3}$, a number $i \in [n]$, and a bit p. Then they verify that there exist pairs (x_a, y_a) and (x_b, y_b) such that:
  - $p = (x_a \oplus g_a(y_a))_k \neq (x_b \oplus g_b(y_b))_k = 1-p$,
  - both players are consisted with s being an extension of the partial transcript T on inputs $((g_a, x_a, y_a), (g_b, x_b, y_b))$, meaning that if a player wants to send a bit in some round, this bit is equal to corresponding bit in s,
  - the transcript ends in a leaf marked labeled $\perp$.

  All together the size of the witness in this case is $d-n+O(\log n)$.

This reduction shows that NEQ_H can be non-deterministically solved with a protocol of size $d-n+O(\log n)$. Thus, the depth of the protocol for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ is at least

$$\begin{aligned} n + \mathrm{NCC}(\mathrm{NEQ}_H) - O(\log n) &\geq n + \log\log|H| - O(\log n) \\ &\geq n + \log\sqrt{N} - O(\log\log(N)) = 1.5n - O(\log n). \end{aligned}$$

Finally, we use Lemma 35 to translate this lower bound for $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ to $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. ◀

5 Conclusion

In this paper we presented a lower bound for $\mathrm{U}_n \boxplus \mathrm{KW}_g$ for some function g. Our result complements the result from [5] where a lower bound for $\mathrm{KW}_g \diamond \mathrm{U}_n$ was shown. It remains to understand if the techniques from these two papers can be forced to work in harmony. We are very optimistic about it: the structure of our proof reminds of the first results regarding $\mathrm{U}_m \diamond \mathrm{U}_n$ from [4]: we maintain the symmetry for as long as possible and then show that some of the hardness still remains in the problem. The proof from [5] shows how to substitute the symmetry with some hardness measure and hopefully the same magic can be applied to this instance.

Open questions

1. Is there a generic ways to convert lower bounds for classical communication into half-duplex and partially half-duplex?
2. Is there another proof of the results from this paper, that doesn't rely on non-classical models?

3. Prove lower bound of $2n - o(n)$ for $\mathrm{U}_n \boxplus \mathrm{MUX}_n$ in classical, partially half-duplex or half-duplex model.
4. Prove that for some $f, g : \mathbb{B}^n \to \mathbb{B}^n$, $\mathrm{CC}(\mathrm{KW}_{f \boxplus g}) \geq (1 + \epsilon)n$.

References

1 Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, and Robert Robere. KRW composition theorems via lifting. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 43–49. IEEE, 2020. doi:10.1109/FOCS46700.2020.00013.

2 Yuriy Dementiev, Artur Ignatiev, Vyacheslav Sidelnik, Alexander Smal, and Mikhail Ushakov. New bounds on the half-duplex communication complexity. In *SOFSEM 2021: Theory and Practice of Computer Science - 47th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2021, Bolzano-Bozen, Italy, January 25-29, 2021, Proceedings*, volume 12607 of *Lecture Notes in Computer Science*, pages 233–248. Springer, 2021. doi:10.1007/978-3-030-67731-2_17.

3 Irit Dinur and Or Meir. Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. *Comput. Complex.*, 27(3):375–462, 2018. doi:10.1007/s00037-017-0159-x.

4 Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jirí Sgall. Communication complexity towards lower bounds on circuit depth. *Comput. Complex.*, 10(3):210–246, 2001. doi:10.1007/s00037-001-8195-x.

5 Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: The composition of a function and a universal relation. *SIAM J. Comput.*, 46(1):114–131, 2017. doi:10.1137/15M1018319.

6 Johan Håstad. The shrinkage exponent of de morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998. doi:10.1137/S0097539794261556.

7 Johan Håstad and Avi Wigderson. Composition of the universal relation. In Jin-Yi Cai, editor, *Advances In Computational Complexity Theory, Proceedings of a DIMACS Workshop, New Jersey, USA, December 3-7, 1990*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 119–134. DIMACS/AMS, 1990. URL: http://dimacs.rutgers.edu/Volumes/Vol13.html, doi:10.1090/dimacs/013/07.

8 Kenneth Hoover, Russell Impagliazzo, Ivan Mihajlin, and Alexander Smal. Half-duplex communication complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:89, 2018. URL: https://eccc.weizmann.ac.il/report/2018/089.

9 Kenneth Hoover, Russell Impagliazzo, Ivan Mihajlin, and Alexander V. Smal. Half-duplex communication complexity. In Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao, editors, *29th International Symposium on Algorithms and Computation, ISAAC 2018, December 16-19, 2018, Jiaoxi, Yilan, Taiwan*, volume 123 of *LIPIcs*, pages 10:1–10:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ISAAC.2018.10.

10 Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5(3/4):191–204, 1995. doi:10.1007/BF01206317.

11 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 539–550. ACM, 1988. doi:10.1145/62212.62265.

12 Valeriy Mihailovich Khrapchenko. Complexity of the realization of a linear function in the class of II-circuits. *Mathematical Notes of the Academy of Sciences of the USSR*, 9(1):21–23, 1971.

13 Sajin Koroth and Or Meir. Improved Composition Theorems for Functions and Relations. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*, volume 116 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 48:1–48:18, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.48.

14 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

15 Or Meir. Toward better depth lower bounds: Two results on the multiplexor relation. *Comput. Complex.*, 29(1):4, 2020. doi:10.1007/s00037-020-00194-8.

16 Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

17 Bella Abramovna Subbotovskaya. Realization of linear functions by formulas using ∧, ∨, ¬. In *Doklady Akademii Nauk*, volume 136-3, pages 553–555. Russian Academy of Sciences, 1961.

18 Avishay Tal. Shrinkage of de morgan formulae by spectral techniques. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 551–560. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.65.

A Lower bound for a block-composition of a universal relation and a function

▶ **Definition 48.** *Let $g : \{0,1\}^m \to \{0,1\}$ be a Boolean function.* The block-composition of a universal relation with a function $\mathrm{U} \diamond g$ *is the following relation:*

$$\mathrm{U}_n \diamond g_m = \{(A, B, (i,j)) \mid A[i,j] \neq B[i,j]\} \cup \{(A, B, \perp) \mid \forall i \in [n] : g(A[i]) = g(B[i])\},$$

where $A, B \in \{0,1\}^{n \times m}$.

▶ **Theorem 49.** *There exists $f : \{0,1\}^n \to \{0,1\}$, such that:* $\mathrm{CC}(\mathrm{U}_n \diamond f_n) \geq 1.5n - O(\log n)$.

In order to prove this, we need to argue that the result in Theorem 21 also holds for the following version of $\mathrm{U}_n \boxplus \mathrm{KW}_g$.

▶ **Definition 50.** *Let $g : \mathbb{B}^n \to \mathbb{B}^n$. In a communication game $\mathrm{U}_n \boxplus \mathrm{KW}'_g$: Alice is given $x_a, y_a \in \mathbb{B}^n$ and Bob is given $x_b, y_b \in \mathbb{B}_n$. Their goal is to find $i \in [2n]$ such that $(x_a \circ y_a)_i \neq (x_b \circ y_b)_i$. If $x_a \oplus g(y_a) = x_b \oplus g(y_b)$ they can output $\perp$.*

This problem relates to $\mathrm{U}_n \boxplus \mathrm{KW}_g$ as $\mathrm{U}_n \boxplus \mathrm{MUX}'_n$ relates to $\mathrm{U}_n \boxplus \mathrm{MUX}_n$. If fact, if we do not use Lemma 35 in the proof of Theorem 21 then we prove the following lower bound.

▶ **Theorem 51.** *For all $n \in \mathbb{N}$, there exists $g : \mathbb{B}^n \to \mathbb{B}^n$ such that*

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}'_g) \geq 1.5n - O(\log n).$$

Now we are ready to prove the lower bound for the block-composition.

Proof of Theorem 49. We prove this Theorem by a reduction from $\mathrm{U}_n \boxplus \mathrm{KW}'_g$. Let $g : \{0,1\}^n \to \{0,1\}^n$ be such that

$$\mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}'_g) \geq 1.5n - O(\log n).$$

Let $f : \{0,1\}^{n+\log n+1} \to \{0,1\}$ be a function that treats it's input x as a n-bit string x', a number $i_x \in [n]$ and a bit b_x. In these terms

$$f(x) = b_x \oplus g(x')[i_x].$$

Given x_a, y_a Alice constructs a matrix A as follows: in the i-th row she puts y_a as the first n bits, then she puts i in binary as the next $\log n$ bits and she adds $x_a[i]$ as the last bit. Then she adds $\log n + 1$ rows with zeroes. As a result she gets a matrix $A \in \{0,1\}^{l \times l}$ for $l = n + \log n + 1$. Bob does the symmetric thing and gets a matrix B. Now it is not hard to see that for every $i \in [n]$, $(x_a \oplus g(y_a))[i] = f(A[i])$. Thus, if we have solved $\mathrm{U}_l \diamond f_l$ on (A, B) and the result was $\perp$ then $\perp$ is the correct answer for $\mathrm{U}_n \boxplus \mathrm{KW}'_g$. Now suppose that $A[i,j] \neq B[i,j]$. If $i = n + \log n + 1$ then $x_a[j] \neq x_b[j]$. If $i \leq n$ then $y_a[i] \neq y_b[i]$. That gives us

$$\mathrm{CC}(\mathrm{U}_l \diamond f_l) \geq \mathrm{CC}(\mathrm{U}_n \boxplus \mathrm{KW}'_g) \geq 1.5n - O(\log n),$$

and hence

$$\mathrm{CC}(\mathrm{U}_n \diamond f_n) \geq 1.5n - O(\log n). \qquad \blacktriangleleft$$

B Proof of Theorem 9

▶ **Theorem 9.** *For any $m, n \in \mathbb{N}$ with $n \geq 6 \log m$, and any non-constant function $f : \{0,1\}^m \to \{0,1\}$,*

$$\mathrm{CC}(\mathrm{KW}_{f \diamond \mathsf{M}_n}) \geq \log L(f) + n - O(\log^* n).$$

Proof. First of all, we show that for any non-constant function $f : \mathbb{B}^m \to \mathbb{B}$,

$$\mathrm{CC}(\mathrm{KW}_{f \diamond \mathsf{M}_n}) \geq \mathrm{CC}(\mathrm{KW}_f \diamond \mathrm{U}_n) - O(\log n)$$

by reducing $\mathrm{KW}_f \diamond \mathrm{U}_n$ to $\mathrm{KW}_{f \diamond \mathsf{M}_n}$, and then we apply the lower bound on $\mathrm{CC}(\mathrm{KW}_f \diamond \mathrm{U}_n)$ proved in [5, 13].

Consider a communication game $\mathrm{KW}_f \diamond \mathrm{U}_n$: Alice and Bob are given (x, X) and (y, Y) respectively, where $x \in f^{-1}(0)$, $y \in f^{-1}(1)$, $X, Y \in \mathbb{B}^{m \times n}$, and they want to find a position where X and Y differ. The following construction describes a reduction from this game to $\mathrm{KW}_{f \diamond \mathsf{M}_n}$. Given x and X Alice defines functions $s_1, \dots, s_n$:

$$s_i(r) = \begin{cases} x[i], & r = X_i \\ 0, & \text{otherwise}, \end{cases}$$

where X_i is the i-th row of X. Given y and Y Bob defines functions $t_1, \dots, t_n$ in the same way. The reduction guarantees that

$$(f \diamond \mathsf{M}_n)(s_1, X_1, \dots, s_m, X_m) = 0 \quad \text{and} \quad (f \diamond \mathsf{M}_n)(t_1, Y_1, \dots, t_m, Y_m) = 1,$$

and hence the players can simulate the KW game for $f \diamond \mathsf{M}_n$ on these inputs. There are two possible outcomes of such a game: Alice and Bob find a difference between either some rows X_i and Y_i or some functions s_i and t_i.

In the first case, they are done – the players have found a difference between X and Y. In the second case, Alice and Bob find a position where two functions s_i and t_i differ for some $i \in [m]$, i.e., at the end of the protocol they both know some r such that $s_i(r) \neq t_i(r)$. Then either $r = X_i$ or $r = Y_i$. Using two extra bits of communication Alice and Bob can find out which of these two cases applies. If $r = X_i \neq Y_i$ then Bob can find a position where $r = X_i$ and Y_i differ, and send it to Alice using $\log n$ bits. The other case is symmetric.

The reduction shows that

$$\mathrm{CC}(\mathrm{KW}_f \diamond \mathrm{U}_n) \leq \mathrm{CC}(\mathrm{KW}_{f \diamond \mathsf{M}_n}) + O(\log n).$$

To complete the proof we use the following bound from [13]:

$$\mathrm{CC}(\mathrm{KW}_f \diamond \mathrm{U}_n) \geq \log \mathrm{L}(f) + n - O(\log^* n). \qquad \blacktriangleleft$$

Fourier Growth of Parity Decision Trees

Uma Girish
Princeton University, NJ, USA

Avishay Tal
University of California at Berkeley, CA, USA

Kewen Wu
University of California at Berkeley, CA, USA

Abstract

We prove that for every parity decision tree of depth d on n variables, the sum of absolute values of Fourier coefficients at level ℓ is at most $d^{\ell/2} \cdot O(\ell \cdot \log(n))^{\ell}$. Our result is nearly tight for small values of ℓ and extends a previous Fourier bound for standard decision trees by Sherstov, Storozhenko, and Wu (STOC, 2021).

As an application of our Fourier bounds, using the results of Bansal and Sinha (STOC, 2021), we show that the k-fold Forrelation problem has (randomized) parity decision tree complexity $\widetilde{\Omega}\left(n^{1-1/k}\right)$, while having quantum query complexity $\lceil k/2 \rceil$.

Our proof follows a random-walk approach, analyzing the contribution of a random path in the decision tree to the level-ℓ Fourier expression. To carry the argument, we apply a careful cleanup procedure to the parity decision tree, ensuring that the value of the random walk is bounded with high probability. We observe that step sizes for the level-ℓ walks can be computed by the intermediate values of level $\le \ell - 1$ walks, which calls for an inductive argument. Our approach differs from previous proofs of Tal (FOCS, 2020) and Sherstov, Storozhenko, and Wu (STOC, 2021) that relied on decompositions of the tree. In particular, for the special case of standard decision trees we view our proof as slightly simpler and more intuitive.

In addition, we prove a similar bound for noisy decision trees of cost at most d – a model that was recently introduced by Ben-David and Blais (FOCS, 2020).

2012 ACM Subject Classification Theory of computation → Oracles and decision trees; Theory of computation → Communication complexity; Theory of computation → Quantum complexity theory

Keywords and phrases Fourier analysis of Boolean functions, noisy decision tree, parity decision tree, query complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.39

Funding *Uma Girish*: Research supported by the Simons Collaboration on Algorithms and Geometry, by a Simons Investigator Award and by the National Science Foundation grant No. CCF-1714779.

Acknowledgements We thank anonymous reviewers for helpful comments.

1 Introduction

A common theme in the analysis of Boolean functions is proving structural results on classes of Boolean devices (e.g., decision trees, bounded-depth circuits) and then exploiting the structure to: (i) devise pseudorandom generators fooling these devices, (ii) prove lower bounds, showing that some explicit function cannot be computed by such Boolean devices of certain size, or (iii) design learning algorithms for the class of Boolean devices in either the membership-query model or the random-samples model. Such structural results can involve properties of the Fourier spectrum of Boolean functions associated with Boolean devices, like concentration on low-degree terms or concentration on a few terms (i.e., "approximate sparsity").

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 39; pp. 39:1–39:36
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this work, we investigate the Fourier spectrum of parity decision trees. A parity decision tree (PDT) is an extension of the standard decision tree model. A PDT is a binary tree where each internal node is marked by a linear function (modulo 2) on the input variables $(x_1, \dots, x_n)$, with two outgoing edges marked with 0 and 1, and each leaf is marked with either 0 or 1. A PDT naturally describes a computational model: on input $x = (x_1, \dots, x_n)$, start at the root and at each step query the linear function specified by the current node on the input x and continue on the edge marked with the value of the linear function evaluated on x. Finally, when reaching a leaf, output the value specified in the leaf. PDTs naturally generalize standard decision trees that can only query the value of a single input bit in each internal node.

PDTs were introduced in the seminal paper of Kushilevitz and Mansour [21]. Aligned with the aforementioned theme, Kushilevitz and Mansour proved a structural result for PDTs and used it to design learning algorithms for PDTs. They showed that every PDT of size s computing a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ has

$$L_1(f) \triangleq \sum_{S \subseteq [n]} \left|\widehat{f}(S)\right| \leq s,$$

where $\widehat{f}(S)$ are the Fourier coefficients of f (see Subsection 2.1 for a precise definition). Then, they gave a learning algorithm in the membership-query model, running in time $\mathsf{poly}(t, n)$ that can learn any function f with $L_1(f) \leq t$. Combining the two results together, they obtained a $\mathsf{poly}(s, n)$-time algorithm for learning PDTs of size s.

Parity decision trees were also studied in relation to communication complexity and the log-rank conjecture [26, 39, 40, 38, 35, 31, 13, 20, 18, 33, 23]. Suppose Alice gets input $x \in \{0,1\}^n$, Bob gets input $y \in \{0,1\}^n$ and they want to compute some function $f(x, y)$. When f is an XOR-function, namely $f(x, y) = g(x \oplus y)$ for some $g : \{0,1\}^n \to \{0,1\}$, then any PDT for g of depth d can be translated into a communication protocol for f at cost $2d$: Alice and Bob simply traverse the PDT together, both exchanging the parity of their part of the input to simulate each query in the PDT. With this view, parity decision trees can be thought of as special cases of communication protocols for XOR functions. A surprising result by Hatami, Hosseini, and Lovett [18], shows that this is not far from the optimal strategy for XOR functions. Namely, if the communication cost for computing f is c, then the parity decision tree complexity of g is at most $\mathsf{poly}(c)$. Due to this connection, the log-rank conjecture for XOR-functions reduces to the question of whether Boolean functions with at most s non-zero Fourier coefficients can be computed by PDTs of depth $\mathsf{polylog}(s)$ [26, 39]. The best known upper bound is that such functions can be computed by PDTs of depth $O(\sqrt{s})$ [38] (or even non-adaptive PDTs of depth $\widetilde{O}(\sqrt{s})$ [33]).

While having small $L_1(f)$ norm implies learning algorithms and also simple pseudorandom generators fooling f [27], this property can be quite restrictive. In particular, very simple functions (e.g., the Tribes function) have $L_1(f)$ exponential in n. Such examples motivated Reingold, Steinke, and Vadhan [32] to study a more refined notion measuring for a given level ℓ, the sum of absolute values of Fourier coefficients of sets S of size exactly ℓ, i.e, to study

$$L_{1,\ell}(f) \triangleq \sum_{S \subseteq [n] : |S| = \ell} \left|\widehat{f}(S)\right|.$$

In particular, for $\ell = 1$, the measure $L_{1,1}(f)$ is tightly related to the total influence of f (and equals to it if f is monotone). The idea behind this more refined notion is that Fourier coefficients of different levels behave differently under standard manipulations to the function like random restrictions or noise operators. For example, when applying a noise operator with parameter γ, level-ℓ coefficients are multiplied by γ^ℓ. This motivates to establish a bound of the form $L_{1,\ell}(f) \leq t^\ell$ for some parameter t and all $\ell = 1, \ldots, n$. If f satisfies such a bound, we say that $f \in \mathcal{L}_1(t)$.[1]

Reingold, Steinke, and Vadhan [32] showed that for read-once permutation branching programs of width w, while $L_1(f)$ could be exponential in n (even for $w = 3$), it nevertheless holds that $L_{1,\ell}(f) \leq (2w^2)^\ell$ for all $\ell = 1, \ldots, n$. Then, they constructed a pseudorandom generator that fools any class of read-once branching programs for which $f \in \mathcal{L}_1(t)$ using only $t \cdot \mathsf{polylog}(n)$ random bits. This result was significantly generalized to a pseudorandom generator that fools any class of functions $f \in \mathcal{L}_1(t)$ using only $t^2 \cdot \mathsf{polylog}(n)$ random bits [9]. Further results established pseudorandom generators assuming $L_{1,\ell}$ bounds only on the first few levels [11, 8].

It turns out that read-once permutation branching programs are just one example of many well-studied Boolean devices with non-trivial $L_{1,\ell}$ bounds. The following classes of Boolean functions are other examples:

1. Width-w CNF and width-w DNF formulae are in $\mathcal{L}_1(O(w))$ [24].
2. AC^0 circuits of size s and depth d are in $\mathcal{L}_1\left(O(\log(s))^{d-1}\right)$ [36].
3. Boolean functions with max-sensitivity at most s are in $\mathcal{L}_1(O(s))$ [17]
4. Read-once branching programs of width w are in $\mathcal{L}_1\left(O(\log(n))^w\right)$[11]
5. Deterministic and randomized decision trees of depth d are in $\mathcal{L}_1\left(O\left(\sqrt{d \log(n)}\right)\right)$ [37, 34].
6. If $f(x, y)$ is a function computed by communication protocol exchanging at most c bits, then $h(z) = \mathbb{E}_x[f(x, x \oplus z)]$ satisfies $h \in \mathcal{L}_1(O(c))$ [15, 16].
7. Polynomials f over $\mathsf{GF}(2)$ of degree d have $L_{1,\ell}(f) \leq \left(2^{3d} \cdot \ell\right)^\ell$ [9].
8. Product tests, i.e., the XOR of multiple Boolean functions operating on disjoint sets of at most m bits each, are in $\mathcal{L}_1(O(m))$ [22].

We remark that Items 1, 2, 4, 5 and 8 are essentially tight, Item 3 can be potentially improved polynomially [28, 30], Item 6 can be potentially improved quadratically [15] and Item 7 can be potentially improved exponentially [10]. Indeed, improving Item 7 exponentially would imply that $\mathsf{AC}^0[\oplus]$ in $\mathcal{L}_1(\mathsf{polylog}(n))$ and would give the first poly-logarithmic pseudorandom generators for this well-studied class of Boolean circuits [10].

The most relevant result to our work is the recent tight bounds on the $L_{1,\ell}$ of decision trees of depth d. Sherstov, Storozhenko and Wu [34] recently proved that for any randomized decision tree of depth d computing a function f, it holds that $L_{1,\ell}(f) \leq \sqrt{\binom{d}{\ell} \cdot O(\log(n))^{\ell-1}}$. Their bound is nearly tight (see [37, Section 7] and [29, Chapter 5.3] for tightness examples). One motivation for showing such a bound for decision trees is that it demonstrates a stark difference between quantum algorithms making few queries and randomized algorithms making a few queries. Indeed, the Fourier spectrum associated with quantum query algorithms making a few queries can be far from being approximately sparse (in the sense that its $L_{1,\ell}$ is quite large). Based on that difference, both [34] and [2] showed that there are partial functions, either k-fold Forrelation or k-fold Rorrelation, that can be correctly computed with probability at least $1/2 + \Omega(1)$ by quantum algorithms making $\lceil k/2 \rceil$ queries, but require $\widetilde{\Omega}\left(n^{1-1/k}\right)$ queries for any randomized algorithm. Moreover, due to the result of Aaronson and Ambainis [1] this is the largest possible separation between the two models.

[1] Note that if $f \in \mathcal{L}_1(t)$ then after applying noise operator with $\gamma = 1/(2t)$, the noisy-version of f has total L_1-norm at most $O(1)$ which makes it is quite easy to fool using small-biased distributions [27].

Indeed, as suggested in [37], one can show that any function with sufficiently good bounds on its $L_{1,\ell}$, for all $\ell = 1, \dots, n$, cannot solve the k-fold Rorrelation, and such bounds were obtained by [34] for randomized decision trees of depth $n^{1-1/k}/\mathsf{polylog}(n)$. Independently, Bansal and Sinha obtained the same separation but only relying on the $L_{1,\ell}$ bounds for $\ell \in \{k, k+1, \dots, k^2\}$. With this additional flexibility, they were able to obtain their separation for the simpler and explicit function called k-fold Forrelation.

For parity decision trees, the work of Blais, Tan, and Wan [4] established a tight bound of $O\left(\sqrt{d}\right)$ on the first level $\ell = 1$. To the best of our knowledge, bounds on higher levels were not considered previously in the literature (in fact, even for standard decision trees, such bounds were not considered prior to [37]).

1.1 Our Results

We prove level-ℓ bounds for any parity decision tree of depth d.

▶ **Theorem 1** (Informal). *Let $\mathcal{T}$ be a depth-d parity decision tree on n variables. Then the sum of absolute Fourier coefficients at level ℓ is bounded by $d^{\ell/2} \cdot O(\ell \cdot \log(n))^{\ell}$.*

See Theorem 32 and Theorem 39 for a precise statement taking into account the probability that $\mathcal{T}$ accepts a uniformly random input. Theorem 1 extends the result of [34] from standard decision trees to parity decision trees at the cost of an $(\ell \cdot \log(n))^{O(\ell)}$ multiplicative factor. We remark that even for standard decision tree there is a lower bound of $L_{1,\ell}(f) \geq \sqrt{\binom{d}{\ell} \cdot (\log(n))^{\ell-1}}$ [37, Section 7] for constant ℓ and $L_{1,\ell}(f) \geq \frac{1}{\mathsf{poly}(\ell)} \cdot \sqrt{\binom{d}{\ell}}$ for all ℓ [29, Chapter 5.3]. Thus, our bounds are tight up to $\mathsf{polylog}(n)$ factors for constant ℓ, and they deteriorate as ℓ grows. Nevertheless, our main application relies on the bounds for small values of ℓ (constant or at most $\log^2 n$).

Noisy Decision Trees

We also investigate the Fourier spectrum of noisy decision trees. Noisy decision trees are a different generalization of the standard model; here in each internal node v we query a noisy version of an input bit, that equals the true bit with probability $(1 + \gamma_v)/2$. Any such query costs γ_v^2. We say that a noisy decision tree has cost at most d if the total cost in any root-to-leaf path is at most d. Recent work studied this model and established connections to the question of how randomized decision tree complexity behaves under composition [3].

We prove level-ℓ bounds for any noisy decision tree of cost at most d. See Theorem 42 for a precise statement.

▶ **Theorem 2** (Informal). *Let $\mathcal{T}$ be a noisy decision tree of cost at most d on n variables. Then the sum of absolute Fourier coefficients at level ℓ is bounded by $O(d)^{\ell/2} \cdot (\ell \cdot \log(n))^{(\ell-1)/2}$.*

Extension to Randomized Query Models

It is simple to verify that if f is a convex combination of Boolean functions $f_1, \dots, f_m$ each with $L_{1,\ell}(f_i) \leq t_\ell$ then also f satisfy $L_{1,\ell}(f) \leq t_\ell$. Thus, if we take a distribution over PDTs of depth d (resp., noisy decision trees of cost d) we get the same bounds on their $L_{1,\ell}$ as those in Theorem 1 (resp., Theorem 2). This is captured in the following corollary.

▶ **Corollary 3.** *Let $\mathcal{T}$ be a randomized parity decision tree of depth at most d on n variables. Then,*

$$\forall \ell \in [n] : L_{1,\ell}(\mathcal{T}) \leq d^{\ell/2} \cdot O(\ell \cdot \log(n))^{\ell}.$$

Let $\mathcal{T}'$ be a randomized noisy decision tree of cost at most d on n variables. Then,

$$\forall \ell \in [n] : L_{1,\ell}(\mathcal{T}') \leq O(d)^{\ell/2} \cdot (\ell \cdot \log(n))^{(\ell-1)/2}.$$

1.2 Applications

Quantum versus Randomized Query Complexity

Let $k \leq \log(n)$. Bansal and Sinha [2] gave a $\lceil k/2 \rceil$ versus $\widetilde{\Omega}\left(n^{1-1/k}\right)$ separation between the quantum and randomized query complexity of k-fold Forrelation (defined by [1]). For our purposes just think of k-fold Forrelation as a partial Boolean function on n input bits. Our main application is an extension of Bansal and Sinha's lower bound for the model of randomized parity decision trees. This follows from their main technical result and Theorem 1.

▶ **Theorem 4** (Restatement of [2, Theorem 3.2])**.** *Let $f\colon \{0,1\}^n \to [0,1]$ such that f and all its restrictions satisfy $L_{1,\ell}(f) \leq t^{\ell}$ for $\ell = \{k, \ldots, k(k-1)\}$. Let $\delta = 2^{-5k}$. Suppose f is δ-close to the value of k-fold Forrelation of x for all x on which k-fold Forrelation is defined. Then, $t \geq \Omega\left(\frac{n^{(1-1/k)/2}}{k^{15}}\right)$.*

▶ **Corollary 5.** *If $\mathcal{T}$ is a randomized parity decision tree of depth d computing k-fold Forrelation with success probability $\frac{1}{2} + \gamma$, then $d \geq \gamma^2 \cdot \frac{n^{1-1/k}}{\mathsf{poly}(k)\log^2 n}$.*

Proof. We can amplify the success probability of the randomized parity decision tree from $1/2+\gamma$ to $1-2^{-5k}$ by repeating the query algorithm $O(k/\gamma^2)$ times independently and taking majority. This results in a randomized parity decision tree $\mathcal{T}'$ of depth $d' = O(d \cdot k/\gamma^2)$. Now, Corollary 3 gives $L_{1,\ell}(\mathcal{T}') \leq (d')^{\ell/2} \cdot O(\ell \cdot \log(n))^{\ell}$ for all ℓ. In particular, $L_{1,\ell}(\mathcal{T}') \leq t^{\ell}$ for all $\ell \leq k(k-1)$ where $t = O\left(\sqrt{d'} \cdot k(k-1) \cdot \log(n)\right)$. This is also true for any restriction of $\mathcal{T}'$, since fixing variables to constants yields another randomized parity decision tree of depth at most d'. Combining the bounds on $L_{1,\ell}(\mathcal{T}')$ for $\ell \in \{k, \ldots, k(k-1)\}$ with Theorem 4 gives $d' \geq \frac{n^{1-1/k}}{O(k^{34})\cdot\log^2(n)}$ and thus $d \geq \gamma^2 \cdot \frac{n^{1-1/k}}{O(k^{35})\cdot\log^2(n)}$. ◀

For constant k and $\gamma = 2^{-O(k)}$, we get a $\lceil k/2 \rceil$ versus $\widetilde{\Omega}\left(n^{1-1/k}\right)$ separation between the quantum query complexity and the randomized parity query complexity of k-fold Forrelation. We remark that separations in the reverse direction are also known: for the n-bit parity function, the (randomized) parity query complexity is 1 whereas the quantum query complexity is $\Omega(n)$ [25].

Similarly, we can obtain the following corollary for noisy decision trees.

▶ **Corollary 6.** *If $\mathcal{T}$ is a randomized noisy decision tree of cost at most d computing k-fold Forrelation with success probability $\frac{1}{2} + \gamma$, then $d \geq \gamma^2 \cdot \frac{n^{1-1/k}}{\mathsf{poly}(k)\log(n)}$.*

Towards Communication Complexity Lower Bounds

We recall an open question from [15], which, if true, would demonstrate that the randomized communication complexity of the Forrelation problem composed with the XOR gadget is $\widetilde{\Omega}(n^{1/2})$. The *simultaneous* quantum communication complexity of this problem is $\mathsf{polylog}(n)$ and the best known randomized lower bound is $\widetilde{\Omega}(n^{1/4})$ due to [15].

▶ **Conjecture 7.** *Let $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ computed by a deterministic communication protocol of cost at most c. Let $h : \{0,1\}^n \to [0,1]$ defined by $h(z) = \mathbb{E}_x[f(x, x \oplus z)]$. Then, $L_{1,2}(h) \le c \cdot \mathsf{polylog}(n)$.*

We view Theorem 1 as a first step towards this conjecture. Indeed, for communication protocols that follow a parity decision tree strategy according to some tree $\mathcal{T}$, it is simple to verify that $h = \mathcal{T}$ (as functions), and thus $L_{1,2}(h) = L_{1,2}(\mathcal{T}) \le c \cdot \mathsf{polylog}(n)$.

We remark that there is a separation of $\mathsf{polylog}(n)$ versus $\widetilde{\Omega}(n^{1/2})$ between *simultaneous* quantum communication complexity and *two-way* randomized communication complexity due to [14]. We also know a separation of $O(k \log n)$ versus $\widetilde{\Omega}(n^{1-1/k})$ between *two-way* quantum communication complexity and *two-way* randomized communication complexity. This can be obtained by combining the optimal quantum versus classical query complexity separations of [2] and [34] and the query-to-communication lifting theorems [7] using the inner product gadget.

Application to Expander Random Walk

Recently, [12] showed that expander random walks fool symmetric functions and also general functions in $\mathcal{L}_1(t)$. To be more precise, assume $f \in \mathcal{L}_1(t)$. Let G be an expander, with second eigenvalue $\lambda \ll \frac{1}{t^4}$, where half of G's vertices are labeled by 0 and the rest are labeled by 1. Then the expected value of f on bits sampled by an $(m-1)$-step random walk on G is approximately the value it would get on a uniformly random string in $\{0,1\}^m$. Combined with our results, this shows that if f can be computed by low-depth parity decision trees then f can be fooled by the expander random walk.

Fourier Bounds for Small-size Parity Decision Trees

By a simple size-to-depth reduction we obtain Fourier bounds for parity decision trees of bounded size. We defer the simple proof to Appendix A.

▶ **Corollary 8.** *Let $\mathcal{T}$ be a parity decision tree of size at most $s > 1$ on n variables. Then,*

$$\forall \ell \in [n] : L_{1,\ell}(f) \le (\log(s))^{\ell/2} \cdot O(\ell \cdot \log(n))^{1.5\ell}.$$

1.3 Technical Overview

For the rest of the paper we consider Boolean functions as functions from $\{\pm 1\}^n$ to $\{0,1\}$. This is for convenience, since most of our calculations become easier under this representation. Observe that under this view, a parity decision tree queries at each internal node the product $\prod_{i \in S} x_i$ for some $S \subseteq [n]$ and goes left/right depending on whether $\prod_{i \in S} x_i = 1$ or -1.

Let $\ell \in \mathbb{N}_+$. For simplicity of notation, we use $\widetilde{O}_\varepsilon\left(d^m\right)$ to denote $\left(d \cdot \mathsf{polylog}\left(n^\ell/\varepsilon\right)\right)^m$ for $m, n, d \in \mathbb{N}_+$ and $\varepsilon \in (0, 1/2]$. When we omit the subscript ε, it is understood that $\varepsilon = 1$. As per this notation, we show a bound of $\widetilde{O}\left(d^{\ell/2}\right)$ on the level-ℓ Fourier mass of parity decision trees of depth d. We first describe the proof for standard decision trees and then show how to generalize to parity decision trees.

Standard Decision Trees

Let $\mathcal{T}$ be a decision tree and for simplicity, assume that every leaf is of depth d. Let $v_0, \ldots, v_d$ be a random root-to-leaf path in $\mathcal{T}$ and $\boldsymbol{v}^{(0)}, \ldots, \boldsymbol{v}^{(d)} \in \{-1,0,1\}^n$ denote the sequence of partial assignments, i.e., for $j \in [n]$ and $i \in \{0, \ldots, d\}$, let

$$\boldsymbol{v}_j^{(i)} = \begin{cases} 1 & \text{if } x_j \text{ is fixed to 1 before reaching } v_i, \\ -1 & \text{if } x_j \text{ is fixed to } -1 \text{ before reaching } v_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

For $u \in \mathbb{R}^n$, we use u_S to denote $\prod_{j \in S} u_j$. Let $a_S = \mathsf{sgn}\left(\widehat{\mathcal{T}}(S)\right)$ for $|S| = \ell$ and 0 otherwise. Note that

$$\sum_{S:|S|=\ell} \left|\widehat{\mathcal{T}}(S)\right| = \sum_{S:|S|=\ell} a_S \widehat{\mathcal{T}}(S) = \sum_{S:|S|=\ell} a_S \mathop{\mathbb{E}}_{v_d}\left[\mathcal{T}(v_d)\boldsymbol{v}_S^{(d)}\right] = \mathop{\mathbb{E}}_{v_d}\left[\mathcal{T}(v_d) \sum_{S:|S|=\ell} a_S \boldsymbol{v}_S^{(d)}\right]. \tag{2}$$

Thus, to bound $\sum_{S:|S|=\ell} |\widehat{\mathcal{T}}(S)|$ it suffices to show that $\left|\sum_{S:|S|=\ell} a_S \cdot \boldsymbol{v}_S^{(d)}\right|$ is bounded by $\widetilde{O}(d^{\ell/2})$ in expectation. Denote by $X^{(i)} := \sum_{S:|S|=\ell} a_S \cdot \boldsymbol{v}_S^{(i)}$ for $i = 0, 1, \ldots, d$. We write $X^{(d)}$ as a telescoping sum $X^{(d)} = \sum_{i=1}^{d} \left(X^{(i)} - X^{(i-1)}\right)$. To analyze the difference sequence, observe that in the expression

$$X^{(i)} - X^{(i-1)} = \sum_{S:|S|=\ell} a_S \cdot \left(\boldsymbol{v}_S^{(i)} - \boldsymbol{v}_S^{(i-1)}\right),$$

if set S contributes to the sum, then S must include the bit queried at the $(i-1)$-th step of the path. Conditioning on $v_0, \ldots, v_{i-1}$, let x_j be the variable queried in v_{i-1}, then we have

$$X^{(i)} - X^{(i-1)} = \sum_{S:|S|=\ell, j \in S} a_S \cdot \boldsymbol{v}_S^{(i)} = x_j \cdot \left(\sum_{S:|S|=\ell, j\in S} a_S \cdot \boldsymbol{v}_{S\setminus\{j\}}^{(i-1)}\right).$$

Furthermore, we observe that the sum $\sum_{S:|S|=\ell, j \in S} a_S \cdot \boldsymbol{v}_{S\setminus\{j\}}^{(i-1)}$ is determined by v_{i-1}; thus conditioning on $v_0, \ldots, v_{i-1}$ the value of $X^{(i)} - X^{(i-1)}$ is a random coin in $\{\pm 1\}$ multiplied by some fixed integer. In other words, we get that $X^{(0)}, \ldots, X^{(d)}$ is a martingale with varying step sizes.

Recall that Azuma's inequality provides concentration bounds for martingales with bounded step sizes, thus now we need to bound $\left|\sum_{S:|S|=\ell, j\in S} a_S \cdot \boldsymbol{v}_{S\setminus\{j\}}^{(i-1)}\right|$, which is similar to our initial goal. Put differently, we wish to analyze the sum

$$\sum_{S' \subseteq [n]\setminus\{j\}:|S'|=\ell-1} a_{S'\cup\{j\}} \cdot \boldsymbol{v}_{S'}^{(i-1)},$$

which calls for an inductive argument on ℓ. In addition, since we eventually apply a union bound on all steps, we need to show that $\left|\sum_{S'} a_{S'\cup\{j\}} \boldsymbol{v}_{S'}^{(i-1)}\right|$ is bounded with high probability (and not just in expectation).

More generally, to carry an inductive argument we define for any set $T \subseteq [n]$, $|T| \leq \ell$ and any $i \in \{0, \ldots, d\}$, the random variable

$$X_T^{(i)} := \sum_{S \supseteq T:|S|=\ell} a_S \cdot \boldsymbol{v}_{S\setminus T}^{(i)} = \sum_{S' \subseteq \overline{T}:|S'|=\ell-|T|} a_{S'\cup T} \cdot \boldsymbol{v}_{S'}^{(i)}.$$

Note that our initial goal was to bound $\left|X_{\emptyset}^{(d)}\right| = \left|X^{(d)}\right|$, which is analyzed by (reverse) induction on $|T|$ going from larger sets to smaller sets as Lemma 9.

▶ **Lemma 9.** *For all $t \in \{0, \ldots, \ell\}$ and $\varepsilon > 0$, the probability that there exist $i \in \{0, \ldots, d\}$ and $T \subseteq [n]$ of size at least t such that $\left|X_T^{(i)}\right| \geq \widetilde{O}_\varepsilon\left(d^{(\ell-t)/2}\right)$ is at most $\varepsilon \cdot (\ell - t)$.*

The main observation for the proof is that $X_T^{(0)}, X_T^{(1)}, \ldots, X_T^{(d)}$ is a martingale whose difference sequence consists of terms of the form $X_{T'}^{(i-1)}$ where $T \subsetneq T'$. To see this, if we are querying x_j at v_{i-1}, then

$$X_T^{(i)} - X_T^{(i-1)} = \begin{cases} 0 & j \in T, \\ x_j \cdot \left(\sum\limits_{j \notin S \subseteq \overline{T}} a_{S \cup T \cup \{j\}} \cdot \boldsymbol{v}_S^{(i-1)} \right) = x_j \cdot X_{T \cup j}^{(i-1)} & j \notin T. \end{cases}$$

Note that $X_{T \cup j}^{(i-1)}$ depends only on the history until v_{i-1}, and x_j is a uniformly random bit independent of this history, thus $X_T^{(i)}$ is a martingale. The inductive hypothesis implies that with at least $1 - \varepsilon \cdot (\ell - t - 1)$ probability, $\left|X_{T \cup j}^{(i-1)}\right| \leq \widetilde{O}_\varepsilon\left(d^{(\ell-t-1)/2}\right)$ for all T of size t and $j \in [n] \setminus T$. Whenever this happens, Azuma's inequality implies that[2] with probability at least $1 - \varepsilon / \left(d \cdot n^t\right)$, we have

$$\left|X_T^{(i)}\right| \leq 2\sqrt{\log(d \cdot n^t/\varepsilon)} \cdot \sqrt{\sum_{i=1}^{d} \widetilde{O}_\varepsilon\left(d^{\ell-t-1}\right)} = \widetilde{O}_\varepsilon\left(d^{(\ell-t)/2}\right).$$

This, along with a union bound over T of size t and $i \in \{0, \ldots, d\}$ completes the inductive step. The Fourier bound for noisy decision trees can be proved using a similar approach.

Parity Decision Trees

The basic approach is as before. Let $\mathcal{T}$ be a parity decision tree. As in (1), we use v_i and $\boldsymbol{v}^{(i)}$ to denote the random walk and the partial assignments to the variables respectively. We say v_i is *k-clean* if

$$\forall S \subseteq [n], |S| \leq k, \quad \boldsymbol{v}_S^{(i)} = \begin{cases} 1 & \text{if } x_S \text{ is fixed to } 1 \text{ before reaching } v_i, \\ -1 & \text{if } x_S \text{ is fixed to } -1 \text{ before reaching } v_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

For (2) to be true, we need that at least v_d is ℓ-clean. Note that this is not always true,[3] but it is useful as it simplifies the study of high-level Fourier coefficients. To address this issue, we define a *cleanup* process for parity decision trees in which we make additional queries to ensure that certain key nodes are k-clean. We do this by recursively cleaning nodes in a top-down fashion so that for every node v in the original tree $\mathcal{T}$, any node v' in the new tree $\mathcal{T}'$ obtained at the end of the cleanup step for v is k-clean.

The cleanup process is simple to describe: Let $v_1, \ldots, v_d$ be any root-to-leaf path in $\mathcal{T}$. Assume we have completed the cleanup process for $v_1, \ldots, v_{i-1}$. We then query the parity at v_i. While there exists a (minimal) set S violating (3), we pick and query an arbitrary

[2] Technically this is not true, since a martingale after conditioning may not still be a martingale. We handle this by truncating the martingale when a bad event happens instead of conditioning on the good event.

[3] For example, let $S = \{1, 2\}$ and consider the parity decision tree whose only query is $x_1 x_2$. At any leaf, the value of $x_1 x_2$ is fixed, however, the values of x_1 and x_2 are free, hence S violates (3).

coordinate in S. Once (3) is satisfied, we proceed to the cleanup process for v_{i+1}. This process increases the depth by a factor of at most k. We set $k = \Theta(\ell \cdot \log(n))$ and work with the new tree $\mathcal{T}'$ of depth $D \leq k \cdot d$.

Let $v_0, \dots, v_D$ be a random root-to-leaf path in $\mathcal{T}'$ and $I_i, i \in [D]$ be the set of coordinates fixed due to the query at v_{i-1}. Note that this set might be of size larger than 1.[4] It follows from simple linear algebra that $\sum_{i=1}^{D} |I_i| \leq D$. Since v_D is k-clean, (2) holds. Defining $X_T^{(i)}$ exactly as before, our goal is to prove Lemma 9 with D instead of d. The proof is still by induction on $\ell - t$. It turns out that $X_T^{(0)}, X_T^{(1)}, \dots, X_T^{(D)}$ is no longer a martingale; instead, $X_T^{(i)} - X_T^{(i-1)} = Y_i + Z_i$ where

$$Y_i := \sum_{\substack{\emptyset \neq J \subseteq I_i \cap \overline{T} \\ |J| \text{ is even}}} x_J \cdot X_{J \cup T}^{(i-1)} \quad \text{and} \quad Z_i := \sum_{\substack{\emptyset \neq J \subseteq I_i \cap \overline{T} \\ |J| \text{ is odd}}} x_J \cdot X_{J \cup T}^{(i-1)}. \tag{4}$$

and Z_i (resp., Y_i) is an odd (resp., even) polynomial of degree at most ℓ over the newly fixed variables $\{x_j \,|\, j \in I_i\}$. Conditioning on v_{i-1}, every pair of random bits $(x_j, x_{j'})$ from $\{x_j \,|\, j \in I_i\}$ is either identical ($x_j \equiv x_{j'}$) or opposite ($x_j \equiv -x_{j'}$), which means Y_i is a constant and Z_i can be written as $z_i \cdot |Z_i|$ where $|Z_i|$ is a constant and $z_i \sim \{\pm 1\}$.

For now, let us ignore Y_i and assume that we have a martingale $X_T^{(i)}$ such that $X_T^{(i)} - X_T^{(i-1)} = z_i \cdot |Z_i|$, where $z_i \sim \{\pm 1\}$ is a uniformly random bit independent of $z_0, \dots, z_{i-1}$ and $|Z_i|$ depends only on v_{i-1}. Combined with an adaptive version of Azuma's inequality, we only need to show the sum of squares of step sizes $\sum_{i=1}^{D} |Z_i|^2$ is $\widetilde{O}_\varepsilon\left(D^{\ell-t}\right)$ to prove $\left|X_T^{(i)}\right| = \widetilde{O}_\varepsilon\left(D^{(\ell-t)/2}\right)$. By the induction hypothesis, with probability at least $1 - \varepsilon \cdot (\ell - t - 1)$ the coefficients of Z_i are bounded appropriately. Since $\sum_{i=1}^{D} |I_i| \leq D$ and in particular $|I_i| \leq D$, we have

$$|Z_i| \leq \sum_{\text{odd } j \geq 1} \binom{|I_i|}{j} \cdot \max_{|T'| = j+t} \left|X_{T'}^{(i-1)}\right| \leq \sum_{j \geq 1}^{\ell - t} \binom{|I_i|}{j} \cdot \widetilde{O}_\varepsilon\left(D^{(\ell-j-t)/2}\right) = \widetilde{O}_\varepsilon\left(|I_i| \cdot D^{(\ell-t-1)/2}\right)$$

and thus $\sum_{i=1}^{D} |Z_i|^2 \leq D^2 \cdot \widetilde{O}_\varepsilon\left(D^{\ell-t-1}\right)$. This is too loose for our purpose.

We instead try to bound the sum of squares of step sizes *with high probability*. Imagine for now that v_{i-1} is 2-clean.[5] Then, the variables $\{x_j \,|\, j \in I_i\}$ are 2-wise independent conditioning on v_{i-1}. This gives

$$\begin{aligned} \mathbb{E}\left[|Z_i|^2 \,\middle|\, v_{i-1}\right] &\leq \sum_{\text{odd } j \geq 1} \binom{|I_i|}{j} \cdot \max_{|T'| = j+t} \left|X_{T'}^{(i-1)}\right|^2 \\ &\leq \sum_{j \geq 1}^{\ell - t} \binom{|I_i|}{j} \cdot \widetilde{O}_\varepsilon\left(D^{\ell-j-t}\right) = \widetilde{O}_\varepsilon\left(|I_i| \cdot D^{\ell-t-1}\right) \end{aligned}$$

and thus $\mathbb{E}\left[\sum_{i=1}^{D} |Z_i|^2\right] \leq \widetilde{O}_\varepsilon\left(D^{\ell-t}\right)$. To show this bound holds with high probability, we use concentration properties of degree-ℓ polynomials under k-wise independent distributions for $k \gg \ell$.

[4] For example, suppose we query x_1x_2, x_1x_3, x_1x_4 and finally x_1. Then, the last query reveals 4 coordinates.

[5] This assumption immediately implies that $|I_i| \leq 1$ and trivially proves our inequality, however, this type of reasoning doesn't generalize to the case when v_{i-1} is not 2-clean.

In the actual proof, we proceed by conditioning on $C(v_{i-1})$, the nearest ancestor of v_{i-1} that is k-clean, instead of conditioning on v_{i-1}, which allows to remove the assumption that v_{i-1} is 2-clean. This is because the queries within a cleanup step are non-adaptive, thus Z_i depends only on $C(v_{i-1})$ and not on v_{i-1}.

Meanwhile, although $X_T^{(i)}$ is not quite a martingale sequence (due to Y_i) and the step sizes (i.e., $|Z_i|$) are adaptive and not always bounded, we are nonetheless able to prove an adaptive version of Azuma's inequality of the form $\mathbf{Pr}\left[\max_{i\in[D]}\left|X_T^{(i)}\right| \geq \mu + t\cdot\sigma\right] \leq e^{-\Omega(t^2)} + \varepsilon$ provided $\mathbf{Pr}\left[\left(\sum_{i=1}^{D}|Y_i| \leq \mu\right) \wedge \left(\sum_{i=1}^{D}|Z_i|^2 \leq \sigma^2\right)\right] \geq 1-\varepsilon$. Then it suffices to bound $\sum_{i=1}^{D}|Y_i|$ similarly to $\sum_{i=1}^{D}|Z_i|^2$ above.

1.4 Related Work

We remark that our proof for level-ℓ Fourier growth (even when specialized to the case of standard decision trees) differs from the proofs appearing in [37] and [34]. There, the results were based on decompositions of decision trees. We view our martingale approach as natural and intuitive. We wonder if one can obtain the tight results from [34] using this approach. It seems that the main bottleneck is a union bound on events related to all sets $T \subseteq [n]$ of size at most ℓ.

Our bounds for level-1 improve those obtained by [4]. They prove that $L_{1,1}(\mathcal{T}) \leq O(\sqrt{p\cdot d})$ when $p = \mathbf{Pr}_x[\mathcal{T}(x)=1]$, whereas we obtain a bound of

$$L_{1,1}(\mathcal{T}) \leq O\left(p\sqrt{d}\cdot\log(1/p)\right).$$

In particular, our bound is almost quadratically better for small values of p. It remains open whether the bound can be further improved to $O\left(p\sqrt{d\cdot\log(1/p)}\right)$, which is the optimal bound for standard decision trees.

We remark that our cleanup technique is inspired by [4], which used cleanup to prove their level-1 bound. However, our proof strategies and the way we use the cleanup procedure is quite different than that of [4].

Organization

We make formal definitions in Section 2. We state and prove the necessary concentration inequalities in Section 3. We present the cleanup process in Section 4. We present the Fourier bounds for parity decision trees in Section 5 and for noisy decision trees in Section 6.

2 Preliminaries

We use $\log(\cdot)$ to denote the logarithm with base 2. We use $[n]$ to denote $\{1,2,\ldots,n\}$; and $\binom{[n]}{k}$ (resp., $\binom{[n]}{\leq k}$) to denote the set of all size-k (resp., size-at-most-k) sets from $[n]$. If S is a set from universe U, then we write $\overline{S}$ for $U\setminus S$. We use $\mathcal{U}_n$ to denote the uniform distribution over $\{\pm1\}^n$. We use $\mathsf{sgn}(\mathsf{value}) \in \{-1,0,1\}$ to denote the sign of value, i.e., $\mathsf{sgn}(\mathsf{value})$ equals -1 if $\mathsf{value} < 0$, 1 if $\mathsf{value} > 0$, and 0 if $\mathsf{value} = 0$.

We use $\mathbb{F}_2 = \{0,1\}$ to denote the binary field, $\mathsf{Span}\langle\mathsf{vectors}\rangle$ to denote the subspace spanned by $\mathsf{vectors}$ over $\mathbb{F}_2$. For a distribution $\mathcal{D}$ we use $x \sim \mathcal{D}$ to represent that x is a random variable sampled from $\mathcal{D}$. For a finite set $\mathcal{X}$ we use $x \sim \mathcal{X}$ to denote that x is a random variable sampled uniformly from $\mathcal{X}$. We use the standard notion of k-wise independent distribution over $\{\pm1\}^n$.

▶ **Definition 10** (k-wise independence). *A distribution $\mathcal{D}$ over $\{\pm 1\}^n$ is k-wise independent if for $x \sim \mathcal{D}$ and any k-indices $1 \le i_1 < i_2 < \ldots < i_k \le n$, the random variables $(x_{i_1}, \ldots, x_{i_k})$ are uniformly distributed over $\{\pm 1\}^k$.*

2.1 Boolean Functions

Here we recall definitions in the analysis of Boolean functions (see [29] for a detailed introduction). Let $f\colon \{\pm 1\}^n \to \mathbb{R}$ be any Boolean function. For any $p > 0$, the p-norm of f is defined as $\|f\|_p = \left(\mathbb{E}_{x \sim \mathcal{U}_n}\left[|f(x)|^p\right]\right)^{1/p}$. For any subset $S \subseteq [n]$, x_S denotes $\prod_{i \in S} x_i$ (in particular, $x_\emptyset = 1$). It is a well-known fact that we can uniquely represent f as a linear combination of $\{x_S\}_{S \subseteq [n]}$:

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) x_S,$$

where the coefficients $\left\{\widehat{f}(S)\right\}_{S \subseteq [n]}$ are referred to as the *Fourier coefficients* of f and are given by $\widehat{f}(S) = \mathbb{E}_{x \sim \mathcal{U}_n}\left[f(x)x_S\right]$. The above representation expresses f as a multilinear polynomial and is called the Fourier representation of f. We say that f is of degree at most d if its Fourier representation is a polynomial of degree at most d, i.e., if $\widehat{f}(S) = 0$ for all $S \subseteq [n], |S| > d$.

2.2 Parity Decision Trees

Here we formally define parity decision trees (with Boolean outputs).

▶ **Definition 11** (Parity decision tree). *A* parity decision tree *$\mathcal{T}$ is a representation of a Boolean function $f\colon \{\pm 1\}^n \to \{0, 1\}$. It consists of a rooted binary tree in which each internal node v is labeled by a non-empty set $Q_v \subseteq [n]$, the outgoing edges of each internal node are labeled by $+1$ and -1, and the leaves are labeled by 0 and 1.*

On input $x \in \{\pm 1\}^n$, the tree $\mathcal{T}$ constructs a computation path *$\mathcal{P}$ from the root to a leaf. Specifically, when $\mathcal{P}$ reaches an internal node v we say that $\mathcal{T}$* queries *Q_v; then $\mathcal{P}$ follows the outgoing edge labeled by $\prod_{i \in Q_v} x_i$. We require that Q_v is not implied by its ancestors' queries. The output of $\mathcal{T}$ (and hence f) on input x is the label of the leaf reached by the computation path. Conversely, we say x is* consistent with *the path $\mathcal{P}$ if $\mathcal{P}$ is the computation path (possibly ending before reaching a leaf) for x.*

We make a few more remarks on a parity decision tree $\mathcal{T}\colon \{\pm 1\}^n \to \{0, 1\}$.

- A node v in $\mathcal{T}$ can be either an internal node or a leaf, and we use $\mathcal{T}(v) \in \{0, 1\}$ to denote the label on v when v is a leaf. Meanwhile, we use $\mathcal{T}_v$ to denote the sub parity decision tree starting with node v.
- The *depth* of a node is the number of its ancestors (e.g., the root has depth 0) and the depth of $\mathcal{T}$ is the maximum depth over all its leaves.
- We say that two parity decision trees $\mathcal{T}$ and $\mathcal{T}'$ are *equivalent* (denoted by $\mathcal{T} \equiv \mathcal{T}'$) if they compute the same function.

2.3 Noisy Decision Trees

▶ **Definition 12** (Noisy oracle). *A noisy query to a bit $b \in \{\pm 1\}$ with correlation $\gamma \in [-1, 1]$ returns a bit $b' \in \{\pm 1\}$ where*

$$b' = \begin{cases} b & \textit{with probability } (1+\gamma)/2, \\ -b & \textit{with probability } (1-\gamma)/2. \end{cases}$$

The cost of a noisy query with correlation γ is defined to be γ^2.

▶ **Definition 13** (Noisy decision tree). *A* noisy decision tree $\mathcal{T}$ *is a rooted binary tree in which each internal node v is labeled by an index $q_v \in [n]$ and a correlation $\gamma_v \in [-1,1]$. The outgoing edges are labeled by $+1$ and -1 and the leaves are labeled by 0 and 1.*

On input $x \in \{\pm 1\}^n$, the tree $\mathcal{T}$ constructs a computation path $\mathcal{P}$ *from the root to leaf as follows. When $\mathcal{P}$ reaches an internal node v, it makes a noisy query to x_{q_v} with correlation γ_v and follows the edge labeled by the outcome of this noisy query. The output of the tree is defined by sampling a root-to-leaf path and returning the label of the leaf. Since the computation path $\mathcal{P}$ is probabilistic, this is an inherently randomized model of computation. We use $\mathcal{T}(x) \in \{0,1\}$ to denote the (probabilistic) output of $\mathcal{T}$ on input x. We also use $\mathcal{T}(v) \in \{0,1\}$ to denote the label on v when v is a leaf. We do* not *require that the indices q_v queried along a path $\mathcal{P}$ are distinct. The* cost *of any path is the sum of costs of the noisy queries along that path; and the cost of $\mathcal{T}$ is the maximum cost of any root-to-leaf path.*

We remark that for any noisy decision tree $\mathcal{T}$, its Fourier coefficient $\widehat{\mathcal{T}}(S)$ is given by $\mathbb{E}[\mathcal{T}(x)x_S]$ where the expectation is over the randomness of both $x \sim \mathcal{U}_n$ and $\mathcal{T}$.

3 Useful Concentration Inequalities

We describe useful concentration inequalities in this section.

3.1 Low Degree Polynomials

We use the fact that low degree polynomials satisfy strong concentration properties under k-wise independent distributions. We will find the following hypercontractive inequality useful.

▶ **Theorem 14** ([5], see also [29, $(2,q)$-hypercontractivity]). *Let $f\colon \{\pm 1\}^n \to \mathbb{R}$ be a degree-d polynomial. Then for any $q \geq 2$, we have $\|f\|_q \leq (q-1)^{d/2}\|f\|_2$.*

▶ **Lemma 15.** *Let $f\colon \{\pm 1\}^n \to \mathbb{R}$ be a degree-d polynomial. Let $\mathcal{D}$ be a $2k$-wise independent distribution over $\{\pm 1\}^n$, where $k \geq d$. Let $\mu = \mathbb{E}_{x\sim\mathcal{D}}[f(x)]$ and $\sigma^2 = \mathbb{E}_{x\sim\mathcal{D}}\left[(f(x)-\mu)^2\right]$. Then for any $\alpha > 0$ and any integer $1 \leq \ell \leq k/d$, we have*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[(f(x)-\mu)^{2\ell}\right] \leq \sigma^{2\ell}\cdot(2\ell-1)^{d\cdot\ell}.$$

In particular we have

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[|f(x)-\mu| \geq \alpha\cdot\sigma\right] \leq \alpha^2\cdot\left(\frac{2k}{d\cdot\alpha^{2/d}}\right)^k.$$

Proof. Observe that $(f(x)-\mu)^{2\ell}$ is a polynomial of degree at most $2\ell\cdot d \leq 2k$. Thus its expectation under $\mathcal{D}$ is the same as its expectation under the uniform distribution over $\{\pm 1\}^n$. By Theorem 14, we have

$$\|f-\mu\|_{2\ell} \leq (2\ell-1)^{d/2}\|f-\mu\|_2 = \sigma\cdot(2\ell-1)^{d/2}.$$

Hence by Markov's inequality, we have

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[|f(x)-\mu| \geq \alpha\cdot\sigma\right] \leq \frac{\mathbb{E}_{x\sim\mathcal{D}}\left[(f(x)-\mu)^{2\ell}\right]}{(\alpha\cdot\sigma)^{2\ell}} = \frac{\|f-\mu\|_{2\ell}^{2\ell}}{(\alpha\cdot\sigma)^{2\ell}} \leq \frac{(2\ell-1)^{\ell\cdot d}}{\alpha^{2\ell}}.$$

Now we derive the second bound. We only need to focus on the case $\alpha \geq 1$ since otherwise the RHS is at least 1. Then by setting $\ell = \lfloor k/d \rfloor$, we have

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[|f(x)-\mu| \geq \alpha\cdot\sigma\right] \leq \frac{(2\lfloor k/d\rfloor-1)^{\lfloor k/d\rfloor\cdot d}}{\alpha^{2\lfloor k/d\rfloor}} \leq \frac{(2k/d)^k}{\alpha^{2(k/d-1)}} = \alpha^2\cdot\left(\frac{2k}{d\cdot\alpha^{2/d}}\right)^k. \quad ◀$$

3.2 Martingales

We show an adaptive version of Azuma's inequality for martingales. The proof is similar to the inductive proof of the standard Azuma's inequality and thus deferred to Appendix B.

▶ **Lemma 16** (Adaptive Azuma's inequality). *Let $X^{(0)},\dots,X^{(D)}$ be a martingale and $\Delta^{(1)},\dots,\Delta^{(D)}$ be a sequence of magnitudes such that $X^{(0)}=0$ and $X^{(i)}=X^{(i-1)}+\Delta^{(i)}\cdot z^{(i)}$ for $i\in[D]$, where if conditioning on $z^{(1)},\dots,z^{(i-1)}$,*
(1) *$z^{(i)}$ is a mean-zero random variable and $\left|z^{(i)}\right|\le 1$ always holds;*
(2) *$\Delta^{(i)}$ is a fixed value.*
If there exists some constant $U\ge 0$ such that $\sum_{i=1}^{D}\left|\Delta^{(i)}\right|^2\le U$ always holds, then for any $\beta\ge 0$ we have

$$\mathbf{Pr}\left[\max_{i=0,1,\dots,D}\left|X^{(i)}\right|\ge\beta\cdot\sqrt{2U}\right]\le 2\cdot e^{-\beta^2/2}.$$

Next, we generalize Lemma 16 as follows.

▶ **Lemma 17.** *Let $m\ge 1$ be an integer. For each $t\in[m]$, let $X_t^{(0)},\dots,X_t^{(D)}$ be a sequence of random variables and $\Delta_t^{(1)},\dots,\Delta_t^{(D)}$ be a sequence of magnitudes such that $X_t^{(0)}=0$ and $X_t^{(i)}=X_t^{(i-1)}+\Delta_t^{(i)}\cdot z_t^{(i)}+\mu_t^{(i)}$ for $i\in[D]$, where if conditioning on $z_t^{(1)},\dots,z_t^{(i-1)}$,*
(1) *$z_t^{(i)}$ is a mean-zero random variable and $\left|z_t^{(i)}\right|\le 1$ always holds;*
(2) *$\Delta_t^{(i)}$ is a fixed value and $\mu_t^{(i)}$ is a random variable.*
If there exist some constants $U,V\ge 0$ and $\eta\in[0,1]$ such that

$$\mathbf{Pr}\left[\exists t\in[m],\ \left(\sum_{i=1}^{D}\left|\Delta_t^{(i)}\right|^2>U\right)\vee\left(\sum_{i=1}^{D}\left|\mu_t^{(i)}\right|>V\right)\right]\le\eta,$$

then for any $\beta\ge 0$ we have

$$\mathbf{Pr}\left[\exists t\in[m],\ \max_{i=0,1,\dots,D}\left|X_t^{(i)}\right|\ge V+\beta\cdot\sqrt{2U}\right]\le\eta+2m\cdot e^{-\beta^2/2}.$$

Proof. We divide the proof into the following two cases.

Case $\eta=0$. Let $\widehat{X}_t^{(i)}=X_t^{(i)}-\sum_{j=1}^{i}\mu_t^{(j)}$ for each t and i. Then $\left|X_t^{(i)}\right|=\left|\widehat{X}_t^{(i)}+\sum_{j=1}^{i}\mu_t^{(j)}\right|\le V+\left|\widehat{X}_t^{(i)}\right|$. By a union bound, it suffices to show for any fixed t, we have

$$\mathbf{Pr}\left[\max_{i=0,1,\dots,D}\left|\widehat{X}_t^{(i)}\right|\ge\beta\cdot\sqrt{2U}\right]\le 2\cdot e^{-\beta^2/2},$$

which follows from Lemma 16.

Case $\eta\ge 0$. Consider $\widetilde{X}_t^{(0)},\dots,\widetilde{X}_t^{(D)}$ defined by setting $\widetilde{X}_t^{(0)}=0$ and $\widetilde{X}_t^{(i)}=\widetilde{X}_t^{(i-1)}+\widetilde{\Delta}_t^{(i)}\cdot z_t^{(i)}+\widetilde{\mu}_t^{(i)}$, where

$$\widetilde{\Delta}_t^{(i)}=\begin{cases}\Delta_t^{(i)} & \sum_{j=1}^{i}\left|\Delta_t^{(j)}\right|^2\le U,\\ 0 & \text{otherwise,}\end{cases}\quad\text{and}\quad\widetilde{\mu}_t^{(i)}=\begin{cases}\mu_t^{(i)} & \sum_{j=1}^{i}\left|\mu_t^{(j)}\right|\le V,\\ 0 & \text{otherwise.}\end{cases}$$

Then Item (1) and (2) hold for $\left(\widetilde{X}_t^{(i)}\right)_{t,i}$ and $\left(\widetilde{\Delta}_t^{(i)}\right)_{t,i},\left(\widetilde{\mu}_t^{(i)}\right)_{t,i}$.

Note that $\mathbf{Pr}\left[\exists t \in [m], i \in \{0,1\dots,D\}, \widetilde{X}_t^{(i)} \neq X_t^{(i)}\right] \leq \eta$ and $\sum_{i=1}^{D} \left|\widetilde{\Delta}_t^{(i)}\right|^2 \leq U$, $\sum_{i=1}^{D} \left|\widetilde{\mu}_t^{(i)}\right| \leq V$ always. Hence from the previous case, we have

$$\begin{aligned}
&\mathbf{Pr}\left[\exists t \in [m], \max_{i=0,1,\dots,D} \left|X_t^{(i)}\right| \geq V + \beta \cdot \sqrt{2U}\right] \\
&\leq \mathbf{Pr}\left[\exists t \in [m], i \in \{0,1\dots,D\}, \ \widetilde{X}_t^{(i)} \neq X_t^{(i)}\right] \\
&\qquad + \mathbf{Pr}\left[\exists t \in [m], \max_{i=0,1,\dots,D} \left|\widetilde{X}_t^{(i)}\right| \geq V + \beta \cdot \sqrt{2U}\right] \\
&\leq \eta + 2m \cdot e^{-\beta^2/2}.
\end{aligned}$$

◀

4 How to Clean Up Parity Decision Trees

In this section we show how to *clean up* the given parity decision tree to make it easier to analyze.

4.1 *k*-cleanness

It will be useful to identify $\mathbb{F}_2^n$ with $\{\pm 1\}^n$ by $\mathsf{Enc}\colon (x_1,\dots,x_n) \mapsto ((-1)^{x_1},\dots,(-1)^{x_n})$. For a subset $X \subseteq \mathbb{F}_2^n$ we will denote $\mathsf{Enc}(X) = \{\mathsf{Enc}(x) : x \in X\}$. Thus, we may think of Boolean functions also as $f\colon \mathbb{F}_2^n \to \{0,1\}$. We observe that under this representation of the input, a parity decision tree $\mathcal{T} : \mathbb{F}_2^n \to \{0,1\}$ indeed queries parity functions (i.e., linear functions over $\mathbb{F}_2$) of the input bits $x \in \mathbb{F}_2^n$ and decides whether to go left or right based on their outcome. Thus, the set of all possible inputs in $\mathbb{F}_2^n$ that reach a given node in a parity decision tree is an affine subspace of $\mathbb{F}_2^n$.

We introduce some notation.

▶ **Notation 18.** Let $\mathcal{T}\colon \{\pm 1\}^n \to \{0,1\}$ be a parity decision tree and let v be a node in it.

- We use $\mathcal{P}_v \subseteq \{\pm 1\}^n$ to denote the set of all points reaching node v. Note that $\mathcal{P}_v = \mathsf{Enc}(H_v + a)$ where H_v is a linear subspace of $\mathbb{F}_2^n$ of dimension $n - \mathsf{depth}(v)$ and $a \in \mathbb{F}_2^n$.
- For any $S \subseteq [n]$, we define $\widehat{\mathcal{P}_v}(S) = \mathbb{E}_{x \sim \mathcal{P}_v}[x_S]$.
- We use $\mathcal{S}_v$ to denote all fully correlated sets with $\mathcal{P}_v$, i.e., $\mathcal{S}_v = \left\{S \subseteq [n] \,\middle|\, \widehat{\mathcal{P}_v}(S) \in \{\pm 1\}\right\}$. We observe that if $\mathcal{P}_v = \mathsf{Enc}(H_v + a)$, then $\mathcal{S}_v = H_v^{\perp}$. Additionally, if the queries on the path from root to v are $Q_{v_0}, \dots, Q_{v_{i-1}}$, then $\mathcal{S}_v = \mathsf{Span}\langle\{Q_{v_0}, \dots, Q_{v_{i-1}}\}\rangle$.
- If v is an internal node, then define $J(v)$ as the set of newly fixed coordinates after querying Q_v, i.e., $i \in J(v)$ iff $\{i\} \notin \mathcal{S}_v$ but $\{i\} \in \mathsf{Span}\left\langle \mathcal{S}_v \cup \{Q_v\}\right\rangle$.

The following simple fact shows that there is no "somewhat" correlated set.

▶ **Fact 19.** *For any parity decision tree $\mathcal{T}$ and any node v in $\mathcal{T}$, $\widehat{\mathcal{P}_v}(S) \in \{+1, 0, -1\}$ holds for any set S.*

Proof. Since $\mathcal{P}_v = \mathsf{Enc}(H_v + a)$ where $H_v + a$ is an affine subspace, $\mathcal{P}_v$ falls into one of the following 3 cases: (a) all points in $\mathcal{P}_v$ satisfy $\chi_S(x) = 1$, (b) all points satisfy $\chi_S(x) = -1$, (c) exactly half of the points satisfy $\chi_S(x) = 1$. ◀

Let $\mathcal{S} \subseteq \mathbb{F}_2^n$ be a subspace and $S \subseteq [n]$. For simplicity, we write $S \in \mathcal{S}$ iff the indicator vector of S is contained in $\mathcal{S}$. Now we describe the desired property: *k-clean.*

▶ **Definition 20** (k-clean subspace and mess-witness)**.** *Let k be a positive integer. A subspace $\mathcal{S}$ is k-*clean *if for any set $S \in \mathcal{S}$ such that $|S| \le k$, we have that $\{i\} \in \mathcal{S}$ holds for any $i \in S$.*

Moreover, when $\mathcal{S}$ is not k-clean, we say i is a mess-witness *if there exists some $S \ni i, |S| \le k$ such that $S \in \mathcal{S}$ but $\{i\} \notin \mathcal{S}$.*

▶ **Definition 21** (k-clean parity decision tree)**.** *A parity decision tree $\mathcal{T}$ is k-*clean *if the following holds:*

- *For any internal node v, either (a) $\mathcal{S}_v$ is k-clean, or (b) $Q_v = \{i\}$ where i is a mess-witness for $\mathcal{S}_v$. Moreover, we say v is k-*clean *if (a) holds; and we say v is* cleaning *if (b) holds.*
- *For any leaf v, $\mathcal{S}_v$ is k-clean (in such a case, we say that v is k-*clean*).*
- *For any k-clean internal node v, $\mathcal{T}_v$ starts with $\ell(v)$ non-adaptive queries[6] where $\ell(v) \ge 1$. In addition, for any $i \in \{1, \ldots, \ell(v) - 1\}$, any node of depth i in $\mathcal{T}_v$ is cleaning; and all node of depth $\ell(v)$ are k-clean.[7]*

▶ **Example 22.** If $\mathcal{T}$ is a decision tree (i.e., $|Q_v| \equiv 1$ for any internal node v) then it is k-clean for any k, where each internal node is k-clean.

If $\mathcal{T}$ is the depth-1 parity decision tree for $\mathcal{T}(x) = x_1x_2x_3$ (i.e., $\mathcal{T}$ only has a root v_0 querying $Q_{v_0} = \{1, 2, 3\}$), then it is 2-clean but not 3-clean, since for either leaf v we have $\{1, 2, 3\} \in \mathcal{S}_v$ but $\{1\} \notin \mathcal{S}_v$.

The benefit of having a k-clean parity decision tree is that it makes the expression of Fourier coefficients simpler.

▶ **Lemma 23.** *Let $\mathcal{T} \colon \{\pm 1\}^n \to \{0, 1\}$ be a k-clean parity decision tree and let S be a set of size $\ell \le k$. Let $v_0, \ldots, v_d$ be a random root-to-leaf path. Define $\boldsymbol{v}^{(0)}, \ldots, \boldsymbol{v}^{(d)} \in \{-1, 0, +1\}^n$ by setting $\boldsymbol{v}^{(i)}_j = \widehat{\mathcal{P}_{v_i}}(j)$ for each i, j. Recall that $\boldsymbol{v}^{(d)}_S = \prod_{j\in S} \boldsymbol{v}^{(d)}_j$. Then we have*

$$\widehat{\mathcal{T}}(S) = \mathop{\mathbb{E}}_{v_0,\ldots,v_d}\left[\mathcal{T}(v_d) \cdot \boldsymbol{v}^{(d)}_S\right].$$

Proof. Observe that for any $j \in J(v_i) \subseteq J$, the j-th coordinate is fixed after querying Q_{v_i}. Therefore we have

$$\widehat{\mathcal{T}}(S) = \mathop{\mathbb{E}}_{y\sim\mathcal{U}_n}[\mathcal{T}(y) \cdot y_S] = \mathop{\mathbb{E}}_{v_0,\ldots,v_d}\left[\mathcal{T}(v_d) \cdot \mathop{\mathbb{E}}_{y\sim\mathcal{P}_{v_d}}[y_S]\right] = \mathop{\mathbb{E}}_{v_0,\ldots,v_d}\left[\mathcal{T}(v_d) \cdot \widehat{\mathcal{P}}_{v_d}(S)\right]$$

By Fact 19, $\widehat{\mathcal{P}}_{v_d}(S) \neq 0$ iff $S \in \mathcal{S}_{v_d}$, which, due to $\ell \le k$ and v_d being a k-clean leaf, is equivalent to all coordinates in S being fixed along this path. Hence $\widehat{\mathcal{P}}_{v_d}(S) = \prod_{j\in S} \boldsymbol{v}^{(d)}_j$. ◀

4.2 Cleanup Process

We first analyze the cleanup process for a subspace.[8]

[6] This means for any $i \in \{0, 1 \ldots, \ell(v) - 1\}$, all nodes of depth i in $\mathcal{T}_v$ make the same query.

[7] This "leveled adaptive" condition is required just for convenience of proofs. In fact, one can show that the first few queries in $\mathcal{T}_v$ can be rearranged to make sure they are non-adaptive until we reach a k-clean node. See Lemma 24.

[8] The $k = 2$ case of Lemma 24 is essentially [4, Proposition 3.5]. However there is a gap in their proof. For example, if the parity decision tree non-adaptively queries $x_1x_2x_3x_4, x_1x_5, x_2x_6$ in order, then their analysis fails.

▶ **Lemma 24** (Clean subspace). *Let $k \geq 2$ be an integer and $\mathcal{S}$ be a subspace of rank at most d. We construct a new subspace $\mathcal{S}'$ (initialized as $\mathcal{S}$) as follows: while $\mathcal{S}'$ is not k-clean, we continue to update $\mathcal{S}' \leftarrow \mathsf{Span}\langle \mathcal{S}' \cup \{\{i\}\}\rangle$ with some mess-witness i. Then* $\mathsf{rank}(\mathcal{S}') \leq d \cdot k$ *and any update choice of mess-witnesses will result in the same final subspace $\mathcal{S}'$.*

Proof. Assume $\mathcal{S}$ is a subspace of $\mathbb{F}_2^n$. Then first note that the number of updates is finite, since we can update for at most n times.

Next we show that the number of updates and the final $\mathcal{S}'$ does not depend on the choice of mess-witnesses. We do so by an exchange argument. Let $i_1, \ldots, i_r$ and $i'_1, \ldots, i'_{r'}$ be two rounds of execution using different mess-witnesses. Then there exists some $t < \min\{r, r'\}$ such that $i_j = i'_j$ for all $j \leq t$, but $i_{t+1} \neq i'_{t+1}$. Let $\mathcal{S}_t = \mathsf{Span}\langle \mathcal{S} \cup \{\{i_1\}, \ldots, \{i_t\}\}\rangle$. Then there exist $S \ni i_{t+1}$ and $S' \ni i'_{t+1}$ (possibly $S = S'$) such that $S, S' \in \mathcal{S}_t$ but $\{i_{t+1}\}, \{i'_{t+1}\} \notin \mathcal{S}_t$. Since the final subspace is k-clean, we know there exists some $T \geq t$ such that

$$\{i_{t+1}\} \notin \mathsf{Span}\langle \mathcal{S} \cup \{\{i'_1\}, \ldots, \{i'_T\}\}\rangle \quad \text{but} \quad \{i_{t+1}\} \in \mathsf{Span}\left\langle \mathcal{S} \cup \left\{\{i'_1\}, \ldots, \{i'_{T+1}\}\right\}\right\rangle,$$

which means $\left\{i'_{T+1}, i_{t+1}\right\} \in \mathsf{Span}\langle \mathcal{S} \cup \{\{i'_1\}, \ldots, \{i'_T\}\}\rangle$. Hence we can safely replace i'_{T+1} with i_{t+1}, and then swap i_{t+1} with i'_{t+1}. We can perform this process as long as $(i_1, \ldots, i_r) \neq (i'_1, \ldots, i'_{r'})$, which means $r = r'$ and the final $\mathcal{S}'$ is always the same.

For any subspace $\mathcal{H}$, we define $\mathsf{rank}_1(\mathcal{H}) = |\{i \mid \{i\} \in \mathcal{H}\}|$ and thus $\mathsf{rank}(\mathcal{H}) - \mathsf{rank}_1(\mathcal{H}) \geq 0$. Now we analyze the following particular way to construct $\mathcal{S}'$: We initialize $\mathcal{S}'$ as $\mathcal{S}$. While $\mathcal{S}'$ is not k-clean, we find a minimal $S = \{i_1, \ldots, i_s\} \in \mathcal{S}'$ such that i_1 is a mess-witness; then we update $\mathcal{S}' \leftarrow \mathsf{Span}\langle \mathcal{S}' \cup \{\{i_1\}, \ldots, \{i_{s-1}\}\}\rangle$. Note that before the update, $1 < s \leq k$ and $\{i_j\} \notin \mathcal{S}'$ holds for each $j \in [s]$, since S is minimal and $\mathcal{S}'$ is not k-clean. Thus after the update, $\mathsf{rank}(\mathcal{S}')$ grows by $s - 1 \leq k - 1$ and $\mathsf{rank}_1(\mathcal{S}')$ grows by s, which means $\mathsf{rank}(\mathcal{S}') - \mathsf{rank}_1(\mathcal{S}')$ shrinks by 1. Hence we have at most $\mathsf{rank}(\mathcal{S}) - \mathsf{rank}_1(\mathcal{S}) \leq d$ updates before $\mathcal{S}'$ is k-clean; and the final $\mathcal{S}'$ has rank at most $\mathsf{rank}(\mathcal{S}) + (k-1) \cdot d \leq d \cdot k$. ◀

We now show how to convert an arbitrary parity decision tree into a k-clean parity decision tree which still has a small depth and fixes a small number of variables along each path. The latter quantity is in fact bounded by the depth as shown in Fact 25.

▶ **Fact 25.** *Let $\mathcal{T}$ be a depth-d parity decision tree. Let $v_0, \ldots, v_{d'}$ be any root-to-leaf path. Then we have $\sum_{i=0}^{d'-1} |J(v_i)| \leq d'$.*

Proof. Observe that $\sum_{i=0}^{d'-1} |J(v_i)| = \left|\left\{i \,\middle|\, \{i\} \in \mathsf{Span}\left\langle Q_{v_0}, \ldots, Q_{v_{d'-1}}\right\rangle\right\}\right| \leq d'$. ◀

▶ **Corollary 26.** *Let $\mathcal{T}$ be a depth-D k-clean parity decision tree. Let $v_0, \ldots, v_{D'}$ be any root-to-leaf path where at most d of the nodes $v_0, \ldots, v_{D'-1}$ are k-clean. Then $\sum_{i:|J(v_{i-1})|>1} |J(v_i)| \leq 2d$.*

Proof. By Fact 25 we have $\sum_{i=0}^{D'-1} |J(v_i)| - 1 \leq 0$. Since any v_i with $J(v_i) = \emptyset$ is not cleaning and therefore must be k-clean. Thus

$$\sum_{i:|J(v_i)|>1} |J(v_i)| - 1 \leq |\{i : J(v_i) = \emptyset\}| \leq d.$$

For $|J(v_i)| > 1$, we have $|J(v_i)| - 1 \geq |J(v_i)|/2$ and thus $\sum_{i:|J(v_i)|>1} |J(v_i)| \leq 2d$. ◀

▶ **Lemma 27** (Clean parity decision tree). *Let $k \geq 2$ be an integer. Let $\mathcal{T}$ be an arbitrary depth-d parity decision tree. Then there exists a k-clean parity decision tree $\mathcal{T}'$ of depth at most $d \cdot k$ equivalent to $\mathcal{T}$. Moreover, any root-to-leaf path in $\mathcal{T}'$ has at most d nodes that are k-clean.*

Algorithm 1 Clean parity decision tree: build $\mathcal{T}'$ from $\mathcal{T}$.

```
    Input: an arbitrary depth-$d$ parity decision tree $\mathcal{T}$
    Output: a parity decision tree $\mathcal{T}'$ with desired properties
 1  $r \leftarrow$ root of $\mathcal{T}$
 2  Initialize the root of $\mathcal{T}'$ as $r'$
 3  Build($r, r', 1$)
 4  Procedure Build($v, v', \ell$)
        /* $(v, v')$ are the current nodes on $(\mathcal{T}, \mathcal{T}')$; $\ell$ is the recursion depth.
           */
 5      if $v$ is a leaf then Label $v'$ with the label of $v$
 6      else
 7          $(v_-, v_+) \leftarrow$ the left and right child of $v$
 8          if $\widehat{\mathcal{P}_{v'}}(Q_v) = -1$ then Build($v_-, v', \ell + 1$)
 9          else if $\widehat{\mathcal{P}_{v'}}(Q_v) = +1$ then Build($v_+, v', \ell + 1$)
10          else                                /* $\widehat{\mathcal{P}_{v'}}(Q_v) = 0$ due to Fact 19 */
11              $Q_{v'} \leftarrow Q_v$
12              $(v'_-, v'_+) \leftarrow$ the left and right child of $v'$
13              Initialize $O \leftarrow \emptyset$
14              while Span $\langle \mathcal{S}_{v'} \cup \{Q_{v'}\} \cup O \rangle$ is not $k$-clean do
15                  Update $O \leftarrow O \cup \{\{i\}\}$, where $i$ is a mess-witness
16              end
17              $\mathcal{T}'$ non-adaptively queries every set (which is a singleton) in $O$ under $v'$ in
                 arbitrary order
18              foreach leaf $\widehat{v}$ under $v'_-$ do Build($v_-, \widehat{v}, \ell + 1$)
19              foreach leaf $\widehat{v}$ under $v'_+$ do Build($v_+, \widehat{v}, \ell + 1$)
20
21          end
22      end
```

Proof. We build $\mathcal{T}'$ by the following recursive algorithm. An example of the algorithm is provided in Figure 1

We now prove the correctness of Algorithm 1, which is guaranteed by the following claims.

- **For any internal node $v' \in \mathcal{T}'$, $Q_{v'}$ is not implied by its ancestors' queries.** By Fact 19, this is equivalent to $Q_{v'} \notin \mathcal{S}_{v'}$, which follows from the conditions in Line 8/9/13.
- **The depth of $\mathcal{T}'$ is at most $d \cdot k$.** Let $v_0, \ldots, v_{d'}$ be any root-to-leaf path of $\mathcal{T}$ and let $\mathcal{P}'$ be its corresponding path in $\mathcal{T}'$. Then the construction process of $\mathcal{P}'$ corresponds to the cleanup process for Span $\left\langle Q_{v_0}, \ldots, Q_{v_{d'-1}} \right\rangle$ in Lemma 24; hence the depth of $\mathcal{T}'$ equals $\mathsf{rank}(\mathcal{S}') \le d' \cdot k \le d \cdot k$ where $\mathcal{S}'$ is the k-clean subspace produced by applying Lemma 24.
- **$\mathcal{T} \equiv \mathcal{T}'$ and any root-to-leaf path in $\mathcal{T}'$ has at most d k-clean nodes.** This is because $\mathcal{T}'$ only refines $\mathcal{T}$ by inserting cleaning nodes.
- **Whenever we call Build($\cdot, v', \cdot$), v' is k-clean.** We prove by induction on ℓ. The base case Line 3 is obvious. For Line 8/9, we recurse on the same v', which is k-clean by induction. For Line 17/18, note that $\mathcal{S}_{\widehat{v}} =$ Span $\langle \mathcal{S}_{v'} \cup \{Q_{v'}\} \cup O \rangle$; hence from the condition in Line 13, it is k-clean.
- **Nodes created in Line 16 are cleaning.** Let $o = |O|$ and let $i_1, i_2, \ldots, i_o$ be the query

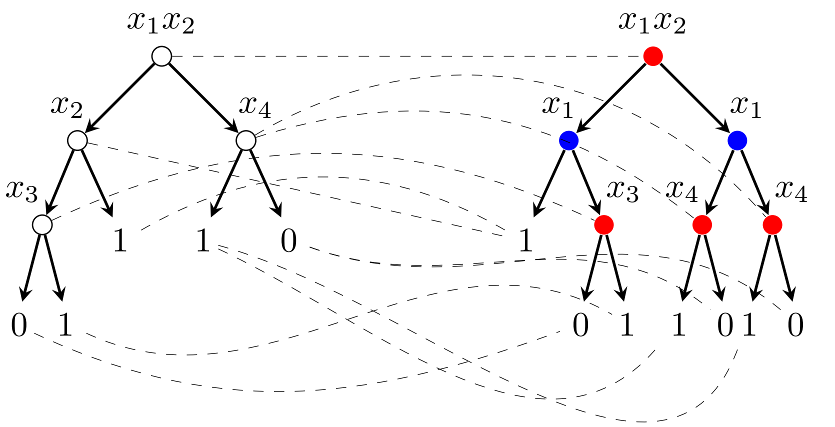


Figure 1 An example of the cleanup process with $k = 2$ where the LHS is $\mathcal{T}$ and the RHS is $\mathcal{T}'$. All the left (resp., right) outgoing edges are labeled with -1 (resp., $+1$). Red nodes and leaves are k-clean, and blue nodes are cleaning (i.e., non-adaptive queries). Nodes connected with dashed curves are invoked by Build.

order. For any $j \in [o]$, let v'_j be any one of the nodes created for i_j, then

$$\mathcal{S}_{v'_j} = \mathsf{Span}\left\langle \mathcal{S}_{v'} \cup \{Q_{v'}\} \cup \{\{i_1\}, \ldots, \{i_{j-1}\}\}\right\rangle,$$

which is not k-clean by Line 13; hence v'_j is cleaning by the condition in Line 13. ◀

5 Fourier Bounds for Parity Decision Trees

Our goal in this section is to prove Theorem 1 with detailed bounds provided.

5.1 Level-1 Bound

We first prove the concentration result for level-1. We start with the following simple bound for general parity decision trees.

▶ **Lemma 28.** *Let $\mathcal{T}\colon \{\pm 1\}^n \to \{0,1\}$ be a depth-D parity decision tree. Let $v_0, \ldots, v_{D'}$ be any root-to-leaf path. Define $\boldsymbol{v}^{(0)}, \ldots, \boldsymbol{v}^{(D')} \in \{-1,0,+1\}^n$ by setting $\boldsymbol{v}_j^{(i)} = \widehat{\mathcal{P}_{v_i}}(j)$ for each $0 \le i \le D'$ and $j \in [n]$. Then for any $a_1, \ldots, a_n \in \{-1,0,1\}$, we have $\left|\sum_{j=1}^n a_j \cdot \boldsymbol{v}_j^{(D')}\right| \le D' \le D$.*

Proof. Note that the set of non-zero coordinates in $\boldsymbol{v}^{(D')}$ is exactly $\bigcup_{i=0}^{D'-1} J(v_i)$. Hence by Fact 25, we have

$$\left|\sum_{j=1}^n a_j \cdot \boldsymbol{v}_j^{(D')}\right| \le \sum_{j=1}^n \left|\boldsymbol{v}_j^{(D')}\right| = \sum_{i=0}^{D'-1} |J(v_i)| \le D' \le D.$$ ◀

Now we give an improved bound for k-clean parity decision trees. To do so, we need one more notation which will be crucial in our analysis.

▶ **Notation 29.** Let $\mathcal{T}$ be a k-clean parity decision tree. For any node v, we define $C(v)$ as the nearest ancestor of v (including itself) that is k-clean.

▶ **Lemma 30.** *There exists a universal constant $\kappa \geq 1$ such that the following holds. Let $\mathcal{T}\colon \{\pm 1\}^n \to \{0,1\}$ be a depth-D $2k$-clean parity decision tree where $k \geq 1$ and any root-to-leaf path has at most d nodes that are $2k$-clean.*

Let $v_0, \dots, v_{D'}$ be a random root-to-leaf path. Define $\boldsymbol{v}^{(0)}, \dots, \boldsymbol{v}^{(D')} \in \{-1,0,+1\}^n$ by setting $\boldsymbol{v}_j^{(i)} = \widehat{\mathcal{P}_{v_i}}(j)$ for each $0 \leq i \leq D'$ and $j \in [n]$. Then for any $a_1, \dots, a_n \in \{-1,0,1\}$ and any $\varepsilon \leq 1/2$, we have $\mathbf{Pr}\left[\left|\sum_{j=1}^n a_j \cdot \boldsymbol{v}_j^{(D')}\right| \geq R(D,d,k,\varepsilon)\right] \leq \varepsilon$, where

$$R(D,d,k,\varepsilon) = \kappa \cdot \sqrt{\left(D + dk\left(\frac{1}{\varepsilon}\right)^{\frac{1}{k}}\right)\log\left(\frac{1}{\varepsilon}\right)}.$$

In the proof of Lemma 30 we will use the following simple claim.

▶ **Fact 31.** *Let $p_1, \dots, p_n$ be a sub-probability distribution, i.e., $p_i \geq 0$ and $\sum_{i=1}^n p_i \leq 1$. Let $a_1, \dots, a_n \in \mathbb{R}$. Then for any $k \in \mathbb{N}$, we have $\sum_{i=1}^n p_i a_i^{2k} \geq \left(\sum_{i=1}^n p_i a_i^2\right)^k$.*

Proof. We add $p_{n+1} = 1 - (\sum_{i=1}^n p_i)$ and $a_{n+1} = 0$ so p is a probability distribution. Then the claim follows from $\mathbb{E}[X^k] \geq \mathbb{E}[X]^k$, where random variable X gets value a_i^2 with probability p_i. ◀

Proof of Lemma 30. Extend $\boldsymbol{v}^{(D'+1)} = \cdots = \boldsymbol{v}^{(D)}$ to equal $\boldsymbol{v}^{(D')}$. For each $0 \leq i \leq D$, let $X^{(i)} = \sum_{j=1}^n a_j \cdot \boldsymbol{v}_j^{(i)}$. We define $\delta^{(i)} = 0$ for $D' < i \leq D$. For $1 \leq i \leq D'$, we let

$$\delta^{(i)} = X^{(i)} - X^{(i-1)} = \sum_{j=1}^n a_j \cdot \left(\boldsymbol{v}_j^{(i)} - \boldsymbol{v}_j^{(i-1)}\right) = \sum_{j \in J(v_{i-1})} a_j \cdot \boldsymbol{v}_j^{(i)},$$

where $J(v_{i-1})$ depends only on $C(v_{i-1})$ since $\mathcal{T}_{C(v_{i-1})}$ performs non-adaptive queries before (and possibly even after) reaching v_i. Note that for the two possible outcomes of querying Q_{v_i}, $\boldsymbol{v}_j^{(i)}$ is fixed to ± 1 respectively for each $j \in J(v_{i-1})$. Thus $\delta^{(i)} = \Delta^{(i)} \cdot z^{(i)}$ where $\Delta^{(i)}$ is a fixed value given $z^{(1)}, \dots, z^{(i-1)}$ and $z^{(1)}, \dots, z^{(D')}$ are independent unbiased coins in $\{\pm 1\}$.

Since $C(v_{i-1})$ is $2k$-clean, the collection of random variables $\left\{\boldsymbol{v}_j^{(i)} \,\middle|\, j \in J(v_{i-1})\right\}$ is $2k$-wise independent conditioning on $C(v_{i-1})$. Note that δ_i is a linear function and

$$\mathbb{E}\left[\delta^{(i)} \,\middle|\, C(v_{i-1})\right] = 0 \quad \text{and} \quad \mathbb{E}\left[\left(\delta^{(i)}\right)^2 \,\middle|\, C(v_{i-1})\right] = \sum_{j \in J(v_{i-1})} a_j^2 \leq |J(v_{i-1})|\,.$$

By the first bound in Lemma 15, we have

$$\mathbb{E}\left[\left(\delta^{(i)}\right)^{2k} \,\middle|\, C(v_{i-1})\right] \leq (2k-1)^k \cdot |J(v_{i-1})|^k\,, \tag{5}$$

and $\left|\delta^{(i)}\right| \leq |J(v_{i-1})|$ always. Our first goal is to bound $\mathbf{Pr}\left[\sum_{i=1}^D \left(\delta^{(i)}\right)^2 > D + 2\alpha^2 d\right]$. Observe that whenever the event $\sum_{i=1}^D \left(\delta^{(i)}\right)^2 > D + 2\alpha^2 d$ happens, it must be the case that $\sum_{i:|J(v_{i-1})|>1} \left(\delta^{(i)}\right)^2 > 2\alpha^2 d$. Thus,

$$\begin{aligned}
\mathbf{Pr}\left[\sum_{i=1}^{D}\left(\delta^{(i)}\right)^2 > D + 2\alpha^2 d\right] &\le \mathbf{Pr}\left[\sum_{i:|J(v_{i-1})|>1}\left(\delta^{(i)}\right)^2 > 2\alpha^2 d\right] \\
&= \mathbf{Pr}\left[\sum_{i:|J(v_{i-1})|>1}\frac{|J(v_{i-1})|}{2d}\cdot\frac{\left(\delta^{(i)}\right)^2}{|J(v_{i-1})|} > \alpha^2\right] \\
&\le \mathbf{Pr}\left[\sum_{i:|J(v_{i-1})|>1}\frac{|J(v_{i-1})|}{2d}\cdot\frac{\left(\delta^{(i)}\right)^{2k}}{|J(v_{i-1})|^k} > \alpha^{2k}\right] \\
&\qquad\qquad\text{(by Fact 31 and Corollary 26)} \\
&= \mathbf{Pr}\left[\sum_{i:|J(v_{i-1})|>1}\frac{\left(\delta^{(i)}\right)^{2k}}{|J(v_{i-1})|^{k-1}} > 2d\cdot\alpha^{2k}\right] \\
&\le \mathbb{E}\left[\sum_{i:|J(v_{i-1})|>1}\frac{\left(\delta^{(i)}\right)^{2k}}{|J(v_{i-1})|^{k-1}}\right]\cdot\frac{1}{2d\cdot\alpha^{2k}}. \\
&\qquad\qquad\text{(by Markov's inequality)}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\mathbb{E}\left[\sum_{i:|J(v_{i-1})|>1}\frac{\left(\delta^{(i)}\right)^{2k}}{|J(v_{i-1})|^{k-1}}\right] &= \sum_{i=1}^{D}\mathop{\mathbb{E}}_{C(v_{i-1})}\left[\frac{1_{|J(v_{i-1})|>1}}{|J(v_{i-1})|^{k-1}}\cdot\mathbb{E}\left[\left(\delta^{(i)}\right)^{2k}\,\middle|\, C(v_{i-1})\right]\right] \\
&\le \sum_{i=1}^{D}\mathop{\mathbb{E}}_{C(v_{i-1})}\left[1_{|J(v_{i-1})|>1}\cdot(2k-1)^k\cdot|J(v_{i-1})|\right] \quad \text{(by (5))} \\
&= (2k-1)^k\cdot\mathbb{E}\left[\sum_{i:|J(v_{i-1}|>1}|J(v_{i-1})|\right] \\
&\le (2k-1)^k\cdot 2d. \qquad\qquad \text{(by Corollary 26)}
\end{aligned}$$

Overall, we have

$$\mathbf{Pr}\left[\sum_{i=1}^{D}\left(\delta^{(i)}\right)^2 > D + 2\alpha^2 d\right] \le \frac{(2k-1)^k}{\alpha^{2k}}.$$

Then by Lemma 17 with $m = 1$, we have

$$\mathbf{Pr}\left[\left|X^{(D)}\right| = \left|\sum_{j=1}^{n} a_j\cdot\boldsymbol{v}_j^{(D)}\right| \ge \beta\sqrt{2\cdot(D+2\alpha^2 d)}\right] \le 2\cdot e^{-\beta^2/2} + \frac{(2k-1)^k}{\alpha^{2k}}.$$

The desired bound follows from setting

$$\alpha = \left(\frac{2}{\varepsilon}\right)^{\frac{1}{2k}}\sqrt{2k-1}, \quad \text{and} \quad \beta = \Theta\left(\sqrt{\log\left(\frac{1}{\varepsilon}\right)}\right). \qquad ◀$$

Now we prove the complete level-1 bound for parity decision trees.

▶ **Theorem 32.** *Let $\mathcal{T}\colon \{\pm 1\}^n \to \{0,1\}$ be a depth-d parity decision tree. Let $p = \mathbf{Pr}\left[\mathcal{T}(x)=1\right] \in \left[2^{-d}, 1/2\right]$.*[9] *Then we have*

$$\sum_{j=1}^{n} \left|\widehat{\mathcal{T}}(j)\right| \le p \cdot \min\left\{d, O\left(\sqrt{d}\cdot \log\left(\frac{1}{p}\right)\right)\right\} = O\left(\sqrt{d}\right).$$

Proof. For any $i \in [n]$, let $a_i = \mathsf{sgn}\left(\widehat{\mathcal{T}}(i)\right)$. Now we prove the two bounds separately.

First Bound. Let $v_0, \ldots, v_{d'}$ be a random root-to-leaf path in $\mathcal{T}$. Define $\boldsymbol{v}^{(0)}, \ldots, \boldsymbol{v}^{(d')} \in \{-1, 0, +1\}^n$ by setting $\boldsymbol{v}_j^{(i)} = \widehat{\mathcal{P}_{v_i}}(j)$ for each $0 \le i \le d'$ and $j \in [n]$. Since $\mathcal{T}$ is 1-clean in itself, by Lemma 23 we have

$$\sum_{j=1}^{n} \left|\widehat{\mathcal{T}}(j)\right| = \sum_{j=1}^{n} a_i \cdot \widehat{\mathcal{T}}(j) = \mathop{\mathbb{E}}_{v_0,\ldots,v_{d'}}\left[\mathcal{T}(v_{d'}) \cdot \sum_{j=1}^{n} a_j \cdot \boldsymbol{v}_j^{(d')}\right] \le \mathop{\mathbb{E}}_{v_0,\ldots,v_{d'}}\left[\mathcal{T}(v_{d'}) \cdot |V|\right], \quad (6)$$

where $V = \sum_{j=1}^{n} a_j \cdot \boldsymbol{v}_j^{(d')}$. Hence by Lemma 28, we have $(6) \le d \cdot \mathbb{E}\left[\mathcal{T}(v_{d'})\right] = p \cdot d$.

Second Bound. By Lemma 27, we construct a $2k$-clean parity decision tree $\mathcal{T}'$ of depth $D \le 2d \cdot k$ equivalent to $\mathcal{T}$, where $k = \Theta(\log(1/p))$. Let $U = \sum_{j=1}^{n} a_j \cdot \boldsymbol{u}_j^{(D')}$. Then we have

$$\sum_{j=1}^{n} \left|\widehat{\mathcal{T}}(j)\right| = \sum_{j=1}^{n} \left|\widehat{\mathcal{T}'}(j)\right| = \mathop{\mathbb{E}}_{u_0,\ldots,u_{D'}}\left[\mathcal{T}'(u_{D'}) \cdot \sum_{j=1}^{n} a_j \cdot \boldsymbol{u}_j^{(D')}\right] \le \mathop{\mathbb{E}}_{u_0,\ldots,u_{D'}}\left[\mathcal{T}'(u_{D'}) \cdot |U|\right]. \quad (7)$$

Lemma 30 implies that for all $\varepsilon > 0$, $\mathbf{Pr}\left[|U| \ge R(\varepsilon)\right] \le \varepsilon$ where

$$R(\varepsilon) = R(D, d, k, \varepsilon) = O\left(\sqrt{dk \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{1}{k}} \cdot \log\left(\frac{1}{\varepsilon}\right)}\right).$$

For integer $i \ge 1$, let $I_i = \left[R\left(p/2^i\right), R\left(p/2^{i+1}\right)\right]$ and $I_0 = [0, R(p/2)]$ be intervals. Then for each $i \ge 1$, $\mathbf{Pr}\left[|U| \in I_i\right] \le p/2^i$. We also know that $\mathbb{E}_{u_0,\ldots,u_{D'}}\left[\mathcal{T}'(u_{D'})\right] \le p$. Thus,

$$\begin{aligned}
(7) &= \mathop{\mathbb{E}}_{u_0,\ldots,u_{D'}}\left[\mathcal{T}'(u_{D'}) \cdot |U| \cdot \sum_{i=0}^{+\infty} \mathbb{1}_{|U| \in I_i}\right] \\
&\le R\left(\frac{p}{2}\right) \cdot \mathop{\mathbb{E}}_{u_0,\ldots,u_{D'}}\left[\mathcal{T}'(u_{D'})\right] + \sum_{i=1}^{+\infty} R\left(\frac{p}{2^{i+1}}\right) \cdot \mathop{\mathbb{E}}_{u_0,\ldots,u_{D'}}\left[\mathbb{1}_{|U| \in I_i}\right] \\
&\le \sum_{i=0}^{+\infty} R\left(\frac{p}{2^{i+1}}\right) \cdot \frac{p}{2^i} \\
&= \sum_{i=0}^{+\infty} O\left(p \cdot \sqrt{dk \cdot \left(\frac{2^{i+1}}{p}\right)^{\frac{1}{k}} \cdot \left(\log\left(\frac{1}{p}\right) + i + 1\right)}\right) \cdot \frac{1}{2^i} \\
&= O\left(p \cdot \sqrt{dk \cdot \log\left(\frac{1}{p}\right)}\right) = O\left(p \cdot \sqrt{d} \cdot \log\left(\frac{1}{p}\right)\right).
\end{aligned}$$

◀

[9] If $p < 2^{-d}$, then $p = 0$ and $\mathcal{T} \equiv 0$. If $p > 1/2$, we can consider $\widetilde{\mathcal{T}} = 1 - \mathcal{T}$ by symmetry.

5.2 Level-ℓ Bound

Now we turn to the general levels.

▶ **Lemma 33.** *There exists a universal constant $\tau \geq 1$ such that the following holds. Let $\ell \geq 1$ be an integer. Let $\mathcal{T} \colon \{\pm 1\}^n \to \{0,1\}$ be a depth-D $2k$-clean parity decision tree where $k \geq 4 \cdot \ell$ and $n \geq \max\{\tau, k, D\}$ and any root-to-leaf path has at most d nodes that are $2k$-clean.*

Let $v_0, \ldots, v_{D'}$ be a random root-to-leaf path. Define $\boldsymbol{v}^{(0)}, \ldots, \boldsymbol{v}^{(D')} \in \{-1,0,+1\}^n$ by setting $\boldsymbol{v}_j^{(i)} = \widehat{\mathcal{P}_{v_i}}(j)$ for each $0 \leq i \leq D'$ and $j \in [n]$. Extend $\boldsymbol{v}^{(D'+1)} = \cdots = \boldsymbol{v}^{(D)}$ to equal $\boldsymbol{v}^{(D')}$. Then for any sequence $a_S \in \{-1,0,1\}, S \in \binom{[n]}{\ell}$, any $\varepsilon \leq 1/2$ and $t \in \{0,\ldots,\ell\}$, we have

$$\mathbf{Pr}\left[\exists t' \in \{0,\ldots,t\}, \exists T \in \binom{[n]}{\ell - t'}, \exists i \in [D], \left|\sum_{S \subseteq \overline{T}, |S| = t'} a_{S \cup T} \cdot \boldsymbol{v}_S^{(i)}\right| \geq M(D,d,k,\ell,t',\varepsilon)\right] \leq \varepsilon \cdot t,$$

where we recall that $\boldsymbol{v}_S^{(i)} = \prod_{j \in S} \boldsymbol{v}_j^{(i)}$ and where

$$M(D,d,k,\ell,t',\varepsilon) = \left(\tau \cdot (D + dk) \cdot \left(\frac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}} \log\left(\frac{n^\ell}{\varepsilon}\right)\right)^{t'/2}.$$

Proof. We prove the bound by induction on $t = 0,1,\ldots,\ell$ and show $\tau = 10^4$ suffices. The base case $t = 0$ is trivial, since for any fixed T and i, we always have $\left|a_T \cdot \boldsymbol{v}_\emptyset^{(i)}\right| \leq 1 = M(D,d,k,\ell,0,\varepsilon)$.

Now we focus on the case where $1 \leq t \leq \ell$. For each $0 \leq i \leq D$ and $T \in \binom{[n]}{\ell - t}$, let

$$X_T^{(i)} = \sum_{S \subseteq \overline{T}, |S| = t} a_{S \cup T} \cdot \boldsymbol{v}_S^{(i)}.$$

For $1 \leq i \leq D'$, we have

$$\begin{aligned}
X_T^{(i)} - X_T^{(i-1)} &= \sum_{S \subseteq \overline{T}, |S| = t, S \cap J(v_{i-1}) \neq \emptyset} a_{S \cup T} \cdot \boldsymbol{v}_S^{(i)} \\
&= \sum_{r=1}^{t} \sum_{\substack{U \subseteq J(v_{i-1}) \cap \overline{T}, \\ |U| = r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{V \subseteq \overline{T \cup J(v_{i-1})}, \\ |U| + |V| = t}} a_{T \cup U \cup V} \cdot \boldsymbol{v}_V^{(i)} \\
&= \sum_{r=1}^{t} \sum_{\substack{U \subseteq J(v_{i-1}) \cap \overline{T}, \\ |U| = r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{V \subseteq \overline{T \cup J(v_{i-1})}, \\ |U| + |V| = t}} a_{T \cup U \cup V} \cdot \boldsymbol{v}_V^{(i-1)} \\
&\qquad \text{(since } \boldsymbol{v}_j^{(i)} = \boldsymbol{v}_j^{(i-1)} \text{ for all } j \notin J(v_{i-1})\text{)} \\
&= \underbrace{\sum_{r=1}^{t} \sum_{\substack{U \subseteq J(v_{i-1}) \cap \overline{T}, \\ |U| = r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{V \subseteq \overline{T \cup U}, \\ |U| + |V| = t}} a_{T \cup U \cup V} \cdot \boldsymbol{v}_V^{(i-1)}}_{A(T,r,i)}. \\
&\qquad \text{(since } \boldsymbol{v}_j^{(i-1)} = 0 \text{ for all } j \in J(v_{i-1})\text{)}
\end{aligned}$$

Observe that conditioning on v_{i-1},

- if r is an even number, then $A(T,r,i)$ is a fixed value independent of $\boldsymbol{v}^{(i)}$;
- if r is an odd number, then $A(T,r,i)$ is an unbiased coin with magnitude independent of $\boldsymbol{v}^{(i)}$.

Therefore, trying to apply Lemma 17, we write $X_T^{(i)} - X_T^{(i-1)} = \mu_T^{(i)} + \Delta_T^{(i)} \cdot z_T^{(i)}$, where $z_T^{(1)}, \dots, z_T^{(D)}$ are independent unbiased coins in $\{\pm 1\}$ and $\mu_T^{(i)} = \Delta_T^{(i)} = 0$ for $D' < i \le D$ and

$$\mu_T^{(i)} = \sum_{\substack{r=2,\\ \text{even}}}^{t} A(T,r,i) \quad \text{and} \quad \Delta_T^{(i)} = \left| \sum_{\substack{r=1,\\ \text{odd}}}^{t} A(T,r,i) \right| \quad \text{for } 1 \le i \le D'. \tag{8}$$

First Bound on $A(T,r,i)$. Let $\mathcal{E}_1$ be the following event:

$$\mathcal{E}_1 = \text{“ } \exists \widehat{t} \in \{0,\dots,t-1\}, \exists T' \in \binom{[n]}{\ell - \widehat{t}}, \exists i' \in [D], \left| X_{T'}^{(i')} \right| \ge M\left(D,k,\ell,\widehat{t},\varepsilon\right) \text{”}.$$

By the induction hypothesis, we have

$$\mathbf{Pr}\left[\mathcal{E}_1\right] \le (t-1) \cdot \varepsilon. \tag{9}$$

We first derive a simple bound, that will be effective for small values of $|J(v_{i-1})|$.

▷ Claim 34. When $\mathcal{E}_1$ does not happen, $|A(T,r,i)| \le |J(v_{i-1})|^r \cdot M(D,d,k,\ell,t-r,\varepsilon)$ holds for all $r \in [t], i \in [D], T \in \binom{[n]}{\ell - t}$.

Proof. Since $\mathcal{E}_1$ does not happen, by union bound we have

$$\begin{aligned} |A(T,r,i)| &= \left| \sum_{\substack{U \subseteq J(v_{i-1}) \cap \overline{T},\\ |U|=r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{V \subseteq \overline{T \cup U},\\ |U|+|V|=t}} a_{T\cup U\cup V} \cdot \boldsymbol{v}_V^{(i-1)} \right| \le |J(v_{i-1})|^r \max_{U \subseteq \overline{T}, |U|=r} \left| X_{T\cup U}^{(i-1)} \right| \\ &\le |J(v_{i-1})|^r \cdot M(D,d,k,\ell,t-r,\varepsilon). \end{aligned}$$

◁

Second Bound on $A(T,r,i)$. The second bound requires a more refined decomposition on $A(T,r,i)$.

Assume that $c(i-1)$ is the index of $C(v_{i-1})$ in $v_0, \dots, v_{D'}$, i.e., $v_{c(i-1)} = C(v_{i-1})$. This means that $v_{c(i-1)}$ is the closest ancestor to v_{i-1} that is $2k$-clean. Then define

$$L(v_{i-1}) = \bigcup_{c(i-1) \le i' < i-1} J(v_{i'}).$$

The elements of $L(v_{i-1})$ are precisely the coordinates fixed by the queries from $Q_{v_{c(i-1)}}$ to $Q_{v_{i-1}}$, excluding the latter. Since $\mathcal{T}_{C(v_{i-1})}$ makes non-adaptive queries before (and possibly even after) reaching v_i, $L(v_{i-1})$ and $J(v_{i-1})$ depend only on $C(v_{i-1})$ and i. We now expand $A(T,r,i)$ by also grouping terms based on the number of coordinates in $L(v_{i-1})$ as follows:

$$\begin{aligned}
A(T,r,i) &= \sum_{\substack{U\subseteq J(v_{i-1})\cap\overline{T},\\ |U|=r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{V\subseteq\overline{T\cup U},\\ |U|+|V|=t}} a_{T\cup U\cup V}\cdot \boldsymbol{v}_V^{(i-1)}\\
&= \sum_{r'=0}^{t-r} \sum_{\substack{U\subseteq J(v_{i-1})\cap\overline{T},\\ |U|=r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{W\subseteq L(v_{i-1})\cap\overline{T},\\ |W|=r'}} \boldsymbol{v}_W^{(i-1)} \sum_{\substack{W'\subseteq\overline{T\cup U\cup L(v_{i-1})}\\ |W'|=t-r-r'}} a_{T\cup U\cup W\cup W'}\cdot \boldsymbol{v}_{W'}^{(i-1)}\\
&= \sum_{r'=0}^{t-r} \sum_{\substack{U\subseteq J(v_{i-1})\cap\overline{T},\\ |U|=r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{W\subseteq L(v_{i-1})\cap\overline{T},\\ |W|=r'}} \boldsymbol{v}_W^{(i-1)} \sum_{\substack{W'\subseteq\overline{T\cup U\cup L(v_{i-1})}\\ |W'|=t-r-r'}} a_{T\cup U\cup W\cup W'}\cdot \boldsymbol{v}_{W'}^{c(i-1)}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\text{(since } \boldsymbol{v}_j^{(i-1)}=\boldsymbol{v}_j^{c(i-1)} \text{ for all } j\notin L(v_{i-1})\text{)}\\
&= \sum_{r'=0}^{t-r} \sum_{\substack{U\subseteq J(v_{i-1})\cap\overline{T},\\ |U|=r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{W\subseteq L(v_{i-1})\cap\overline{T},\\ |W|=r'}} \boldsymbol{v}_W^{(i-1)} \sum_{\substack{W'\subseteq\overline{T\cup U\cup W}\\ |W'|=t-r-r'}} a_{T\cup U\cup W\cup W'}\cdot \boldsymbol{v}_{W'}^{c(i-1)}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\text{(since } \boldsymbol{v}_j^{c(i-1)}=0 \text{ for all } j\in L(v_{i-1})\text{)}\\
&= \sum_{r'=0}^{t-r} \underbrace{\sum_{\substack{U\subseteq J(v_{i-1})\cap\overline{T},\\ |U|=r}} \boldsymbol{v}_U^{(i)} \sum_{\substack{W\subseteq L(v_{i-1})\cap\overline{T},\\ |W|=r'}} \boldsymbol{v}_W^{(i-1)}\cdot X_{T\cup U\cup W}^{c(i-1)}\cdot}_{\Gamma_T^{(i)}(r,r')}
\end{aligned}$$

Since $C(v_{i-1})$ is $2k$-clean, by Fact 19, the collection of random variables

$$\left\{\boldsymbol{v}_j^{(i)}\,\middle|\, j\in J(v_{i-1})\right\}\cup\left\{\boldsymbol{v}_j^{(i-1)}\,\middle|\, j\in L(v_{i-1})\right\}$$

is $2k$-wise independent conditioning on $C(v_{i-1})$. Note that $\Gamma_T^{(i)}(r,r')$ is a polynomial of degree at most $r+r'\le\ell<k$, that $\mathbb{E}\left[\Gamma_T^{(i)}(r,r')\,\middle|\,C(v_{i-1})\right]=0$, and

$$\begin{aligned}
\sigma_T^2(r,r',C(v_{i-1}),i) &:= \mathbb{E}\left[\left(\Gamma_T^{(i)}(r,r')\right)^2\,\middle|\,C(v_{i-1})\right]\\
&= \sum_{\substack{U\subseteq J(v_{i-1})\cap\overline{T},\\ |U|=r}}\ \sum_{\substack{W\subseteq L(v_{i-1})\cap\overline{T},\\ |W|=r'}} \left(X_{T\cup U\cup W}^{c(i-1)}\right)^2\\
&\le \left(|J(v_{i-1})|\right)^r\left(|L(v_{i-1})|\right)^{r'}\left(\max_{|T'|=r+r'+\ell-t,\,i'\in[D]}\left|X_{T'}^{(i')}\right|\right)^2\\
&\le \left(|J(v_{i-1})|\right)^r D^{r'}\left(\max_{|T'|=r+r'+\ell-t,\,i'\in[D]}\left|X_{T'}^{(i')}\right|\right)^2.\\
&\qquad\qquad\qquad\qquad\text{(since } |L(v_{i-1})|\le D \text{ by Fact 25)}
\end{aligned}$$

We also have the following claim, the proof of which follows from Lemma 15 applied to the low degree polynomial $\Gamma_T^{(i)}$. The proof is deferred to Appendix C.

▷ Claim 35. $\mathbf{Pr}\left[\mathcal{E}_2\right]\le\varepsilon/3$, where $\mathcal{E}_2$ is the following event: $\exists T\in\binom{[n]}{\ell-t}, i, r, r'$, such that

$$\left|\Gamma_T^{(i)}(r,r')\right|\ge\left(100\min\left\{k,\log\left(\frac{n^\ell}{\varepsilon}\right)\right\}\cdot\left(\frac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}}\right)^{\frac{r+r'}{2}}\cdot\sigma_T(r,r',C(v_{i-1}),i).$$

On the other hand, when $\mathcal{E}_1 \vee \mathcal{E}_2$ does not happen, the following calculation holds for all $T \in \binom{[n]}{\ell-t}$, $i \in [D']$, $r \in [t]$, $0 \le r' \le t-r$:

$$\begin{aligned}
&\left|\Gamma_T^{(i)}(r,r')\right| \\
&\le M\left(D,k,\ell,t-r-r',\varepsilon\right) \cdot \sqrt{\left(100 \min\left\{k, \log\left(\tfrac{n^\ell}{\varepsilon}\right)\right\} \cdot \left(\tfrac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}}\right)^{r+r'} \left(|J(v_{i-1})|\right)^r \cdot D^{r'}} \\
&\le M\left(D,k,\ell,t-r-r',\varepsilon\right) \cdot \sqrt{\left(100 \cdot \left(\tfrac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}}\right)^{r+r'} \left(|J(v_{i-1})| \cdot k\right)^r \cdot \left(D \cdot \log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{r'}} \\
&= \sqrt{\left(\tau(D+dk)\left(\tfrac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}} \log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{t-r-r'} \left(100\left(\tfrac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}}\right)^{r+r'} \left(|J(v_{i-1})| \cdot k\right)^r \left(D \cdot \log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{r'}} \\
&\le \sqrt{\left(\tau(D+dk)\left(\tfrac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}} \log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{t} \left(\tfrac{100}{\tau}\right)^{r+r'} \left(\tfrac{|J(v_{i-1})|}{d \cdot \log(n^\ell/\varepsilon)}\right)^r} \\
&\le \sqrt{\left(\tau(D+dk)\left(\tfrac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}} \log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{t} \left(\tfrac{200}{\tau}\right)^{r+r'} \left(\tfrac{|J(v_{i-1})|}{2d}\right)^r \tfrac{1}{\log(n^\ell/\varepsilon)}} \\
&= M(D,d,k,\ell,t,\varepsilon) \cdot \sqrt{\left(\tfrac{200}{\tau}\right)^{r+r'} \left(\tfrac{|J(v_{i-1})|}{2d}\right)^r \tfrac{1}{\log(n^\ell/\varepsilon)}}.
\end{aligned}$$

Hence we have a second bound on $A(T,r,i)$.

▷ Claim 36. When $\mathcal{E}_1 \vee \mathcal{E}_2$ does not happen, the following holds for all $r \in [t], i \in [D], T \in \binom{[n]}{\ell-t}$:

$$|A(T,r,i)| \le \frac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log\left(n^\ell/\varepsilon\right)}} \cdot \sqrt{\left(\frac{800}{\tau}\right)^r \left(\frac{|J(v_{i-1})|}{2d}\right)^r}.$$

Proof. Since $\mathcal{E}_1 \vee \mathcal{E}_2$ does not happen, by union bound and noticing $\tau \ge 800$ we have

$$\begin{aligned}
&|A(T,r,i)| \\
&\le \sum_{r'=0}^{t-r} \left|\Gamma_T^{(i)}(r,r')\right| \le \frac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log\left(n^\ell/\varepsilon\right)}} \cdot \sqrt{\left(\frac{200}{\tau}\right)^r \left(\frac{|J(v_{i-1})|}{2d}\right)^r} \cdot \sum_{r'=0}^{+\infty} \left(\frac{200}{\tau}\right)^{r'/2} \\
&\le \frac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log\left(n^\ell/\varepsilon\right)}} \cdot \sqrt{\left(\frac{800}{\tau}\right)^r \left(\frac{|J(v_{i-1})|}{2d}\right)^r}.
\end{aligned}$$

◁

Final Bound on $\mu_T^{(i)}$ and $\delta_T^{(i)}$. Combining Claim 34 and Claim 36, if $\mathcal{E}_1 \vee \mathcal{E}_2$ does not happen we have

$$|A(T,r,i)| \le M(D,d,k,\ell,t-r,\varepsilon) + \frac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log\left(n^\ell/\varepsilon\right)}} \cdot \sqrt{\left(\tfrac{800}{\tau}\right)^r \left(\tfrac{|J(v_{i-1})|}{2d}\right)^r} \cdot 1_{|J(v_{i-1})|>1} \quad (10)$$

To see this, if $|J(v_{i-1})| \le 1$, we use the bound from Claim 34 as the first term in (10). Otherwise $|J(v_{i-1})| > 1$, in which case we use the bound from Claim 36 as the second term in (10).

By Corollary 26, we can now bound $\sum_{i=1}^{D} \left|\mu_T^{(i)}\right|$ and $\sum_{i=1}^{D} \left|\Delta_T^{(i)}\right|^2$ as Claim 37. Its proof is deferred in Appendix D.

▷ Claim 37. When $\mathcal{E}_1 \vee \mathcal{E}_2$ does not happen, $\sum_{i=1}^{D}\left|\mu_T^{(i)}\right| \le R$ and $\sum_{i=1}^{D}\left|\Delta_T^{(i)}\right|^2 \le R^2$ hold for all $T \in \binom{[n]}{\ell-t}$, where

$$R = \frac{M(D,d,k,\ell,t,\varepsilon)}{5 \cdot \sqrt{\log\left(n^\ell/\varepsilon\right)}}. \tag{11}$$

Complete Induction. Let $\beta = \sqrt{2 \cdot \log\left(n^\ell/\varepsilon\right)} \ge 1$ and observe that

$$\begin{aligned} R + \beta \cdot \sqrt{2} \cdot R &\le \beta \cdot 2\sqrt{2} \cdot R && \text{(due to } \beta \ge 1\text{)} \\ &= \frac{2\sqrt{2} \cdot \sqrt{2 \cdot \log\left(n^\ell/\varepsilon\right)}}{5 \cdot \sqrt{\log\left(n^\ell/\varepsilon\right)}} \cdot M(D,d,k,\ell,t,\varepsilon) && \text{(due to (11))} \\ &\le M(D,d,k,\ell,t,\varepsilon). \end{aligned}$$

Then we have

$$\begin{aligned} &\mathbf{Pr}\left[\exists t' \in \{0,\dots,t\}, \exists T' \in \binom{[n]}{\ell-t'}, \exists i \in [D],\ \left|X_{T'}^{(i)}\right| \ge M\left(D,d,k,\ell,t',\varepsilon\right)\right] \\ &= \mathbf{Pr}\left[\mathcal{E}_1 \bigvee \left(\exists T \in \binom{[n]}{\ell-t}, \exists i \in [D],\ \left|X_T^{(i)}\right| \ge M\left(D,d,k,\ell,t,\varepsilon\right)\right)\right] \\ &\le \mathbf{Pr}\left[(\mathcal{E}_1 \vee \mathcal{E}_2) \bigvee \left(\exists T \in \binom{[n]}{\ell-t}, \exists i \in [D],\ \left|X_T^{(i)}\right| \ge R + \beta \cdot \sqrt{2} \cdot R\right)\right] \\ &\le (t-1) \cdot \varepsilon + \frac{\varepsilon}{3} + 2n^{\ell-t} \cdot e^{-\beta^2/2} \qquad \text{(due to (9), Claim 35, Lemma 17, and Claim 37)} \\ &\le (t-1) \cdot \varepsilon + \frac{\varepsilon}{3} + \frac{1}{3} \cdot n^\ell \cdot e^{-\beta^2/2} \\ &\le t \cdot \varepsilon. \end{aligned}$$

◀

Before we prove the complete level-ℓ bound for parity decision trees, we first prove a simple bound for the number of vectors with a given weight in a subspace.

▶ **Lemma 38.** *Let $\ell \ge 1$ be an integer and $\mathcal{S}$ be a subspace of rank at most d. Let $U = \{S \mid |S| = \ell, S \in \mathcal{S}\}$, then $|U| \le \min\left\{\binom{d\cdot\ell}{\ell}, 2^d - 1\right\}$.*

Proof. Let $\{S_1,\dots,S_{d'}\}$ be a maximal set of independent vectors in U. Then $d' \le d$ and $|S_i| = \ell$ holds for all $i \in [d']$. Since $U \subseteq \mathsf{Span}\langle S_1,\dots,S_{d'}\rangle$ and $\emptyset \notin U$, we have

$$|U| \le \left|\mathsf{Span}\langle S_1,\dots,S_{d'}\rangle\right| - 1 = 2^{d'} - 1 \le 2^d - 1.$$

On the other hand, observe that $U \subseteq \binom{S_1 \cup \dots \cup S_{d'}}{\ell}$, hence we also have

$$|U| \le \left|\binom{S_1 \cup \dots \cup S_{d'}}{\ell}\right| \le \binom{d' \cdot \ell}{\ell} \le \binom{d \cdot \ell}{\ell}.$$

◀

We remark that in Lemma 38, it is conjectured the bound should be $\binom{d+1}{\ell}$ when $d \ge 2 \cdot \ell$ [19, 6].

▶ **Theorem 39.** *Let $\ell \ge 1$ be an integer. Let $\mathcal{T} \colon \{\pm 1\}^n \to \{0,1\}$ be a depth-d parity decision tree where $n \ge \max\{d,\ell\}$. Let $p = \mathbf{Pr}[\mathcal{T}(x) = 1] \ge 2^{-d}$.*[10] *Then we have*

$$\sum_{S \subseteq [n] : |S| = \ell} \left|\widehat{\mathcal{T}}(S)\right| \le p \cdot \min\left\{\binom{d \cdot \ell}{\ell}, 2^d - 1,\ O\Big(\sqrt{d} \cdot \log\left(\tfrac{n^\ell}{p}\right)\Big)^\ell\right\} = O\Big(\sqrt{d} \cdot \ell \cdot \log(n)\Big)^\ell.$$

[10] If $p < 2^{-d}$, then $p = 0$ and $\mathcal{T} \equiv 0$.

Proof. For any $S \in \binom{[n]}{\ell}$, let $a_S = \mathsf{sgn}\left(\widehat{\mathcal{T}}(S)\right)$. Now we prove the bounds separately.

First Two Bounds. Let $v_0, \dots, v_{d'}$ be a random root-to-leaf path. Then by the definition of $\widehat{\mathcal{P}_v}$ and $\mathcal{S}_v$ and Fact 19, we have

$$\begin{aligned}\sum_S \left|\widehat{\mathcal{T}}(S)\right| &= \sum_S a_S \cdot \widehat{\mathcal{T}}(S) = \mathop{\mathbb{E}}_{v_0,\dots,v_{d'}}\left[\mathcal{T}(v_{d'}) \cdot \sum_S a_S \cdot \widehat{\mathcal{P}_{v_{d'}}}(S)\right] \\ &\le \mathop{\mathbb{E}}_{v_0,\dots,v_{d'}}\left[\mathcal{T}(v_{d'}) \cdot \sum_S \left|\widehat{\mathcal{P}_{v_{d'}}}(S)\right|\right] = \mathop{\mathbb{E}}_{v_0,\dots,v_{d'}}\left[\mathcal{T}(v_{d'}) \cdot |V|\right], \qquad (12)\end{aligned}$$

where $a_S = \mathsf{sgn}\left(\widehat{\mathcal{T}}(S)\right)$ and $V = \left\{S \in \binom{[n]}{\ell} \,\middle|\, S \in \mathcal{S}_{v_{d'}}\right\}$. Note that

$$\mathsf{rank}\left(\mathcal{S}_{v_{d'}}\right) = \mathsf{rank}\left(\mathsf{Span}\left\langle Q_{v_0}, \dots, Q_{v_{d'-1}}\right\rangle\right) \le d' \le d.$$

Hence by Lemma 38, we have $(12) \le \min\left\{\binom{d\cdot\ell}{\ell}, 2^d - 1\right\} \cdot \mathbb{E}\left[\mathcal{T}(v_{d'})\right] = p \cdot \min\left\{\binom{d\cdot\ell}{\ell}, 2^d - 1\right\}$.

Third Bound. By Lemma 27, we construct a $2k$-clean parity decision tree $\mathcal{T}'$ of depth $D \le 2d \cdot k$ equivalent to $\mathcal{T}$, where $k = \Theta\left(\log\left(n^\ell/p\right)\right) \ge 4 \cdot \ell$. We also add dummy variables to make sure $n' = \max\{\tau, k, 6D, n\}$, where $\mathcal{T}'$ has n' inputs and τ is the universal constant in Lemma 33.

Let $u_0, \dots, u_{D'}$ be a random root-to-leaf path in $\mathcal{T}'$. Define $\boldsymbol{u}^{(0)}, \dots, \boldsymbol{u}^{(D')} \in \{-1, 0, +1\}^n$ by setting $\boldsymbol{u}_j^{(i)} = \widehat{\mathcal{P}_{u_i}}(j)$ for each $0 \le i \le D'$ and $j \in [n]$. Then extend $\boldsymbol{u}^{(D'+1)} = \boldsymbol{u}^{(D'+2)} = \dots = \boldsymbol{u}^{(D)}$ to equal $\boldsymbol{u}^{(D')}$. By Lemma 23, we have

$$\sum_S \left|\widehat{\mathcal{T}}(S)\right| = \sum_S \left|\widehat{\mathcal{T}'}(S)\right| = \mathop{\mathbb{E}}_{u_0,\dots,u_{D'}}\left[\mathcal{T}(u_{D'}) \cdot \sum_S a_S \cdot \boldsymbol{u}_S^{(D)}\right] \le \mathop{\mathbb{E}}_{u_0,\dots,u_{D'}}\left[\mathcal{T}(u_{D'}) \cdot |U|\right], \quad (13)$$

where $U = \sum_S a_S \cdot \boldsymbol{u}_S^{(D)}$.

Now we apply Lemma 33 with $t = \ell, \varepsilon = \Theta\left(p/d^{\ell/2}\right) \le 1/2$ to obtain the following bound[11]

$$M = M(D, d, k, \ell, \ell, \varepsilon) = \left(O\left(\sqrt{d} \cdot \log\left(\tfrac{n^\ell}{p}\right)\right)\right)^\ell$$

such that $\mathbf{Pr}\left[|U| \ge M\right] \le \ell \cdot \varepsilon$. Then, combining the first bound, we have

$$\begin{aligned}(13) &= \mathbb{E}\left[\mathcal{T}(u_{D'}) \cdot |U| \cdot \left(1_{|U|<M} + 1_{|U|\ge M}\right)\right] \le M \cdot \mathbb{E}\left[\mathcal{T}(u_{D'})\right] + \ell \cdot \varepsilon \cdot \binom{d \cdot \ell}{\ell} \\ &= p \cdot \left(O\left(\sqrt{d} \cdot \log\left(\tfrac{n^\ell}{p}\right)\right)\right)^\ell,\end{aligned}$$

which is maximized at $p = 1$, hence $(13) = O\left(\sqrt{d} \cdot \ell \cdot \log(n)\right)^\ell$ as desired. ◀

[11] Since $n \ge \max\{\ell, d\}$, we know $k = \Theta\left(\log\left(n^\ell/p\right)\right) = O(n^2)$ and $D \le 2d \cdot k = O(n^3)$. Hence $n' = \max\{\tau, k, 6D, n\} = O(n^3)$. Also $n^\ell/\varepsilon \le n^{O(\ell)}/p$ and by our choice of $k = \Theta\left(\log(n^\ell/p)\right)$ we have $\left(n^\ell/\varepsilon\right)^{6/k} = O(1)$.

6 Fourier Bounds for Noisy Decision Trees

Let $\mathcal{T}$ be a noisy decision tree. By adding queries with zero correlation, we assume without loss of generality each root-to-leaf path in the noisy decision tree is of the same length. Let v be any node of $\mathcal{T}$. We use $\mathcal{P}_v$ to denote the uniform distribution over $\{\pm1\}^n$ *conditioning* on reaching v. Note that $\mathcal{P}_v$ is always a *product distribution*. As before, for any $S \subseteq [n]$ we define $\widehat{\mathcal{P}_v}(S) = \mathbb{E}_{x\sim\mathcal{P}_v}[x_S]$.

▷ Claim 40. Let $\mathcal{T}\colon \{\pm1\}^n \to \{0,1\}$ be a cost-d noisy decision tree. Let $v_0, \dots, v_D$ be any root-to-leaf path in $\mathcal{T}$. Define $\boldsymbol{v}^{(0)}, \dots, \boldsymbol{v}^{(D)} \in [-1,1]^n$ by setting $\boldsymbol{v}_j^{(i)} = \widehat{\mathcal{P}_{v_i}}(j)$ for each $0 \le i \le D$ and $j \in [n]$. Then for any $i \in \{0, \dots, D-1\}$, $\boldsymbol{v}_{q_{v_i}}^{(i+1)} - \boldsymbol{v}_{q_{v_i}}^{(i)}$ is a mean-zero random variable with magnitude bounded by $2 \cdot |\gamma_{v_i}|$.

Proof. Fix $i \in \{0, \dots, D-1\}$. For convenience, let $j = q_{v_i}$, $\gamma = \gamma_{v_i}$, and $\alpha = \boldsymbol{v}_j^{(i)}$. Suppose $|\gamma| = 1$ then $\left|v_j^{(i+1)} - v_j^{(i)}\right| \le 2 = 2 \cdot |\gamma_{v_i}|$ as desired. Now we turn to the case $|\gamma| < 1$.

Note that for the distribution $\mathcal{P}_{v_i}$, the measure of $x_j = 1$ (resp., $x_j = -1$) inputs is $(1+\alpha)/2$ (resp., $(1-\alpha)/2$). The measure of $x_j = 1$ (resp., $x_j = -1$) inputs that follow the edge labeled 1 is $a := (1+\alpha)(1+\gamma)/4$ (resp., $b := (1-\alpha)(1-\gamma)/4$). The total measure of inputs that take the edge labeled 1 is $a + b$ and the resulting node v_{i+1} satisfies $\boldsymbol{v}_j^{(i+1)} = (a-b)/(a+b)$. This implies that

$$\boldsymbol{v}_j^{(i+1)} = \begin{cases} \frac{\alpha+\gamma}{1+\gamma\cdot\alpha} & \text{with probability } \frac{1+\gamma\cdot\alpha}{2}, \\ \frac{\alpha-\gamma}{1-\gamma\cdot\alpha} & \text{with probability } \frac{1-\gamma\cdot\alpha}{2}. \end{cases}$$

The above calculation implies

$$\boldsymbol{v}_j^{(i+1)} - \boldsymbol{v}_j^{(i)} = \begin{cases} \gamma \cdot \frac{1-\alpha^2}{1+\gamma\cdot\alpha} & \text{with probability } \frac{1+\gamma\cdot\alpha}{2}, \\ -\gamma \cdot \frac{1-\alpha^2}{1-\gamma\cdot\alpha} & \text{with probability } \frac{1-\gamma\cdot\alpha}{2}, \end{cases}$$

and thus $\boldsymbol{v}_j^{(i+1)} - \boldsymbol{v}_j^{(i)}$ is a mean-zero random variable. Since $\alpha \in [-1,1]$ and $\gamma \in (-1,1)$, we have

$$\max\left\{\frac{1-\alpha^2}{1-\gamma\cdot\alpha}, \frac{1-\alpha^2}{1+\gamma\cdot\alpha}\right\} \le \frac{1-\alpha^2}{1-|\alpha|} = 1 + |\alpha| \le 2,$$

which implies $\left|\boldsymbol{v}_j^{(i+1)} - \boldsymbol{v}_j^{(i)}\right| \le 2 \cdot |\gamma|$. ◁

We now prove the general Fourier bounds. As before, for any $S \subseteq [n]$, let $\boldsymbol{v}_S^{(i)}$ be $\prod_{j\in S} \boldsymbol{v}_j^{(i)}$.

▶ **Lemma 41.** *There exists a universal constant τ such that the following holds. Let $\ell \ge 1$ be an integer. Let $\mathcal{T}\colon \{\pm1\}^n \to \{0,1\}$ be a cost-d noisy decision tree.*

Let $v_0, \dots, v_D$ be a random root-to-leaf path in $\mathcal{T}$. Define $\boldsymbol{v}^{(0)}, \dots, \boldsymbol{v}^{(D)} \in [-1,1]^n$ by setting $\boldsymbol{v}_j^{(i)} = \widehat{\mathcal{P}_{v_i}}(j)$ for each $0 \le i \le D$ and $j \in [n]$. Then for any sequence $a_S \in \{-1,0,1\}, S \in \binom{[n]}{\ell}$, any $\varepsilon \le 1/2$ and $t \in \{0, \dots, \ell\}$, we have

$$\mathbf{Pr}\left[\exists T \in \binom{[n]}{\ell-t}, \exists i \in [D], \left|\sum_{S\subseteq\overline{T}, |S|=t} a_{S\cup T} \cdot \boldsymbol{v}_S^{(i)}\right| \ge S(d,\ell,t,\varepsilon)\right] \le \varepsilon \cdot t,$$

where $S(d,\ell,0,\varepsilon) = 1$ and

$$S(d,\ell,t,\varepsilon) = \sqrt{(\tau\cdot d)^t \cdot \log\left(\frac{n^{\ell-t}}{\varepsilon}\right) \cdots \log\left(\frac{n^{\ell-1}}{\varepsilon}\right)} \qquad \textit{for } t \in [\ell].$$

Proof. We prove the bound by induction on t and show $\tau = 32$ suffices. The base case $t = 0$ is trivial, since for any T of size ℓ and any i, we have $\left|a_T \cdot v_\emptyset^{(i)}\right| \le 1 = S(d,\ell,0,\varepsilon)$.

Now we focus on the case $1 \le t \le \ell$. For any $T \in \binom{[n]}{\le \ell}$, define $X_T^{(0)},\dots,X_T^{(D)}$ by $X_T^{(i)} = \sum_{S\subseteq \overline{T}, |S|+|T|=\ell} a_{S\cup T}\cdot \boldsymbol{v}_S^{(i)}$. Define $\delta_T^{(i)}$ for $i \in [D]$ as follows:

$$\begin{aligned}\delta_T^{(i)} = X_T^{(i)} - X_T^{(i-1)} &= \sum_{S\subseteq\overline{T},|S|=t,S\ni q_{v_{i-1}}} a_{S\cup T}\cdot\left(\boldsymbol{v}_S^{(i)} - \boldsymbol{v}_S^{(i-1)}\right)\\ &= \left(\boldsymbol{v}_{q_{v_{i-1}}}^{(i)} - \boldsymbol{v}_{q_{v_{i-1}}}^{(i-1)}\right)\cdot \sum_{S'\subseteq\overline{T\cup\{q_{v_{i-1}}\}},|S'|=t-1} a_{S'\cup\{q_{v_{i-1}}\}\cup T}\cdot \boldsymbol{v}_S^{(i-1)}\\ &= \left(\boldsymbol{v}_{q_{v_{i-1}}}^{(i)} - \boldsymbol{v}_{q_{v_{i-1}}}^{(i-1)}\right)\cdot X_{T\cup\{q_{v_{i-1}}\}}^{(i-1)}.\end{aligned}$$

Note that by Claim 40 and conditioning on v_{i-1}, $\delta_T^{(i)}$ is a mean-zero random variable.

The induction hypothesis implies that with all but $\varepsilon\cdot(t-1)$ probability, for all $i\in[D]$ and $T' \in \binom{[n]}{\ell-t+1}$, we have $\left|X_{T'}^{(i)}\right| \le S(d,\ell,t-1,\varepsilon)$. By Claim 40, we have

$$\left|\delta_T^{(i)}\right| = \left|\boldsymbol{v}_{q_{v_{i-1}}}^{(i)} - \boldsymbol{v}_{q_{v_{i-1}}}^{(i-1)}\right|\cdot\left|X_{T\cup\{q_{v_{i-1}}\}}^{(i-1)}\right| \le 2\cdot\left|\gamma_{v_{i-1}}\right|\cdot S(d,\ell,t-1,\varepsilon).$$

Denote by $\Delta_T^{(i)} = 2\cdot\left|\gamma_{v_{i-1}}\right|\cdot S(d,\ell,t-1,\varepsilon)$. We can thus express $X_T^{(i)} = X_T^{(i-1)} + \Delta_T^{(i)}\cdot z_T^{(i)}$ where $\left|z_T^{(i)}\right| \le 1$. Then we apply Lemma 17 to the family of martingales $X_T^{(0)},\dots,X_T^{(D)}, |T| \in \binom{[n]}{\ell-t}$ with difference sequence $\delta_T^{(i)} = \Delta_T^{(i)}\cdot z_T^{(i)}$ satisfying

$$\sum_{i=1}^{D}\left(\Delta_T^{(i)}\right)^2 = 4\cdot(S(d,\ell,t-1,\varepsilon))^2\cdot\sum_{i=1}^{D}\left|\gamma_{v_{i-1}}\right|^2 \le 4d\cdot(S(d,\ell,t-1,\varepsilon))^2.$$

Hence for any $\beta \ge 0$, we have

$$\mathbf{Pr}\left[\exists T\in\binom{[n]}{\ell-t}, \exists i\in[D], \left|X_T^{(i)}\right| \ge 2\beta\cdot\sqrt{2d}\cdot S(d,\ell,t-1,\varepsilon)\right] \le \varepsilon\cdot(t-1)+2\cdot n^{\ell-t}\cdot e^{-\beta^2/2}.$$

Since $\varepsilon \le 1/2$, we can set $\beta = 2\cdot\sqrt{\log(n^{\ell-t}/\varepsilon)}$ so that $2\cdot n^{\ell-t}\cdot e^{-\beta^2/2} \le \varepsilon$, which completes the induction by noticing

$$2\beta\cdot\sqrt{2d}\cdot S(d,\ell,t-1,\varepsilon) = \sqrt{32\cdot d\cdot\log\left(\frac{n^{\ell-t}}{\varepsilon}\right)}\cdot S(d,\ell,t-1,\varepsilon) \le S(d,\ell,t,\varepsilon). \quad ◀$$

▶ **Theorem 42.** *Let $\ell \ge 1$ and $n \ge \max\{\ell, 2\}$ be integers. Let $\mathcal{T}\colon\{\pm1\}^n \to \{0,1\}$ be a cost-d noisy decision tree. Let $p = \mathbf{Pr}[\mathcal{T}(x) = 1] \in (0, 1/2]$.*[12] *Then we have*

$$\sum_{S\subseteq[n],|S|=\ell}\left|\widehat{\mathcal{T}}(S)\right| \le p\cdot O(d)^{\ell/2}\cdot\sqrt{\log\left(\frac{1}{p}\right)\left(\log\left(\frac{n^\ell}{p}\right)\right)^{\ell-1}} = O(d)^{\ell/2}\cdot\sqrt{1+(\ell\log(n))^{\ell-1}}.$$

Proof. For any $S\in\binom{[n]}{\ell}$, let $a_S = \mathsf{sgn}\left(\widehat{\mathcal{T}}(S)\right)$. Let $v_0,\dots,v_D$ be a random root-to-leaf path in $\mathcal{T}$. Note that

$$\sum_S\left|\widehat{\mathcal{T}}(S)\right| = \sum_S a_S\cdot\widehat{\mathcal{T}}(S) = \mathbb{E}\left[\mathcal{T}(v_D)\cdot\sum_S a_S\cdot\boldsymbol{v}_S^{(D)}\right] \le \mathbb{E}\left[\mathcal{T}(v_D)\cdot|V|\right], \tag{14}$$

[12] If $p > 1/2$, then we can consider $\widetilde{\mathcal{T}} = 1 - \mathcal{T}$ by symmetry.

where $V = \sum_S a_S \cdot_S \boldsymbol{v}_S^{(D)}$. By Lemma 41, we know $\mathbf{Pr}\left[|V| \ge S(\varepsilon)\right] \le \varepsilon \cdot \ell$, where

$$S(\varepsilon) = S(d,\ell,\ell,\varepsilon) = \sqrt{O(d)^\ell \cdot \log\left(\tfrac{n^{\ell-1}}{\varepsilon}\right)\cdots\log\left(\tfrac{n^0}{\varepsilon}\right)} \le \sqrt{O(d)^\ell \cdot \left(\log\left(\tfrac{n^{\ell-1}}{\varepsilon}\right)\right)^{\ell-1}\log\left(\tfrac{1}{\varepsilon}\right)}.$$

For integer $i \ge 1$, let $I_i = \left[S\left(p/\left(\ell 2^i\right)\right), S\left(p/\left(\ell 2^{i+1}\right)\right)\right]$ and $I_0 = [0, S(p/\ell)]$ be intervals. Then for each $i \ge 1$, $\mathbf{Pr}\left[|V| \in I_i\right] \le p/2^i$. We also know that $\mathbb{E}_{v_0,\dots,v_D}\left[\mathcal{T}(v_D)\right] \le p$. Thus,

$$\begin{aligned}
(14) &\le \mathop{\mathbb{E}}_{v_0,\dots,v_D}\left[\mathcal{T}(v_D)\cdot|V|\cdot\sum_{i=0}^{+\infty}1_{|V|\in I_i}\right]\\
&\le S\left(\frac{p}{\ell}\right)\cdot\mathbb{E}\left[\mathcal{T}(v_D)\right] + \sum_{i=1}^{+\infty} S\left(\frac{p}{\ell\cdot 2^{i+1}}\right)\cdot\mathbb{E}\left[1_{|V|\in I_i}\right]\\
&\le \sum_{i=0}^{+\infty} S\left(\frac{p}{\ell\cdot 2^{i+1}}\right)\cdot\frac{p}{2^i}\\
&= \sum_{i=0}^{+\infty} p\cdot\sqrt{O(d)^\ell\cdot\left(\log\left(\tfrac{n^{\ell-1}\cdot\ell}{p}\right)+i+1\right)^{\ell-1}\cdot\left(\log\left(\tfrac{1}{p}\right)+\log(\ell)+i+1\right)}\cdot\frac{1}{2^i}\\
&\le \sum_{i=0}^{+\infty} p\cdot\sqrt{O(d)^\ell\cdot\left(\left(\log\left(\tfrac{n^\ell}{p}\right)\right)^{\ell-1}+(i+1)^{\ell-1}\right)\cdot\left(\log\left(\tfrac{1}{p}\right)+i+1\right)}\cdot\frac{1}{2^i}\\
&\quad\text{(since } n\ge\ell\text{, and } (x+y)^b\le 2^b\cdot\left(x^b+y^b\right)\text{ and }\sqrt{x+y}\le\sqrt{x}+\sqrt{y}\text{ for } x,y,b\ge 0)\\
&\le p\cdot\sqrt{O(d)^\ell\cdot\log\left(\tfrac{1}{p}\right)\left(\log\left(\tfrac{n^\ell}{p}\right)\right)^{\ell-1}},
\end{aligned}$$

where the last inequality follows from $p \le 1/2$, $n \ge 2$ and

$$\sum_{i=0}^{+\infty}(i+1)^{\ell/2}\cdot 2^{-i} = O(\ell)^{\ell/2} \le O(1)^\ell\cdot\ell^{(\ell-1)/2} \le O(1)^\ell\cdot\left(\log\left(n^\ell/p\right)\right)^{(\ell-1)/2}.$$

Note that $p\cdot(\log(1/p))^k \le O(k)^k$ for $p\in(0,1)$ and $k \ge 0$, thus

$$\begin{aligned}
p\cdot\sqrt{\log\left(\tfrac{1}{p}\right)\left(\log\left(\tfrac{n^\ell}{p}\right)\right)^{\ell-1}} &= p\cdot\sqrt{\log\left(\tfrac{1}{p}\right)\left(\ell\log(n)+\log\left(\tfrac{1}{p}\right)\right)^{\ell-1}}\\
&\le O(1)^\ell\cdot\left(\sqrt{(\ell\log(n))^{\ell-1}}+\ell^{\ell/2}\right)\\
&= O(1)^\ell\cdot\sqrt{1+(\ell\log(n))^{\ell-1}}.
\end{aligned}$$

◀

References

1 Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. *SIAM J. Comput.*, 47(3):982–1038, 2018.

2 Nikhil Bansal and Makrand Sinha. k-forrelation optimally separates quantum and classical query complexity. *Electron. Colloquium Comput. Complex.*, 27:127, 2020.

3 Shalev Ben-David and Eric Blais. A tight composition theorem for the randomized query complexity of partial functions: Extended abstract. In *FOCS*, pages 240–246. IEEE, 2020.

4 Eric Blais, Li-Yang Tan, and Andrew Wan. An inequality for the fourier spectrum of parity decision trees. *CoRR*, abs/1506.01055, 2015.

5 Aline Bonami. Étude des coefficients de fourier des fonctions de $l^p(g)$. *Annales de l'institut Fourier*, 20(2):335–402, 1970. URL: `http://eudml.org/doc/74019`.

6 Joseph Briggs and Wesley Pegden. Extremal collections of k-uniform vectors. *arXiv preprint*, 2018. arXiv:1801.09609.

7 Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting for BPP using inner product. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 35:1–35:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.35.

8 Eshan Chattopadhyay, Jason Gaitonde, Chin Ho Lee, Shachar Lovett, and Abhishek Shetty. Fractional pseudorandom generators from any fourier level. *CoRR*, abs/2008.01316, 2020. arXiv:2008.01316.

9 Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. *Theory Comput.*, 15:1–26, 2019.

10 Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. Pseudorandom generators from the second fourier level and applications to AC0 with parity gates. In *ITCS*, volume 124 of *LIPIcs*, pages 22:1–22:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

11 Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *STOC*, pages 363–375. ACM, 2018.

12 Gil Cohen, Noam Peri, and Amnon Ta-Shma. Expander random walks: A fourier-analytic approach. *Electron. Colloquium Comput. Complex.*, 27:163, 2020.

13 Gil Cohen and Igor Shinkar. The complexity of DNF of parities. In *ITCS*, pages 47–58. ACM, 2016.

14 Dmitry Gavinsky. Entangled simultaneity versus classical interactivity in communication complexity. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 877–884, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2897518.2897545.

15 Uma Girish, Ran Raz, and Avishay Tal. Quantum versus randomized communication complexity, with efficient players. In *ITCS*, volume 185 of *LIPIcs*, pages 54:1–54:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

16 Uma Girish, Ran Raz, and Wei Zhan. Lower bounds for XOR of forrelations. *Electron. Colloquium Comput. Complex.*, 27:101, 2020.

17 Parikshit Gopalan, Rocco A. Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions. *Electron. Colloquium Comput. Complex.*, 23:69, 2016.

18 Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for XOR functions. *SIAM J. Comput.*, 47(1):208–217, 2018.

19 Joshua Brown Kramer. On the most weight w vectors in a dimension k binary code. *Electron. J. Comb.*, 17(1), 2010.

20 Raghav Kulkarni, Youming Qiao, and Xiaoming Sun. On the power of parity queries in boolean decision trees. In *TAMC*, volume 9076 of *Lecture Notes in Computer Science*, pages 99–109. Springer, 2015.

21 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993.

22 Chin Ho Lee. Fourier bounds and pseudorandom generators for product tests. In *Computational Complexity Conference*, volume 137 of *LIPIcs*, pages 7:1–7:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

23 Nikhil S. Mande and Swagato Sanyal. On parity decision trees for fourier-sparse boolean functions. In *FSTTCS*, volume 182 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

24 Yishay Mansour. An o(nˆ(log log n)) learning algorithm for DNF under the uniform distribution. *J. Comput. Syst. Sci.*, 50(3):543–550, 1995.

25 Ashley Montanaro, Harumichi Nishimura, and Rudy Raymond. Unbounded-error quantum query complexity. *Theor. Comput. Sci.*, 412(35):4619–4628, 2011. doi:10.1016/j.tcs.2011.04.043.

26 Ashley Montanaro and Tobias Osborne. On the communication complexity of XOR functions. *CoRR*, abs/0909.3392, 2009. arXiv:0909.3392.

27 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.

28 Ryan O'Donnell. Open problems in analysis of boolean functions. *CoRR*, abs/1204.6447, 2012. arXiv:1204.6447.

29 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

30 Ryan O'Donnell and Rocco A. Servedio. Learning monotone decision trees in polynomial time. *SIAM J. Comput.*, 37(3):827–844, 2007.

31 Ryan O'Donnell, John Wright, Yu Zhao, Xiaorui Sun, and Li-Yang Tan. A composition theorem for parity kill number. In *Computational Complexity Conference*, pages 144–154. IEEE Computer Society, 2014.

32 Omer Reingold, Thomas Steinke, and Salil P. Vadhan. Pseudorandomness for regular branching programs via fourier analysis. In *APPROX-RANDOM*, volume 8096 of *Lecture Notes in Computer Science*, pages 655–670. Springer, 2013.

33 Swagato Sanyal. Fourier sparsity and dimension. *Theory Comput.*, 15:1–13, 2019.

34 Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. An optimal separation of randomized and quantum query complexity. *Electron. Colloquium Comput. Complex.*, 27:128, 2020.

35 Amir Shpilka, Avishay Tal, and Ben lee Volk. On the structure of boolean functions with small spectral norm. *Comput. Complex.*, 26(1):229–273, 2017.

36 Avishay Tal. Tight bounds on the fourier spectrum of AC0. In *Computational Complexity Conference*, volume 79 of *LIPIcs*, pages 15:1–15:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

37 Avishay Tal. Towards optimal separations between quantum and randomized query complexities. In *FOCS*, pages 228–239. IEEE, 2020.

38 Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *FOCS*, pages 658–667. IEEE Computer Society, 2013.

39 Zhiqiang Zhang and Yaoyun Shi. Communication complexities of symmetric XOR functions. *Quantum Inf. Comput.*, 9(3&4):255–263, 2009.

40 Zhiqiang Zhang and Yaoyun Shi. On the parity complexity measures of boolean functions. *Theor. Comput. Sci.*, 411(26-28):2612–2618, 2010.

A Proof of Corollary 8

▶ **Corollary** (Corollary 8 restated). *Let $\mathcal{T}$ be a parity decision tree of size at most $s > 1$ on n variables. Then,*

$$\forall \ell \in [n] : L_{1,\ell}(f) \leq (\log(s))^{\ell/2} \cdot O(\ell \cdot \log(n))^{1.5\ell}.$$

Proof. We approximate $\mathcal{T}$ with error $\varepsilon = 1/n^{\ell}$ by another parity decision tree $\mathcal{T}'$ of depth $d = \lceil \log\left(s \cdot n^{\ell}\right) \rceil$, where we simply replace all nodes of depth d in $\mathcal{T}$ with leaves that return 0. Since there are at most s nodes in $\mathcal{T}$, the probability that a random input would reach one of the nodes of depth d is at most $2^{-d} \cdot s \leq 1/n^{\ell}$. Hence $\mathbf{Pr}_x\left[\mathcal{T}(x) \neq \mathcal{T}'(x)\right] \leq \varepsilon$. This implies that $\left|\widehat{\mathcal{T}}(S) - \widehat{\mathcal{T}'}(S)\right| \leq \varepsilon$ for any subset $S \subseteq [n]$. Thus,

$$L_{1,\ell}(\mathcal{T}) = \sum_{S:|S|=\ell} \left|\widehat{\mathcal{T}}(S)\right| \leq \sum_{S:|S|=\ell} \left(\left|\widehat{\mathcal{T}'}(S)\right| + \varepsilon\right) \leq L_{1,\ell}(\mathcal{T}') + 1.$$

Since $\mathcal{T}'$ is of depth at most $d = \lceil \log(s) + \ell \cdot \log(n) \rceil = O\left(\log(s) \cdot \ell \cdot \log(n)\right)$, we obtain our bound. ◀

B Proof of Lemma 16

We will use the definition of sub-Gaussian random variables.

▶ **Definition 43** (Sub-Gaussian random variables). *We say a random variable x is Δ-sub-Gaussian if $\mathbb{E}\left[e^{t\cdot x}\right] \le e^{t^2\Delta^2}$ holds for all $t \in \mathbb{R}$.*

Now we prove the following sub-Gaussian adaptive Azuma's inequality.

▶ **Lemma 44** (Sub-Gaussian adaptive Azuma's inequality). *Let $X^{(0)},\dots,X^{(D)}$ be a martingale with respect to a filtration $\left(\mathcal{F}^{(i)}\right)_{i=0}^{D}$*[13] *and $\Delta^{(1)},\dots,\Delta^{(D)}$ be a sequence of magnitudes such that $X^{(0)}=0$ and $X^{(i)}=X^{(i-1)}+\delta^{(i)}$ for $i\in[D]$, where if conditioning on $\mathcal{F}^{(i-1)}$, $\delta^{(i)}$ is a $\Delta^{(i)}$-sub-Gaussian random variable and $\Delta^{(i)}$ is a fixed value.*

If there exists some constant $U \ge 0$ such that $\sum_{i=1}^{D}\left|\Delta^{(i)}\right|^2 \le U$ always holds, then for any $\beta \ge 0$ we have

$$\mathbf{Pr}\left[\max_{i=0,1,\dots,D}\left|X^{(i)}\right| \ge \beta\cdot\sqrt{2U}\right] \le 2\cdot e^{-\beta^2/2}.$$

Proof. The bound holds trivially when $\beta=0$, hence we assume $\beta>0$ from now on. We construct another martingale $\widehat{X}^{(0)},\dots,\widehat{X}^{(D)}$ as follows:

$$\widehat{X}^{(i)} = \begin{cases} X^{(i)} & 0\le i\le d,\\ X^{(d)} & i>d,\end{cases} \qquad \text{where} \quad d=\min\left\{D\right\}\cup\left\{i\in\{0,1\dots,D\}\,\middle|\,\left|X^{(i)}\right|\ge\beta\cdot\sqrt{2U}\right\}.$$

We write $\widehat{\delta}^{(i)}=\widehat{X}^{(i)}-\widehat{X}^{(i-1)}$, then $\widehat{\delta}^{(i)}=\delta^{(i)}$ for all $i\le d$; and $\widehat{\delta}^{(i)}\equiv 0$ for all $i>d$. Let $\widehat{\Delta}^{(i)}=\Delta^{(i)}$ for all $i\le d$; and $\widehat{\Delta}^{(i)}\equiv 0$ for all $i>d$. Thus $\widehat{\delta}^{(i)}$ is $\widehat{\Delta}^{(i)}$-sub-Gaussian given $\mathcal{F}^{(i-1)}$; and

$$\sum_{i=1}^{D}\left|\widehat{\Delta}^{(i)}\right|^2=\sum_{i=1}^{d}\left|\Delta^{(i)}\right|^2\le U.$$

Moreover, we have

$$\mathbf{Pr}\left[\max_{i=0,1,\dots,D}\left|X^{(i)}\right|\ge\beta\cdot\sqrt{2U}\right]=\mathbf{Pr}\left[\left|\widehat{X}^{(D)}\right|\ge\beta\cdot\sqrt{2U}\right].$$

Let $t>0$ be a parameter and we bound $\mathbb{E}\left[e^{t\cdot\widehat{X}^{(D)}}\right]$ as follows

$$\mathbb{E}\left[e^{t\cdot\widehat{X}^{(D)}}\right]=\mathop{\mathbb{E}}_{\mathcal{F}^{(D-1)}}\left[e^{t\cdot\widehat{X}^{(D-1)}}\cdot\mathop{\mathbb{E}}_{\mathcal{F}^{(D)}}\left[e^{t\cdot\left(\widehat{X}^{(D)}-\widehat{X}^{(D-1)}\right)}\,\middle|\,\mathcal{F}^{(D-1)}\right]\right] \tag{15}$$

$$=\mathop{\mathbb{E}}_{\mathcal{F}^{(D-1)}}\left[e^{t\cdot\widehat{X}^{(D-1)}}\cdot\mathop{\mathbb{E}}_{\mathcal{F}^{(D)}}\left[e^{t\cdot\widehat{\delta}^{(D)}}\,\middle|\,\mathcal{F}^{(D-1)}\right]\right] \tag{16}$$

[13] $\mathcal{F}^{(0)}\subseteq\mathcal{F}^{(1)}\subseteq\cdots\subseteq\mathcal{F}^{(D)}$ is an increasing sequence of σ-algebra where each $\mathcal{F}^{(i)}$ makes $X^{(0)},\dots,X^{(i+1)}$ measurable and $\mathbb{E}\left[X^{(i)}\,\middle|\,\mathcal{F}^{(i-1)}\right]=X^{(i-1)}$. Intuitively, the filtration is the history of the martingale.

$$
\begin{aligned}
&\le \mathop{\mathbb{E}}_{\mathcal{F}^{(D-1)}}\left[e^{t\cdot\widehat{X}^{(D-1)}}\cdot e^{t^2\left(\widehat{\Delta}^{(D)}\right)^2}\right] && \text{(since $\widehat{\delta}^{(D)}$ is $\widehat{\Delta}^{(D)}$-sub-Gaussian)}\\
&\le \mathop{\mathbb{E}}_{\mathcal{F}^{(D-1)}}\left[e^{t\cdot\widehat{X}^{(D-1)}}\cdot e^{t^2\left(U-\left(\widehat{\Delta}^{(1)}\right)^2-\cdots-\left(\widehat{\Delta}^{(D-1)}\right)^2\right)}\right]\\
&\le \mathop{\mathbb{E}}_{\mathcal{F}^{(D-2)}}\left[e^{t\cdot\widehat{X}^{(D-2)}}\cdot e^{t^2\left(U-\left(\widehat{\Delta}^{(1)}\right)^2-\cdots-\left(\widehat{\Delta}^{(D-1)}\right)^2\right)}e^{t^2\left(\widehat{\Delta}^{(D-1)}\right)^2}\right]\\
& && \text{(similar to (15) and (16))}\\
&= \mathop{\mathbb{E}}_{\mathcal{F}^{(D-2)}}\left[e^{t\cdot\widehat{X}^{(D-2)}}\cdot e^{t^2\left(U-\left(\widehat{\Delta}^{(1)}\right)^2-\cdots-\left(\widehat{\Delta}^{(D-2)}\right)^2\right)}\right]\\
&\le\cdots\le \mathop{\mathbb{E}}_{\mathcal{F}^{(D-k)}}\left[e^{t\cdot\widehat{X}^{(D-k)}}\cdot e^{t^2\left(U-\left(\widehat{\Delta}^{(1)}\right)^2-\cdots-\left(\widehat{\Delta}^{(D-k)}\right)^2\right)}\right]\le\cdots\\
&\le e^{t^2U}. && (17)
\end{aligned}
$$

Setting $t=\beta/\sqrt{2U}$ implies that

$$
\mathbf{Pr}\left[\widehat{X}^{(D)}\ge\beta\cdot\sqrt{2U}\right]\le\frac{\mathbb{E}\left[e^{t\cdot\widehat{X}^{(D)}}\right]}{e^{t\cdot\beta\cdot\sqrt{2U}}}\le\frac{e^{t^2U}}{e^{\beta^2}}=e^{-\beta^2/2}.
$$

Similarly we can show $\mathbf{Pr}\left[\widehat{X}^{(D)}\le-\beta\cdot\sqrt{2U}\right]\le e^{-\beta^2/2}$, which completes the proof by a union bound. ◀

For our applications, we need the following fact.

▶ **Fact 45.** *Let x be a mean-zero random variable and assume $|x|\le\Delta$ always holds. Then x is Δ-sub-Gaussian.*

Proof. Note that $e^{t\cdot x}$ is convex for all $t\in\mathbb{R}$. By Jensen's inequality, we have

$$
\mathbb{E}\left[e^{t\cdot x}\right]\le\frac{1}{2}\left(e^{-t\Delta}+e^{t\Delta}\right)=\sum_{i=0}^{+\infty}\frac{(t\Delta)^{2i}}{(2i)!}\le\sum_{i=0}^{+\infty}\frac{(t\Delta)^{2i}}{i!}=e^{t^2\Delta^2}. \quad ◀
$$

As a corollary of Lemma 44 and Fact 45, we obtain Lemma 16.

▶ **Corollary** (Lemma 16 restated). *Let $X^{(0)},\ldots,X^{(D)}$ be a martingale and $\Delta^{(1)},\ldots,\Delta^{(D)}$ be a sequence of magnitudes such that $X^{(0)}=0$ and $X^{(i)}=X^{(i-1)}+\Delta^{(i)}\cdot z^{(i)}$ for $i\in[D]$, where if conditioning on $z^{(1)},\ldots,z^{(i-1)}$,*

(1) *$z^{(i)}$ is a mean-zero random variable and $\left|z^{(i)}\right|\le 1$ always holds;*

(2) *$\Delta^{(i)}$ is a fixed value.*

If there exists some constant $U\ge 0$ such that $\sum_{i=1}^{D}\left|\Delta^{(i)}\right|^2\le U$ always holds, then for any $\beta\ge 0$ we have

$$
\mathbf{Pr}\left[\max_{i=0,1,\ldots,D}\left|X^{(i)}\right|\ge\beta\cdot\sqrt{2U}\right]\le 2\cdot e^{-\beta^2/2}.
$$

C Proof of Claim 35

▷ Claim (Claim 35 restated). $\mathbf{Pr}\left[\mathcal{E}_2\right]\le\varepsilon/3$, where $\mathcal{E}_2$ is the following event: $\exists T\in\binom{[n]}{\ell-t}, i, r, r'$, such that

$$
\left|\Gamma_T^{(i)}(r,r')\right|\ge\left(100\min\left\{k,\log\left(\tfrac{n^\ell}{\varepsilon}\right)\right\}\cdot\left(\tfrac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}}\right)^{\frac{r+r'}{2}}\cdot\sigma_T(r,r',C(v_{i-1}),i).
$$

Proof. Let $k' = \min\left\{k, \left\lceil 6\log\left(n^\ell/\varepsilon\right)\right\rceil\right\} \le 12\min\left\{k, \log\left(n^\ell/\varepsilon\right)\right\}$. Then $\mathcal{T}$ is also a depth-D $2k'$-clean parity decision tree. Observe that

$$\begin{aligned}
&\mathbf{Pr}\left[\left|\Gamma_T^{(i)}(r,r')\right| \ge \left(\frac{4k'}{\eta^{2/k'}}\right)^{(r+r')/2} \cdot \sigma_T(r,r',C(v_{i-1}),i)\right] \\
&\le \max_{C(v_{i-1})} \mathbf{Pr}\left[\left|\Gamma_T^{(i)}(r,r')\right| \ge \left(\frac{4k'}{\eta^{2/k'}}\right)^{(r+r')/2} \cdot \sigma_T(r,r',C(v_{i-1}),i) \,\middle|\, C(v_{i-1})\right] \\
&\le \underbrace{\frac{(4\cdot k')^{r+r'}}{(2\cdot(r+r'))^{k'}}}_{\le 1} \cdot \underbrace{\eta^{2-\frac{2(r+r')}{k'}}}_{\le \eta} \\
&\qquad\qquad\qquad \text{(due to the second bound in Lemma 15 and } k \ge 4\cdot\ell \ge 4\cdot(r+r')\text{)} \\
&\le \eta.
\end{aligned}$$

Thus by union bound over all $T \in \binom{[n]}{\ell-t}, i \in [D'], r \in [t], 0 \le r' \le t-r$, we have

$$\mathbf{Pr}\left[\exists T, i, r, r',\ \left|\Gamma_T^{(i)}(r,r')\right| \ge \left(\tfrac{4k}{\eta^{2/k}}\right)^{(r+r')/2} \cdot \sigma_T(r,r',C(v_{i-1}),i)\right] \le Dt^2 n^{\ell-t} \cdot \eta \le \tfrac{n^{3\cdot\ell}\cdot\eta}{3},$$

where we use the fact $n \ge \max\{D, 3\cdot t\}$ and $t \ge 1$. By setting $\eta = \varepsilon/n^{3\cdot\ell}$, we have

$$\frac{4k'}{\eta^{2/k'}} = 4k'\left(\frac{n^{3\cdot\ell}}{\varepsilon}\right)^{\frac{2}{k'}} \le 4k'\left(\frac{n^\ell}{\varepsilon}\right)^{\frac{6}{k'}} \le 4\cdot 12\min\left\{k, \log\left(\frac{n^\ell}{\varepsilon}\right)\right\}\cdot 2\left(\frac{n^\ell}{\varepsilon}\right)^{\frac{6}{k}},$$

as desired. ◁

D Proof of Claim 37

We first need the following simple bound on M.

▶ **Lemma 46.** *For any integer $s \ge 1$, we have*

$$\sum_{r=s}^{t} M(D,d,k,\ell,t-r,\varepsilon) \le \frac{2\cdot M(D,d,k,\ell,t,\varepsilon)}{\left(\tau D\cdot\log\left(n^\ell/\varepsilon\right)\right)^{s/2}}.$$

Proof. We simply expand the formula of M as follows:

$$\begin{aligned}
\frac{\sum_{r=s}^{t} M(D,d,k,\ell,t-r,\varepsilon)}{M(D,d,k,\ell,t,\varepsilon)} &= \sum_{r=s}^{t}\left(\tau\cdot(D+dk)\cdot\left(\tfrac{n^\ell}{\varepsilon}\right)^{6/k}\log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{-r/2} \\
&\le \sum_{r=s}^{+\infty}\left(\tau\cdot(D+dk)\cdot\left(\tfrac{n^\ell}{\varepsilon}\right)^{6/k}\log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{-r/2} \\
&\le 2\cdot\left(\tau\cdot(D+dk)\cdot\left(\tfrac{n^\ell}{\varepsilon}\right)^{6/k}\log\left(\tfrac{n^\ell}{\varepsilon}\right)\right)^{-s/2} \\
&\qquad\qquad\qquad \text{(due to } \tau \ge 4 \text{ and } s \ge 1\text{)} \\
&\le 2\cdot\left(\tau D\cdot\log\left(n^\ell/\varepsilon\right)\right)^{-s/2}.
\end{aligned}$$

◀

Now we prove Claim 37.

▷ Claim (Claim 37 restated). When $\mathcal{E}_1 \vee \mathcal{E}_2$ does not happen, $\sum_{i=1}^{D} \left|\mu_T^{(i)}\right| \le R$ and $\sum_{i=1}^{D} \left|\delta_T^{(i)}\right|^2 \le R^2$ hold for all $T \in \binom{[n]}{\ell-t}$, where

$$R = \frac{M(D,d,k,\ell,t,\varepsilon)}{5 \cdot \sqrt{\log\left(n^\ell/\varepsilon\right)}}.$$

Proof. We verify for each $T \in \binom{[n]}{\ell-t}$ as follows:

$$\begin{aligned}
&\sum_{i=1}^{D} \left|\mu_T^{(i)}\right| \\
&= \sum_{i=1}^{D'} \left|\mu_T^{(i)}\right| \le \sum_{i=1}^{D'} \sum_{\substack{r=2,\\ \text{even}}}^{t} |A(T,r,i)| && \text{(due to (8))} \\
&\le \sum_{i=1}^{D'} \sum_{\substack{r=2,\\ \text{even}}}^{t} \left(M(D,d,k,\ell,t-r,\varepsilon) + \tfrac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log(n^\ell/\varepsilon)}} \cdot \sqrt{\left(\tfrac{800}{\tau}\right)^r \left(\tfrac{|J(v_{i-1})|}{2d}\right)^r} \cdot 1_{|J(v_{i-1})|>1} \right) \\
& && \text{(due to (10))} \\
&\le \sum_{i=1}^{D'} \sum_{\substack{r=2,\\ \text{even}}}^{t} \left(M(D,d,k,\ell,t-r,\varepsilon) + \tfrac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log(n^\ell/\varepsilon)}} \cdot \left(\tfrac{|J(v_{i-1})|}{2d}\right) \left(\tfrac{800}{\tau}\right)^{r/2} \cdot 1_{|J(v_{i-1})|>1} \right) \\
& && \text{(Since } |J(v_{i-1})| \le 2d \text{ from Corollary 26)} \\
&\le \tfrac{2 \cdot M(D,d,k,\ell,t,\varepsilon)}{\tau \cdot \log(n^\ell/\varepsilon)} + \tfrac{1.1 \cdot 800 \cdot M(D,d,k,\ell,t,\varepsilon)}{\tau \cdot \sqrt{\log(n^\ell/\varepsilon)}} \\
& && \text{(due to Lemma 46 and Corollary 26 and } \tau = 10^4) \\
&\le \tfrac{M(D,d,k,\ell,t,\varepsilon)}{5 \cdot \sqrt{\log(n^\ell/\varepsilon)}} = R
\end{aligned}$$

and with similar calculation, we have

$$\begin{aligned}
&\sum_{i=1}^{D} \left|\delta_T^{(i)}\right|^2 \\
&\le \sum_{i=1}^{D'} \left(\sum_{\substack{r=1,\\ \text{odd}}}^{t} \left(M(D,d,k,\ell,t-r,\varepsilon) + \tfrac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log(n^\ell/\varepsilon)}} \cdot \sqrt{\tfrac{|J(v_{i-1})|}{2d}} \left(\tfrac{800}{\tau}\right)^{r/2} \cdot 1_{|J(v_{i-1})|>1} \right) \right)^2 \\
&\le \sum_{i=1}^{D'} \left(\tfrac{2 \cdot M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\tau D \cdot \log(n^\ell/\varepsilon)}} + \tfrac{1.1 \cdot \sqrt{800} \cdot M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\tau}\sqrt{\log(n^\ell/\varepsilon)}} \cdot \sqrt{\tfrac{|J(v_{i-1})|}{2d}} \cdot 1_{|J(v_{i-1})|>1} \right)^2 \\
& && \text{(due to } \tau = 10^4) \\
&\le \left(\tfrac{M(D,d,k,\ell,t,\varepsilon)}{\sqrt{\log(n^\ell/\varepsilon)}} \right)^2 \sum_{i=1}^{D'} 2 \cdot \left(\frac{4}{\tau D} + \tfrac{968}{\tau} \cdot \tfrac{|J(v_{i-1})|}{2d} \cdot 1_{|J(v_{i-1})|>1} \right) \\
& && \text{(due to } (a+b)^2 \le 2(a^2+b^2)) \\
&\le \left(\tfrac{2000 \cdot M(D,d,k,\ell,t,\varepsilon)}{\tau \cdot \sqrt{\log(n^\ell/\varepsilon)}} \right)^2 = R^2.
\end{aligned}$$

◁

The Power of Negative Reasoning

Susanna F. de Rezende
Institute of Mathematics of the Czech Academy of Sciences, Prague, Czech Republic

Massimo Lauria
Sapienza Università di Roma, Italy

Jakob Nordström
University of Copenhagen, Denmark
Lund University, Sweden

Dmitry Sokolov
St. Petersburg State University, Russia
PDMI RAS, St. Petersburg, Russia

Abstract

Semialgebraic proof systems have been studied extensively in proof complexity since the late 1990s to understand the power of Gröbner basis computations, linear and semidefinite programming hierarchies, and other methods. Such proof systems are defined alternately with only the original variables of the problem and with special formal variables for positive and negative literals, but there seems to have been no study how these different definitions affect the power of the proof systems. We show for Nullstellensatz, polynomial calculus, Sherali-Adams, and sums-of-squares that adding formal variables for negative literals makes the proof systems exponentially stronger, with respect to the number of terms in the proofs. These separations are witnessed by CNF formulas that are easy for resolution, which establishes that polynomial calculus, Sherali-Adams, and sums-of-squares cannot efficiently simulate resolution without having access to variables for negative literals.

2012 ACM Subject Classification Theory of computation → Proof complexity; Computing methodologies → Representation of polynomials

Keywords and phrases Proof complexity, Polynomial calculus, Nullstellensatz, Sums-of-squares, Sherali-Adams

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.40

Funding Susanna F. de Rezende was supported by Knut and Alice Wallenberg grant KAW 2018.0371. Jakob Nordström received funding from the Swedish Research Council grant 2016-00782 and the Independent Research Fund Denmark grant 9040-00389B. Part of this work was carried out while visiting the Simons Institute for the Theory of Computing.

Acknowledgements We thank Or Meir for fruitful discussions and the anonymous reviewers for comments on the presentation.

1 Introduction

Given a set of polynomial equalities

$$p_j = 0 \qquad\qquad j \in [m] \tag{1}$$

and/or inequalities

$$r_j \geq 0 \qquad\qquad j \in [\ell] \tag{2}$$

in some field $\mathbb{F}$ (which should be ordered if $\ell > 0$), the problem of determining whether there exists solutions satisfying all constraints is a natural and well-known NP-hard problem. If one includes among the equalities (1) also equations $x_i^2 - x_i = 0$ for all variables x_i, then this setting can also be used to decide satisfiability of formulas in conjunctive normal

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 40; pp. 40:1–40:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

form (CNF). This is done by identifying 1 with *true* and 0 with *false* and then translating disjunctive clauses like $x_1 \lor \overline{x}_2 \lor x_k$ into equalities $(1-x_1)x_2(1-x_3) = 0$ or inequalities $x_1 + (1-x_2) + x_3 \geq 1$ (using the *multiplicative* or *additive* translation, respectively).

For polynomial equalities as in (1), it follows from (a mild extension of) Hilbert's Nullstellensatz that there is no solution if and only if there are polynomials q_j such that the syntactic equality

$$\sum_{j \in [m]} q_j p_j = 1 \tag{3}$$

holds. Such *Nullstellensatz certificates* can be viewed as proof system in the sense of Cook and Reckhow [15], and the study of this *Nullstellensatz* proof system was initiated in [7]. In the *polynomial calculus* proof system introduced in [14] such certificates can be constructed step by step by explicitly deriving polynomials in the ideal generated by $\{\ p_j \mid j \in [m] \}$. This can be seen to correspond to *Gröbner basis* computations, which can potentially yield more concise certificates of unsatisfiability. When there are also inequalities (2), linear combinations of polynomial products

$$\sum_{j \in [m]} q_j p_j + \sum_{j \in [\ell]} s_j r_j = -1 \tag{4}$$

for $s_j \geq 0$ with different syntactic restrictions yield proof systems such as *Sherali-Adams* [38] and *sums-of-squares (SOS)* [30, 25], corresponding to linear and semidefinite programming hierarchies.

By now there is a rich literature on upper and lower bounds for these proof systems. An excellent general reference on proof complexity is [28]. For more details on Nullstellensatz and polynomial calculus the reader can consult [13] and the references therein, and a recent survey covering Sherali-Adams and sums-of-squares is [21].

1.1 Encoding of Variables and Literals

One slightly annoying aspect when translating CNF formulas to the algebraic setting described above is that the translation is quite sensitive to the signs of the literals in clauses. Normally, polynomials are represented as linear combinations of monomials, which means that a clause

$$x_1 \lor x_2 \lor \cdots \lor x_3 \tag{5a}$$

with k positive literals turns into a polynomial equation

$$\prod_{i=1}^{k} (1 - x_i) = 0 \tag{5b}$$

with 2^k monomials if we use the multiplicative translation. This problem does not immediately arise for the additive translation, but it is still conceivable that it could be helpful to encode polynomials of the form (5b) more concisely.

This problem was perhaps first addressed in [1], where a version of polynomial calculus was defined with separate formal variables $\overline{x}_i$ for negative literals, together with equations $x_i + \overline{x}_i - 1 = 0$ enforcing the intended meaning of negation. This proof system was called *polynomial calculus with resolution*, or *PCR* for short, in [1], since the introduction of negative literals can be seen to allow polynomial calculus to simulate the *resolution* proof system efficiently, but in this paper we will refer to this flavour of the proof system as *polynomial* *calculus with negative literals* (as opposed to *polynomial calculus without negative literals*). When the proof system has access to separate variables for positive and negative literals, this ensures that lower bounds do not depend on the choice of signs for literals encoding the input, but reflect more intrinsic properties of the problem under study. As far as we are aware, essentially all lower bounds for polynomial calculus holds even when negative literals are allowed (with the exception of some of the lower bounds in [20]), and to the best of our knowledge there are no polynomial calculus upper bounds that are known to hold only for polynomial calculus with negative literals and not for polynomial calculus without them. Papers such as [5, 31, 11, 4] have studied the Sherali-Adams and sums-of-squares proof systems both with and without variables for negative literals, but again without really distinguishing between the two versions of the proof system thus obtained.

The purpose of this work is to understand if and how the introduction of formal variables for negative literals affect the power of reasoning of (semi)algebraic proof systems. This is arguably quite a natural question, and we find it somewhat surprising that nothing seems to be known regarding how the two variants of these (semi)algebraic proof systems are related.

Somewhat intriguingly, this does not seem to be just a theoretical concern. For, e.g., Gröbner basis computations, one could expect that this whole question should be irrelevant, since the basis reduction algorithm will immediately remove whichever literal over a given variable that comes later in the order. This appears not to be the case, however, and papers such as [36, 27] use “bit-flipping” (i.e., the introduction of formal variables for negated literals) to try to avoid blow-ups in polynomial size during hardware circuit verification.

1.2 Our Results

We show that for all of the proof systems Nullstellensatz, polynomial calculus, Sherali-Adams, and sums-of-squares, adding separate formal variables for negative literals results in an exponential increase in power. Our main results can be summarized as follows (where we refer to Section 2 for the missing formal definitions).

▶ **Theorem 1.** *Let $\mathcal{P}$ be any of the proof systems Nullstellensatz or polynomial calculus (over any field), or Sherali-Adams or sums-of-squares. Then there is a family of CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size polynomial in n such that the proof system $\mathcal{P}$ has polynomial size refutations of F_n that use formal variables for negative literals, whereas $\mathcal{P}$ refutations of F_n requires exponential size when such formal variables are not allowed.*

We remark that, except for sums-of-squares, the separating formulas above are CNFs of constant width. It is known from [1, 5] that polynomial calculus, Sherali-Adams, and sums-of-squares over literals can simulate the resolution proof system efficiently. Since the formulas in Theorem 1 are all easy for resolution, it follows that negative literals are necessary for the simulation.

▶ **Corollary 2.** *None of the proof systems polynomial calculus, Sherali-Adams, or sums-of-squares can polynomially simulate resolution, unless formal variables for negative literals are allowed.*

For Nullstellensatz and polynomial calculus we also give some more refined separation results involving size-degree trade-offs and space-degree trade-offs.

1.3 Outline of This Paper

In Section 2 we review the relevant preliminaries. In Section 3 we establish our separation results for polynomial calculus. Analogous results for Sherali-Adams and sums-of-squares are obtained in Section 4, and the separation for Sherali-Adams is sharpened somewhat in Section 5. Our results for Nullstellensatz are presented in Section 6.

2 Preliminaries

We encode propositional variables as algebraic variables with $\{0,1\}$ values, with the intended meaning that 1 represents true and 0 represents false. For each variable x we consider a corresponding variable $\overline{x}$ that represents the logical negation of x, i.e., it holds that $\overline{x} = 1 - x$. We say that x is a *positive literal* and $\overline{x}$ is a *negative literal*. A (partial) boolean assignment ρ is a mapping from some algebraic variables to $\{0,1\}$, with the constraint that $x \in \mathrm{dom}(\rho)$ if and only if $\overline{x} \in \mathrm{dom}(\rho)$ and that in such case $\rho(x) = 1 - \rho(\overline{x})$. Given a polynomial p *the restriction of* p *by* ρ, denoted as $p{\restriction_\rho}$, is the polynomial obtained from p by substituting in it all variables $x \in \mathrm{dom}(\rho)$ with the corresponding value $\rho(x)$. Given a set of polynomials $\mathcal{S}$ we denote as $\mathcal{S}{\restriction_\rho}$ the set of restricted polynomials. A random restriction is a distribution over partial boolean assignments. A polynomial is *multilinear* if no variable appears with degree larger than one and no monomial contains two opposite literals.

For a set $\mathcal{S} = \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ of polynomial equations and inequalities we say that a boolean assignment satisfies $\mathcal{S}$ if it satisfies all the equations and inequalities in it. We say that $\mathcal{S}$ implies an equation $p = 0$ when every boolean assignment which satisfies $\mathcal{S}$ also satisfies $p = 0$. In the same way $\mathcal{S}$ implies an inequality $r \geq 0$ when every boolean assignment which satisfies $\mathcal{S}$ also satisfies $r \geq 0$. We now discuss encodings of a clause

$$x_1 \vee \cdots \vee x_j \vee \neg x_{j+1} \vee \cdots \vee \neg x_k \tag{6a}$$

into polynomial constraints

$$(1-x_1)\cdots(1-x_j)\cdot x_{j+1}\cdots x_k = 0 \ , \tag{6b}$$

$$\overline{x}_1 \cdots \overline{x}_j \cdot x_{j+1} \cdots x_k = 0 \ , \text{ and} \tag{6c}$$

$$x_1 + \cdots + x_j + (1 - x_{j+1}) + \cdots + (1 - x_k) \geq 1 \ . \tag{6d}$$

A clause (6a) is naturally encoded as the polynomial equation (6b), which has 2^j monomials of degree up to k. Using negative literals we get the more efficient encoding (6c) which has a single monomial. We would like to stress that (6b) and (6c) are algebraic representations of the same boolean function, even though they are syntactically different. For semi-algebraic proofs, clauses are naturally represented as inequalities (6d).

We now define all proof systems discussed in this paper.

Resolution. We first introduce some basic notation. We denote the negation of a variable x by $\neg x$ or $\overline{x}$. The *width* of a clause C is the number of literals in C. A *CNF formula* is a conjunction of clauses and a *width-k CNF formula*, or simply a *k-CNF formula*, is a CNF formula where every clause has width at most k. A resolution proof from a CNF formula F of a clause C is a sequence of clauses $(C_1, \ldots, C_\tau)$ such that $C_\tau = C$ and, for each $i \in [\tau]$, C_i is either a clause of F, or is some clause $C_j \vee D$ obtained by *weakening* a clause C_j, for some $j < i$, or is derived from C_j and $C_{j'}$, for some $j, j' < i$ by applying the *resolution rule*

$$\frac{B \vee x \qquad D \vee \neg x}{B \vee D} \ , \tag{7}$$

where $C_j = B \vee x$, $C_{j'} = D \vee \neg x$, and $C_i = B \vee D$. When applying rule (7), we say that we *resolve on* x. The *size/length* of a resolution proof $(C_1, \ldots, C_\tau)$ is τ and its *width* is the maximum width of any clause in the proof. A *resolution refutation* (i.e., proof of unsatisfiability) of F is a proof of the empty clause $\perp$ from it.

A resolution proof $(C_1, \dots, C_\tau)$ can also be viewed as a DAG, with nodes $[\tau]$ and, for all $i, j \in [\tau]$, a directed edge from j to i if C_j was used to derive C_i. The *depth* of a proof is the length of the longest directed path in the underlying DAG. If the DAG is a tree the proof is *tree-like*.

The following (semi-)algebraic proof systems reason about polynomial equations and/or inequalities over $\{0,1\}$, expressed in term of variables representing positive and negative literals. To deal with CNF formulas we use the encodings (6), plus appropriate axioms enforcing boolean values.

Nullstellensatz. Consider an initial set of polynomial equations $\mathcal{S} = \{p_1 = 0, \dots, p_m = 0\}$ over a field $\mathbb{F}$ and over variables $x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n$, where we require the set $\mathcal{S}$ to include *variable axioms* $x_i^2 - x_i = 0$, $\overline{x}_i^2 - \overline{x}_i = 0$ and $x_i + \overline{x}_i - 1 = 0$ for each $i \in [n]$. A Nullstellensatz (NS) proof of $p = 0$ from $\mathcal{S}$ is a set of polynomials $\{q_1, \dots, q_m\}$ in $\mathbb{F}[x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n]$ such that

$$\sum_{j \in [m]} q_j p_j = p \ , \tag{8}$$

where we stress that the equality is syntactical. Since all polynomials in $\mathcal{S}$ are zero by hypothesis, the proof is sound. The *(monomial) size* of any such proof is the sum over $j \in [m]$ of the number of monomials occurring in each polynomial $q_j p_j$, when expanded out as a linear combination of monomials. The *degree* of any such proof is the maximum degree among all $q_j p_j$ for $j \in [m]$. A refutation of $\mathcal{S}$ is a proof of the equation $1 = 0$. A refutation of a CNF formula F in NS is a refutation of a set $\mathcal{S}$ of polynomials containing the variable axioms specified above plus the clauses of F encoded as in (6b), unless a different encoding is specified.

▶ **Proposition 3** (NS with negative literals simulates tree-like resolution)**.** *Let F be an unsatisfiable CNF formula that has a tree-like resolution refutation of F in size s and depth d. Then the set of polynomial equations obtained by encoding each clause of F as in* (6c) *has an* NS *refutation with negative literals in size $2s - 1$ and degree d.*

Polynomial calculus. As was the case for Nullstellensatz, we consider an initial set of polynomial equations $\mathcal{S} = \{p_1 = 0, \dots, p_m = 0\}$ over a field $\mathbb{F}$ and over variables $x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n$, and we require that $\mathcal{S}$ include *variable axioms* $x_i^2 - x_i = 0$, $\overline{x}_i^2 - \overline{x}_i = 0$ and $x_i + \overline{x}_i - 1 = 0$ for each $i \in [n]$. A polynomial calculus (PC) proof of $p = 0$ from $\mathcal{S}$ is a sequence of polynomials $(q_1, q_2, \dots, q_\tau)$ in $\mathbb{F}[x_1, \dots, x_n, \overline{x}_1, \dots, \overline{x}_n]$ such that $q_\tau = p$ and each q_t for $1 \leq t \leq \tau$ is either

- some polynomial p_j with $p_j = 0 \in \mathcal{S}$;
- a *linear combination* $\alpha q_{t_1} + \beta q_{t_2}$ for some $\alpha, \beta \in \mathbb{F}$ and $1 \leq t_1, t_2 < t$;
- a *multiplication* $x \cdot q_{t'}$ for some $t' < t$ and variable $x = x_i$ or $x = \overline{x}_i$.

When the equations in $\mathcal{S}$ are satisfied, all derived polynomials, p in particular, are zero. The *(monomial) size* of such a proof is the sum over $1 \leq t \leq \tau$ of the number of monomials occurring in each polynomial q_t, when written as a sum of monomials. The *degree* of such a proof is the maximum degree among all q_t for $1 \leq t \leq \tau$. A refutation of $\mathcal{S}$ is a proof of $1 = 0$. A refutation of a CNF formula F in PC is a refutation of a set $\mathcal{S}$ of polynomials containing the variable axioms specified above plus the clauses of F encoded as in (6b), unless a different encoding is specified. It is a simple observation that when dealing with CNF formulas of constant width, it is possible to efficiently deduce the representation (6b) from the representation (6c) and vice versa.

We stress that all results proved here for NS and PC hold independently of the field $\mathbb{F}$.

Sherali-Adams. We consider an initial set of polynomial equations and inequalities $\mathcal{S} = \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ over the real field and over variables $x_1, \ldots, x_n$, and we require that the set $\mathcal{S}$ include, for each $i \in [n]$, *variable axioms* $x_i^2 - x_i = 0$, $\overline{x}_i^2 - \overline{x}_i = 0$, $x_i + \overline{x}_i - 1 = 0$, $x_i \geq 0$, $\overline{x}_i \geq 0$, $1 - x_i \geq 0$, and $1 - \overline{x}_i \geq 0$. We also assume that $\mathcal{S}$ includes the axiom $1 \geq 0$. We refer to an arbitrary product of factors of the form x_i, $\overline{x}_i$, $1 - x_i$, $1 - \overline{x}_i$ as a *generalized monomial*.[1] A Sherali-Adams (SA) proof/derivation of $r \geq 0$ from $\mathcal{S}$ is a set of polynomials $\{q_1, \ldots, q_m; s_1, \ldots, s_\ell\}$ such that

$$\sum_{j \in [m]} q_j p_j + \sum_{j \in [\ell]} s_j r_j = r \ , \tag{9}$$

where each s_j is a positive linear combination of generalized monomials. That is, s_j can be written as $s_j = \sum_i \alpha_{j,i} h_{j,i}$ for some $\alpha_{j,i}$'s that are positive real numbers and $h_{j,i}$'s that are generalized monomials. Under the assumption that all polynomial equations and inequalities in $\mathcal{S}$ are satisfied, the addends $q_j p_j$ are equal to zero and the addends $s_j r_j$ are nonnegative; hence $r \geq 0$.

The *(monomial) size* of an SA proof is the sum over $j \in [m]$ and $j \in [\ell]$ of the number of monomials occurring in each summand in (9), when written as a sum of monomials. The *degree* of an SA proof is the maximum degree among all $q_j p_j$ for $j \in [m]$ and all $s_j r_j$ for $j \in [\ell]$. An SA refutation of $\mathcal{S}$ is an SA proof of $-1 \geq 0$. A refutation of a CNF formula F in SA is a refutation of a set $\mathcal{S}$ of polynomials containing the variable axioms specified above plus the clauses of F encoded as in (6d), unless a different encoding is specified.

Sums-of-squares. As was the case for Sherali-Adams, we consider an initial set of polynomial equations and inequalities $\mathcal{S} = \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ over the real field and over variables $x_1, \ldots, x_n$, and we require that $\mathcal{S}$ include, for each $i \in [n]$, *variable axioms* $x_i^2 - x_i = 0$, $\overline{x}_i^2 - \overline{x}_i = 0$, $x_i + \overline{x}_i - 1 = 0$, $x_i \geq 0$, $\overline{x}_i \geq 0$, $1 - x_i \geq 0$, and $1 - \overline{x}_i \geq 0$, and also the axiom $1 \geq 0$. A sum of squares (SOS) proof of $r \geq 0$ from $\mathcal{S}$ is a set of polynomials $\{q_1, \ldots, q_m; s_1, \ldots, s_\ell\}$ in $\mathbb{F}[x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n]$ such that

$$\sum_{j \in [m]} q_j p_j + \sum_{j \in [\ell]} s_j r_j = r \ , \tag{10}$$

where each s_j is a positive linear combination of squared polynomials, that is, s_j can be written as $s_j = \sum_i \alpha_{j,i} h_{j,i}^2$ for some $\alpha_{j,i}$'s that are positive real numbers and $h_{j,i}$'s that are polynomials. Under the assumption that all polynomial equations and inequalities in $\mathcal{S}$ are satisfied, the summands $q_j p_j$ are equal to zero and the summands $s_j r_j$ are nonnegative; hence, $r \geq 0$.

The *(monomial) size* of an SOS proof is the sum over $j \in [m]$ and $j \in [\ell]$ of the number of monomials occurring in each summand in (10), when written as a sum of monomials. The *degree* of an SOS proof is the maximum degree among all $q_j p_j$ for $j \in [m]$ and all $s_j r_j$ for $j \in [\ell]$. An SOS refutation of $\mathcal{S}$ is an SOS proof of $-1 \geq 0$. A refutation of a CNF formula F in SOS is a refutation of a set $\mathcal{S}$ of polynomials containing the variable axioms specified above plus the clauses of F encoded as in (6d), unless a different encoding is specified.

For the rest of the paper we say a proof in either NS, PC, SA, or SOS is *without negative literals* if none of the variables $\overline{x}_1, \ldots, \overline{x}_n$ occur in any of the polynomials occurring in the proof. Otherwise we say that the proof is *with negative literals*.

[1] For instance $(1 - x_2)x_3\overline{x}_4x_5(1 - \overline{x}_9)$ is a generalized monomial. It is positive under the assumption that all variables are between 0 and 1.

▶ **Proposition 4.** *Consider a CNF formula F with a resolution refutation of length L and width w. It holds that*

- *the clauses of F, encoded as in (6c), have a* PC *refutation with negative literals of size $O(L)$ and degree $w+1$;*
- *when F is a k-CNF formula with m clauses, its representation using encoding (6b) has a* PC *refutation with negative literals of size $O(2^k m + L)$ and degree $w+1$;*
- *the clauses of F, represented using encoding (6b), have a* PC *refutation without negative literals of size $O(2^w L)$ and degree $w+1$.*

▶ **Proposition 5** ([11]). *Let $\mathcal{S} \coloneqq \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ be a set of polynomial equations and inequalities. If $\mathcal{S}$ has a Sherali-Adams refutation of degree d and size N, then it has a sums-of-squares refutation of degree $d+1$ and size N^c for some $c > 0$.*

The next lemma is a fundamental tool for the results in the next section.

▶ **Lemma 6.** *Let $\mathcal{S}$ be a set of monomials over (positive) variables $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$. There is a restriction ρ that for all $i \in [n]$ sets exactly one of $\{y_i, z_i\}$ to 0 and is such that $\mathcal{S}\restriction_\rho$ has degree at most $\log|\mathcal{S}|$.*

Proof. We consider a random restriction ρ that for each i, chooses either y_i or z_i with probability $1/2$ and sets it to 0. Note that a monomial of degree d is set to 0 by ρ with probability at least $1-(1/2)^d$. Indeed, if the monomial contains both y_i and z_i for some $i \in [n]$, then it is set to 0 with probability 1; otherwise every variable is set to 0 independently with probability $1/2$ and thus the monomials is not set to 0 with probability $(1/2)^d$. Therefore, by union bound over all monomials in $\mathcal{S}$ we have that

$$\Pr[\mathcal{S}\restriction_\rho \text{ has a monomial of degree} > \log|\mathcal{S}|\,] \;\leq\; |\mathcal{S}| \cdot (1/2)^{\log|\mathcal{S}|+1} < 1 \;. \tag{11}$$

We conclude that there is some restriction ρ such that $\mathcal{S}\restriction_\rho$ has degree at most $\log|\mathcal{S}|$. ◀

3 Negative literals and polynomial calculus

The main goal of this section is to exhibit a formula that has short refutations in resolution but requires exponential size refutations in PC without negative literals. In particular, this implies that not using negative literals can lead to an exponential blow-up in the size of refutations. The starting point is the *graph ordering principle*, a formula introduced in [37] that falsely claims that it is possible to partially order vertices of some finite graph such that each vertex has at least one neighbour that is smaller (according to the ordering) than itself.

Consider a finite undirected graph $G = (V, E)$. The graph ordering principle on G, denoted as GOP(G), is a CNF formula defined on propositional variables $x_{u,v}$ for every two distinct $u, v \in V$, with the intended meaning that $x_{u,v}$ is true when u is smaller than v in the partial order. The clauses of GOP(G) are

$$\overline{x}_{u,v} \vee \overline{x}_{v,w} \vee x_{u,w} \qquad \text{for every three distinct } u, v, w \in V, \tag{12a}$$

$$\overline{x}_{u,v} \vee \overline{x}_{v,u} \qquad \text{for every two distinct } u, v \in V, \tag{12b}$$

$$\bigvee_{u\,:\,\{u,v\}\in E} x_{u,v} \qquad \text{for every } v \in V. \tag{12c}$$

The graph ordering principle is a generalization of the *ordering principle*, considered for the first time in [29]. The latter principle falsely claims that it is possible to partially order a set of n element so that no element is minimal. The ordering principle, expressed as a

CNF formula, is often denoted by OP_n, and is exactly the formula $\mathsf{GOP}(K_n)$, where K_n is the complete graph over n vertices. Proposition 7 claims an upper bound that holds for any graph, even the complete one. The degree lower bound in Proposition 8, however, holds only for specific families of expander graphs.

▶ **Proposition 7** ([39]). *Given any graph G with n vertices and maximum degree d, the formula $\mathsf{GOP}(G)$ is a d-CNF formula with $\Theta(n^2)$ variables and $\Theta(n^3)$ clauses. Furthermore, $\mathsf{GOP}(G)$ has a resolution refutation of length $\Theta(n^3)$ where every clause in the refutation contains at most two negative literals.*

▶ **Proposition 8** ([23]). *There exists a sequence of graphs $\{G_n\}_n$ such that each G_n has $\Theta(n)$ vertices and constant maximum degree d, and any PC refutation of $\mathsf{GOP}(G_n)$ requires polynomials of degree $\Omega(n)$.*

The degree lower bound implies, in particular, that any resolution refutation of $\mathsf{GOP}(G_n)$ must have width $\Omega(n)$ (due to Proposition 4). Given the resolution upper bound in Proposition 7, the simulation of resolution in Proposition 4 gives a small PC refutation of $\mathsf{GOP}(G_n)$ only when using with negative literals. This suggests that negative literals are essential to obtain small refutations of $\mathsf{GOP}(G_n)$. Is this really the case? A positive answer would give us the separation we are looking for, but unfortunately we are not able to prove a size lower bound for refuting $\mathsf{GOP}(G_n)$ in PC without negative literals. Instead, we compose $\mathsf{GOP}(G_n)$ with the 2-bit OR function, thus obtaining a new formula that will remain easy for resolution but will be provably hard for PC without negative literals.

Let us make this construction explicit. We denote by $\mathsf{GOP}^{\mathsf{OR}}(G)$ the CNF formula obtained from $\mathsf{GOP}(G)$ by substituting each variable $x_{u,v}$ in $\mathsf{GOP}(G)$ by the disjunction of two fresh variables, $y_{u,v} \vee z_{u,v}$. In order to obtain a CNF formula, after the substitution we must apply distributivity. This process transforms a clause of width k, with j negative literals and $k-j$ positive literals, into a set of 2^j clauses with j negative literals and $2(k-j)$ positive literals. For example, see how the substitution transforms this clause with 2 negative literals

$$\overline{x}_{u_1,v_1} \vee \overline{x}_{u_2,v_2} \vee x_{u_3,v_3} \vee x_{u_4,v_4} \vee \ldots \vee x_{u_k,v_k} \ , \tag{13}$$

into four clauses

$$\overline{y}_{u_1,v_1} \vee \overline{y}_{u_2,v_2} \vee y_{u_3,v_3} \vee z_{u_3,v_3} \vee y_{u_4,v_4} \vee z_{u_4,v_4} \vee \ldots \vee y_{u_k,v_k} \vee z_{u_k,v_k} \tag{14a}$$
$$\overline{y}_{u_1,v_1} \vee \overline{z}_{u_2,v_2} \vee y_{u_3,v_3} \vee z_{u_3,v_3} \vee y_{u_4,v_4} \vee z_{u_4,v_4} \vee \ldots \vee y_{u_k,v_k} \vee z_{u_k,v_k} \tag{14b}$$
$$\overline{z}_{u_1,v_1} \vee \overline{y}_{u_2,v_2} \vee y_{u_3,v_3} \vee z_{u_3,v_3} \vee y_{u_4,v_4} \vee z_{u_4,v_4} \vee \ldots \vee y_{u_k,v_k} \vee z_{u_k,v_k} \tag{14c}$$
$$\overline{z}_{u_1,v_1} \vee \overline{z}_{u_2,v_2} \vee y_{u_3,v_3} \vee z_{u_3,v_3} \vee y_{u_4,v_4} \vee z_{u_4,v_4} \vee \ldots \vee y_{u_k,v_k} \vee z_{u_k,v_k} \ . \tag{14d}$$

This transformation has not increased the size of the formula by much: $\mathsf{GOP}^{\mathsf{OR}}(G)$ has $\Theta(n^2)$ variables, $\Theta(n^3)$ clauses, and the maximum width of its clauses is at most 2 times the maximum width of a clause in $\mathsf{GOP}(G)$. Moreover, $\mathsf{GOP}^{\mathsf{OR}}(G)$ still admits short resolution refutations.

▶ **Lemma 9.** *For every graph G with n vertices, the formula $\mathsf{GOP}^{\mathsf{OR}}(G)$ has a resolution refutation of length $\Theta(n^3)$ where every clause in the refutations contains at most two negative literals.*

Proof. The idea is to use the resolution refutation of $\mathsf{GOP}(G)$ from Proposition 7 as a scheme for the refutation of $\mathsf{GOP}^{\mathsf{OR}}(G)$. Let $C_1, C_2, \ldots, C_\tau$ be the sequence of clauses in this refutation. For each C_i we consider the set $\mathcal{C}_i$ of at most four clauses that we get by applying substitution (14) to it. Every clause in $\mathcal{C}_i$ has the same number of negative literals as C_i, and that is at most two.

We show how to derive each $\mathcal{C}_i$ from $\mathsf{GOP}^{\mathsf{OR}}(G)$ by induction on i, assuming all previous set $\mathcal{C}_j$ for $j < i$ have already been derived. Furthermore, we show that each such derivation takes a constant number of resolution steps.

If C_i is an initial clause of $\mathsf{GOP}(G_n)$ then all clauses of $\mathcal{C}_i$ are in $\mathsf{GOP}^{\mathsf{OR}}(G)$ by construction. If C_i follows from C_j for some $j < i$ by weakening, then each clause of $\mathcal{C}_i$ is a superset of some clause in $\mathcal{C}_j$ and thus follows from it by weakening. The remaining case is when C_i is derived by a resolution step from two previous clauses C_j and C_k. Without loss of generality, we rewrite clause C_j as $A \vee x_{u,v}$, clause C_k as $B \vee \overline{x}_{u,v}$, and clause C_i as $A \vee B$. The structure of sets $\mathcal{C}_j$ and $\mathcal{C}_k$ is as follows,

$$\begin{array}{ll}
\mathcal{C}_j : & \mathcal{C}_k : \\
A_1 \vee y_{u,v} \vee z_{u,v} & \overline{y}_{u,v} \vee B_1 \\
A_2 \vee y_{u,v} \vee z_{u,v} & \overline{y}_{u,v} \vee B_2 \\
A_3 \vee y_{u,v} \vee z_{u,v} & \overline{z}_{u,v} \vee B_1 \\
A_4 \vee y_{u,v} \vee z_{u,v} & \overline{z}_{u,v} \vee B_2 \; ,
\end{array}$$

where A_1, A_2, A_3, A_4 and B_1, B_2 are the result of applying the substitution to A and B respectively. These clauses may contain repetitions: if A does not contain negative literals then $A_1, \ldots, A_4$ are all the same. If A contains one negative literal then we get two clauses repeated twice each. If A contains two negative literals then they are all different. Similarly for B: if it contains no negative literals then B_1 is equal to B_2, otherwise it contains one negative literal and B_1 is different from B_2. B cannot contain two negative literals.

By resolving on both variables $y_{u,v}$ and $z_{u,v}$ we obtain clauses $A_\mu \vee B_\nu$ for $\mu \in \{1,2,3,4\}$ and $\nu \in \{1,2\}$. We can exclude the possibility that A contains two negative literals and simultaneously B contains one, because otherwise $A \vee B$ would have three negative literals. Therefore, the set of newly derived clauses has size at most four and is indeed the sequence of clauses obtained by applying the substitution to $A \vee B$. This concludes the induction and gives us a refutation of $\mathsf{GOP}^{\mathsf{OR}}(G)$ since C_τ is the empty clause and, therefore, $\mathcal{C}_\tau$ is the set containing only the empty clause. ◀

▶ **Lemma 10.** *There exists a sequence of graphs $\{G_n\}_n$ such that each G_n has $\Theta(n)$ vertices and constant degree d, and any PC refutation of $\mathsf{GOP}^{\mathsf{OR}}(G_n)$ without negative literals requires monomial size $2^{\Omega(n)}$.*

Proof. Let $\{G_n\}_n$ be the sequence of graphs given by Proposition 8. Let $\mathcal{P}$ be a refutation of $\mathsf{GOP}^{\mathsf{OR}}(G_n)$ in monomial size s. By Lemma 6, there is a restriction ρ that sets exactly one of $\{y_{u,v}, z_{u,v}\}$ to 0 and is such that all monomials in $\mathcal{P}\restriction_\rho$ have degree at most $\log s$. Note that the formula $\mathsf{GOP}^{\mathsf{OR}}(G_n)\restriction_\rho$ is an isomorphic copy $\mathsf{GOP}(G_n)$, where each variable $x_{u,v}$ has been renamed either to $y_{u,v}$ or to $z_{u,v}$, and thus, by Proposition 8, it requires refutations of degree $\Omega(n)$. Since $\mathcal{P}\restriction_\rho$ is a PC refutation of $\mathsf{GOP}^{\mathsf{OR}}(G_n)\restriction_\rho$, we conclude that $\log s \geq \Omega(n)$ and the lemma follows. ◀

We collect the two lemmas in the following theorem.

▶ **Theorem 11.** *There is a family of constant width CNF formulas $\{F_n\}_n$ of size $\Theta(n^3)$ such that F_n has a resolution refutation of length $\Theta(n^3)$, but any* PC *refutation of F_n with no negative literals must contain $2^{\Omega(n)}$ monomials.*

▶ Remark 12. It is legitimate to ask whether the result holds when we reverse the encoding of true and false and adopt the classic standard for PC literature, where 0 is true and 1 is false. In this case, $\mathsf{GOP}^{\mathsf{OR}}(G_n)$ becomes easy for PC, but nevertheless we can get the same separation by simply flipping the polarity of all literals in $\mathsf{GOP}^{\mathsf{OR}}(G_n)$, i.e., by substituting each $x_{u,v}$ with $\overline{y}_{u,v} \vee \overline{z}_{u,v}$ instead of $y_{u,v} \vee z_{u,v}$, and then changing the random restriction to assign to true the variable chosen from each pair. Since in this case true is 0, monomials of large degree will be set to zero with overwhelmingly high probability.

We end this section by presenting a family of formulas that have small size, small space refutations in resolution – and, therefore, also in PC with negative literals – but exhibit a strong size-space trade-off for PC without negative literals. To define the *space* of a refutation, we think of it as a proof being presented on a blackboard. At each step we can either write down an axiom of the formula being refuted or a new clause obtained by one of the derivation rules of the proof system applied to what is already on the blackboard, or we can erase a line from the blackboard. The resolution space of the refutation is then the maximum number of clauses on the blackboard at any given moment, and the PC space of the refutation is the maximum number of monomials on the blackboard at any given moment.

▶ **Theorem 13.** *There exists a family of constant-width CNF formulas $\{F_n\}_{n\in\mathbb{N}}$ of size $\Theta(n)$ such that:*

1. *there is a resolution refutation of F_n in size $\mathrm{O}(n)$ and space $\mathrm{O}(1)$; but*
2. *any* PC *refutation without negative literals of F_n in monomial size t and space s must satisfy $s \log t = \Omega(n/\log n)$.*

The CNF formulas we consider are lifted pebbling formulas as defined next. Let $G = (V, E)$ be a DAG. If $(u, v) \in E$ we say that u is a *predecessor* of v and v a *successor* of u. We write $\mathrm{pred}(v)$ to denote the set of all predecessors of v. A vertex with no predecessor (resp. successor) is called a source (resp. sink).

The *pebbling formula* [9] over a DAG $G = (V, E)$ with a single sink z, denoted Peb_G, consists of the clauses $x_v \vee \bigvee_{u\in\mathrm{pred}(v)} \neg x_u$ for all $v \in V$ (note that if v is a source, then $\mathrm{pred}(v) = \emptyset$) encoding that sources are true and truth propagates upwards, and the clause $\neg x_z$ encoding that the sink is false. We encode this formula by a set of polynomials in the standard way. Given a set $U \subseteq V$, we denote by x_U the monomial $\prod_{u\in U} x_u$ (in particular, $x_\emptyset = 1$). For every vertex $v \in V$, we have the polynomial equation

$$x_{\mathrm{pred}(v)} \cdot (1 - x_v) = 0 \ , \tag{15}$$

and for the sink z we also have the polynomial equation

$$x_z = 0 \ . \tag{16}$$

The formulas that witness the trade-off separation of Theorem 13 are based on the family of graphs defined by Gilbert and Tarjan [24]. These graphs have large pebbling cost $\Omega(n/\log n)$, even in the stronger, so-called *black-white* pebbling model and were used in [8] to obtain a space-degree trade-off for PC.

▶ **Lemma 14** ([24, 8]). *There is a family of graphs $\{G_n\}_{n\in\mathbb{N}}$ with indegree 2 of size $\Theta(n)$ such that any* PC *refutation, even with negative literals, of Peb_{G_n} in space s and degree d must satisfy $sd = \Omega(n/\log n)$.*

With this result, we are now ready to prove Theorem 13.

Proof of Theorem 13. Let $\{G_n\}_{n\in\mathbb{N}}$ be the family of graphs given by Lemma 14 and let N be the number of vertices of G_n. Let $x_1, \ldots, x_N$ be the variables of Peb_{G_n} in inverse topological order. We define $F_n = \mathrm{Peb}^{\mathsf{NOR}}_{G_n}$, that is, we substitute each variable x_i by $\neg(y_i \vee z_i)$ and rewrite the formula in CNF.

The linear size resolution refutation of $\mathrm{Peb}^{\mathsf{NOR}}_{G_n}$ in space $\mathrm{O}(1)$ can be described in rounds. We start with the clause $y_1 \vee z_1$. At the end of round i, we will have derived a clause $\bigvee_{j\in S_i}(y_j \vee z_j)$ for some set $S_i \subseteq [i]$ such that S_i forms a cut in G_n, that is, the sink of G_n and the sources of G_n are not connected in $G_n \setminus S_i$; and moreover every vertex in S_i has at least one predecessor not in S_i. Furthermore, at each round, the cut S_i moves towards the sources, i.e., the set of vertices connected to the sink in $G_n \setminus S_i$ increases when i increases.

For round $i+1$, we first weaken $\bigvee_{j\in S_i}(y_j \vee z_j)$ to $\bigvee_{j\in S_i\cup\{i+1\}}(y_j \vee z_j)$. Now, for all $v \in S_i$ such that both predecessors of v, say u and w, are in $S_i \cup \{i+1\}$ we resolve $\bigvee_{j\in S_i\cup\{i+1\}}(y_j \vee z_j)$ with $y_u \vee z_u \vee y_w \vee z_w \vee \bar{y}_v$ and then with $y_u \vee z_u \vee y_w \vee z_w \vee \bar{z}_v$, thus obtaining a clause $\bigvee_{j\in S_{i+1}}(y_j \vee z_j)$ for some set S_{i+1} that satisfies the invariant. Finally, after round N, we have derived $\bigvee_{j\in S_N}(y_j \vee z_j)$ where S_N only contains sources. Thus, we can easily derive contraction by resolving this with $\bar{y}_j$ and $\bar{z}_j$ for all $j \in S_N$. Note that this refutation has space 3 and size $O(N) = O(n)$.

Now for proving item 2, let $\mathcal{P}$ be a PC refutation without negative literals of $\mathrm{Peb}^{\mathsf{NOR}}_{G_n}$ in monomial size t and space s. By Lemma 6, there is a restriction ρ that for all $i \in [N]$ sets exactly one of $\{y_i, z_i\}$ to 0 and such that all monomials in $\mathcal{P}$ when restricted by ρ have degree at most $\log t$. Since space does note increase with restriction, we have that $\mathcal{P}\restriction_\rho$ is a refutation of $\mathrm{Peb}^{\mathsf{NOR}}_{G_n}\restriction_\rho$ in space at most $\mathrm{O}(s)$ and degree at most $\log t$.

We now argue that there is a PC refutation with negative literals of Peb_{G_n} in space $\mathrm{O}(s)$ and degree $\mathrm{O}(\log t)$ and, by Lemma 14, this will imply that $s \log t = \Omega(n/\log n)$. Let H be the formula $\mathrm{Peb}^{\mathsf{NOR}}_{G_n}\restriction_\rho$ with any y_i substituted by $(1-\bar{y}_i)$ and any z_i by $(1-\bar{z}_i)$. Since H is an isomorphic copy of Peb_{G_n}, where each variable x_i has been substituted by either $\bar{y}_i$ or $\bar{z}_i$, it is enough to show that there is a PC refutation with negative literals of H in space $\mathrm{O}(s)$ and degree $\mathrm{O}(\log t)$. Indeed, this follows since we can derive each axiom of $\mathrm{Peb}^{\mathsf{NOR}}_{G_n}\restriction_\rho$ from an axiom of H and variable axioms in constant space and degree. ◀

4 Negative Literals and Semialgebraic Proofs

We show that allowing negative literals makes Sherali-Adams and sums-of-squares exponentially stronger, too. The main result of this section is that there is a family of formulas that have short resolution refutations but require exponential size SA and SOS refutations without negative literals. This implies, in both systems, an exponential separation between the power of proofs with and without negative literals.

The following auxiliary lemma states the well-known semantic completeness of SA.

▶ **Lemma 15** (Folklore). *If some multilinear inequalities $\mathcal{S} = \{r_1 \geq 0, \ldots, r_\ell \geq 0\}$ on variables $\vec{x} = (x_1, \ldots, x_n)$ semantically imply a multilinear inequality $r \geq 0$ then there is an* SA *derivation of r from $\mathcal{S}$ in degree $2n$ and size $2^{\mathrm{O}(n)}$.*

Proof. For a multilinear polynomial p, we define the sets $S_p^- := \{\alpha \in \{0,1\}^n \mid p(\alpha) < 0\}$ and $S_p^+ := \{\alpha \in \{0,1\}^n \mid p(\alpha) \geq 0\}$. The fact that inequality $r \geq 0$ is semantically implied by $\mathcal{S}$ means that $S_r^- \subseteq \bigcup_i S_{r_i}^-$.

Let $Q_i \coloneqq S^-_{r_i} \setminus \bigcup_{j=1}^{i-1} S^-_{r_j}$. Consider the polynomial

$$\sum_{i\in[\ell]} \left(\sum_{\alpha\in Q_i \cap S^-_r} \frac{|r(\alpha)|}{|r_i(\alpha)|} r_i(\vec{x})\chi_\alpha(\vec{x}) \right) + \sum_{\alpha\in S^+_r} r(\alpha)\chi_\alpha(\vec{x}) \ , \tag{17}$$

where $\chi_\alpha(\vec{x})$ is the characteristic function of a point α. The polynomial (17) is pointwise equivalent to r on the boolean cube because of the definition of the characteristic functions. Moreover, (17) is a legal SA derivation from $\mathcal{S}$ because $S^-_r \subseteq \bigcup S^-_{r_i}$ implies that coefficients $\frac{|r(a)|}{|r_i(a)|}$ in (17) are all positives.

The degree of the polynomial (17) is at most $2n$ by definition and size is at most 2^{3n}. Since it is pointwise equivalent to r on the boolean it is enough to multilinearize to transform it into r.

For multilinearization we apply the following procedure. Denote by $h(\vec{x})$ the polynomial (17) after expanding brackets. While polynomial $h(\vec{x})$ has a term of the form $x_i^d t$ we subtract a polynomial $tx_i^{d-2}(x_i^2 - x_i)$ from polynomial (17) where $i \in [n]$ and $d \geq 2$ is an integer. In one step we reduce the individual degree of one variable in one term in the polynomial $h(x)$ and increase the size of polynomial (17) by 2. At the end of the process (17) is a multilinear polynomial of degree at most $2n$ and size at most $2n2^{3n}$, pointwise equal to r. After expanding brackets it will be a multilinear polynomial that is pointwise equivalent to r on the boolean cube. ◀

▶ **Lemma 16.** *Consider two sets $\mathcal{S}_1 \coloneqq \{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ and $\mathcal{S}_2 \coloneqq \{f_1 = 0, \ldots, f_{m'} = 0; g_1 \geq 0, \ldots, g_{\ell'} \geq 0\}$. If there is an* SA *(resp.* SOS*) refutation of $\mathcal{S}_2$ in size N_2 and degree d_2 and each element $f_i \geq 0$, $-f_i \geq 0$, and $g_i \geq 0$ can be derived in* SA *(resp.* SOS*) from $\mathcal{S}_1$ in size N_1 and degree d_1, then there is an* SA *(resp.* SOS*) refutation of $\mathcal{S}_1$ in size N_1N_2 and degree d_1d_2 (resp. in size $N_1N_2^{O(1)}$ and degree $O(d_1d_2)$).*

Proof. First consider a set $\mathcal{S}_2$ without equations (i.e., $m' = 0$). Let $\{h_1, \ldots, h_d\}$ be an SA (or SOS) refutation of $\mathcal{S}_2$ in size N_2 and degree d_2, so that we have

$$\sum_{i\in[\ell']} h_i g_i = -1 \ . \tag{18}$$

For $i \in [\ell']$, let $\{q_{1,i}, \ldots, q_{m,i}; s_{1,i}, \ldots, s_{\ell,i}\}$ be an SA (or SOS) derivation of $g_i \geq 0$ from $\mathcal{S}_1$ in size N_1 and degree d_1, so that

$$\sum_{j\in[m]} q_{j,i}p_j + \sum_{j\in[\ell]} s_{j,i}r_j = g_i \ . \tag{19}$$

The composition of these derivations

$$-1 = \sum_{i\in[\ell']} h_i g_i = \sum_{i\in[\ell']} h_i \left(\sum_{j\in[m]} q_{j,i}p_j + \sum_{j\in[\ell]} s_{j,i}r_j \right) \tag{20}$$

$$= \sum_{j\in[m]} \left(\sum_{i\in[\ell']} h_i q_{j,i} \right) p_j + \sum_{j\in[\ell]} \left(\sum_{i\in[\ell']} h_i s_{j,i} \right) r_j \tag{21}$$

gives us the desired refutation of $\mathcal{S}_1$ in size N_1N_2 and degree d_1d_2. Notice that (21) is a valid SA (or SOS) refutation because polynomials h_i and $s_{j,i}$ are valid multipliers for inequalities and thus so are their products and sums of products.

When $\mathcal{S}_2$ contains equations, we reduce to the case where $m' = 0$, using the observation that the set

$$\mathcal{S}_2' := \{-f_1 \geq 0, \ldots, -f_{m'} \geq 0; f_1 \geq 0, \ldots, f_{m'} \geq 0; g_1 \geq 0, \ldots, g_{\ell'} \geq 0\} \tag{22}$$

has an SA refutation of size N_2 and degree d_2 (or an SOS refutation of size $N_2^{O(1)}$ and degree $O(d_2)$). To see this, start from a refutation $\{e_1, \ldots, e_m; h_1, \ldots, h_\ell\}$ of $\mathcal{S}_2$ in size N_2 and degree d_2, so that we have

$$\sum_{i \in [m']} e_i f_i + \sum_{i \in [\ell']} h_i g_i = -1 \ . \tag{23}$$

To make it a valid SA refutation of $\mathcal{S}_2'$, rewrite each $e_i f_i$ as $e_i^+(f_i) + e_i^-(-f_i)$ where $e_i = e_i^+ - e_i^-$ and both e^+ and e^- are positive sums of monomials. Note that this operation does not change neither size nor degree. To make it a valid SOS refutation of $\mathcal{S}_2'$, rewrite each $e_i f_i$ as $\left(\frac{e_i+1}{2}\right)^2 \cdot f_i + \left(\frac{e_i-1}{2}\right)^2 \cdot (-f_i)$. Note that this refutation has degree at most $2d_2$ and size at most N_2^2. The result follows. ◀

Recall the ordering principle formula OP_n, which is the graph ordering principle formula (12) over the complete graph K_n. As mentioned in Section 2, for SA and SOS the default encoding of CNF formulas is (6d). For OP_n this enconding consists of inequalities:

$$(1 - x_{u,v}) + (1 - x_{v,w}) + x_{u,w} - 1 \geq 0 \qquad \text{for any three distinct } u, v, w \in [n], \tag{24a}$$

$$(1 - x_{u,v}) + (1 - x_{v,u}) - 1 \geq 0 \qquad \text{for any two distinct } u, v \in [n], \tag{24b}$$

$$\sum_{u \in [n]} x_{u,v} - 1 \geq 0 \qquad \text{for any } u, v \in [n]. \tag{24c}$$

The reason we cannot use the graph ordering principle as we did in Section 3 is that we do not know how to prove strong SA degree lower bounds for GOP. Instead we use OP_n which can be still encoded in low degree using inequalities, and for which we have degree lower bounds.

For the separation we use the $\mathsf{OP}_n^{\mathsf{OR}}$ formula. We have already showed in Lemma 9 that $\mathsf{OP}_n^{\mathsf{OR}}$ is easy for resolution. In the presence of negative literals, this transfers to SA by the following known simulation result.

▶ **Lemma 17** ([5]). *If a CNF formula F has a resolution refutation of width w and length L, then it has an* SA *refutation with negative literals of degree $w + 1$ and size* $\mathrm{O}(w^2 L)$.

Since SOS can simulate SA we obtain the following upper bound.

▶ **Lemma 18.** *The formula $\mathsf{OP}_n^{\mathsf{OR}}$ has* SA *and* SOS *refutations with negative literals of size $n^{O(1)}$.*

Proof. By Lemma 9 the formula $\mathsf{OP}_n^{\mathsf{OR}}$ has a resolution refutation of size $O(n^3)$. The width of any resolution refutation cannot exceed the number of variables that appear in the formula, hence the considered refutation has width at most $\mathrm{O}(n^2)$. Together with Lemma 17, this implies the desired result for SA. To conclude the proof it is enough to recall that, by Proposition 5, SOS can simulate any SA proof with at most a polynomial blowup in size. ◀

We now proceed to prove the lower bounds for SA and SOS without negative literals. The main idea is analogous to that of Lemma 10: we show that we can reduce any small SA or SOS refutation without negative literals of $\mathsf{OP}_n^{\mathsf{OR}}$ to a low degree refutation of OP_n. To conclude the proof we then apply the following degree lower bounds.

▶ **Lemma 19** ([16]). *Any* SA *refutation of* OP_n *has degree at least* $n-2$.

For SOS the lower bound we know holds for the following, slightly different encoding:

$$x_{u,v}x_{v,w}(1-x_{u,w})=0 \qquad \text{for any three distinct } u,v,w\in[n], \tag{25a}$$

$$x_{u,v}x_{v,u}=0 \qquad \text{for any two distinct } u,v\in[n], \tag{25b}$$

$$\sum_{u\in[n]} x_{u,v}=1+z_v^2 \qquad \text{for any } u,v\in[n], \tag{25c}$$

where z_v are real valued extension variables.

▶ **Lemma 20** ([35]). *For any* $\varepsilon>0$*, there is a constant* $c_\varepsilon>0$ *such that any* SOS *proof of the system of equations (25) has degree at least* $c_\varepsilon n^{1/2-\varepsilon}$.

We show that this result implies a degree lower bound for the standard encoding of OP_n as in (24).

▶ **Corollary 21.** *For any* $\varepsilon>0$*, there is a constant* $c_\varepsilon>0$ *such that any* SOS *proof of the* OP_n *has degree at least* $c_\varepsilon n^{1/2-\varepsilon}$.

Proof. For the sake of completeness, let us argue a well known fact. If $p=0$ is the product encoding, as per (6b), of a clause C of width w, and $r\geq 0$ is the additive encoding of C, as per (6d), then there is an SA (and hence also SOS) derivation of r from p and boolean axioms in degree $w+1$. Indeed, this follows from Lemma 15 by noting that the product encoding $p=0$ is equivalent to the two inequalities $p\geq 0$ and $-p\geq 0$ that semantically imply the inequality $r\geq 0$.

By using the above fact we can derive inequalities (24a) and (24b) from the constraints (25a) and (25b) in degree 4. Finally, the inequality

$$\sum_{u\in[n]} x_{u,v}-1\geq 0 \tag{26}$$

can be derived in SOS from (25c) by adding the square z_v^2 and thus obtaining

$$\left(\sum_{u\in[n]} x_{u,v}-1-z_v^2\right)+z_v^2=\sum_{u\in[n]} x_{u,v}-1 \ . \tag{27}$$

Therefore, if there an SOS refutation of (24) in degree d, then by Lemma 16 there is an SOS refutation of (25) in degree $O(d)$. Together with Lemma 20, this implies the desired lower bound. ◀

We are now ready to prove the size lower bounds for SA and SOS.

▶ **Lemma 22.** *Any* SA *refutation of* $\mathsf{OP}_n^{\mathsf{OR}}$ *without negative literals requires monomial size* $2^{\Omega(n)}$*. For any* $\varepsilon>0$ *there is a constant* $c_\varepsilon>0$ *such that any* SOS *refutation of* $\mathsf{OP}_n^{\mathsf{OR}}$ *without negative literals requires monomial size* $2^{c_\varepsilon n^{1/2-\varepsilon}}$.

Proof. The proof is very similar to that of Lemma 10. Let $y_{u,v}$, $z_{u,v}$ for $u,v\in[n]$ be the variables of $\mathsf{OP}_n^{\mathsf{OR}}$, that is, $\mathsf{OP}_n^{\mathsf{OR}}$ is obtained by substituting in OP_n each variable $x_{u,v}$ by $y_{u,v}+z_{u,v}$. Let $\{p_1=0,\ldots,p_m=0;r_1\geq 0,\ldots,r_\ell\geq 0\}$ is the encoding of $\mathsf{OP}_n^{\mathsf{OR}}$ and let $\{q_1,\ldots,q_m;s_1,\ldots,s_\ell\}$ be an SA refutation of $\mathsf{OP}_n^{\mathsf{OR}}$ without negative literals, so that

$$\sum_{j\in[m]} q_jp_j+\sum_{j\in[\ell]} s_jr_j=-1 \ . \tag{28}$$

Let S be the monomial size of this refutation. By Lemma 6, there is a restriction ρ that sets exactly one of $\{y_{u,v}, z_{u,v}\}$ to 0 and is such that all monomials appearing in (28) when restricted by ρ have degree at most $\log S$. Note that the formula $\mathsf{OP}_n^{\mathsf{OR}}\restriction_\rho$ is an isomorphic copy OP_n, where each variable $x_{u,v}$ has been renamed either to $y_{u,v}$ or to $z_{u,v}$, and thus, by Lemma 19, it requires refutation of degree $\Omega(n)$. Since $\mathcal{P}\restriction_\rho$ is an SA refutation of $\mathsf{OP}_n^{\mathsf{OR}}\restriction_\rho$ in degree at most $\log S$, we conclude that $\log S \geq \Omega(n)$ and the size lower bound for SA follows.

The proof of the size lower bound for SOS is analogous, except that we use Corollary 21 for the degree lower bound instead of Lemma 19. ◀

We collect Lemmas 9,18 and 22 in the following theorem.

▶ **Theorem 23.** *There is a family of CNF formulas $\{F_n\}_n$ of size $\Theta(n^3)$ such that F_n has a resolution refutation and* SA *and* SOS *refutations with negative literals in monomial size $n^{\mathrm{O}(1)}$. But any* SA *refutation of F_n without negative literals requires monomial size $2^{\Omega(n)}$, and for any $\varepsilon > 0$ there is a constant $c_\varepsilon > 0$ such that any* SOS *refutation of F_n without negative literals requires monomial size $2^{c_\varepsilon n^{1/2-\varepsilon}}$.*

5 Pigeonhole and Sherali-Adams

In this section we improve the previous result for Sherali-Adams and show a separation between SA with and without negative literals, using constant width formulas, and hence independent of the encoding of the clauses. Note that, in contrast to the previous section, this result does not give a corresponding separation for SOS.

We start with the formula that encodes the (negation of the) pigeonhole principle (PHP). The formula is defined on propositional variables $x_{i,j}$ for $i \in [n+1]$ and $j \in [n]$, with the intended meaning that $x_{i,j}$ is true if and only if the i-th pigeon goes into hole j. The clauses of PHP are:

$$\mathsf{P}_i := \bigvee_{j\in[n]} x_{i,j} \qquad \text{for every } i \in [n+1], \text{ and} \tag{29a}$$

$$\mathsf{H}_{i,k}^{j} := \overline{x}_{i,j} \vee \overline{x}_{k,j} \qquad \text{for every two distinct } i,k \in [n+1] \text{ and every } j \in [n]. \tag{29b}$$

In order to reduce the width of the formula we introduce extension variables $e_{i,j}$ for $i \in [n+1]$ and $j \in [n]$ and replace the clauses (29a) by

$$\mathsf{EP}_{i,j} := e_{i,j-1} \vee x_{i,j} \vee \overline{e}_{i,j} \qquad \text{for every } i \in [n+1] \text{ and } j \in [n], \tag{30a}$$

$$\mathsf{EP}_{i,0} := \overline{e}_{i,0}, \qquad \mathsf{EP}_{i,n+1} := e_{i,n} \qquad \text{for every } i \in [n+1]. \tag{30b}$$

Intuitively, the variable $e_{i,j}$ represents the disjunction of the variables $x_{i,\ell}$ for $\ell \leq j$. We denote this 3-CNF formula with extension variables by EPHP.

Similarly to previous cases, we substitute the variables in the formula by a 2-bit function. In this case, however, we use $\mathsf{NOR}(y,z) := \neg(y \vee z)$ which is equivalent to $\overline{y} \wedge \overline{z}$. We apply this substitution to the formula EPHP, to obtain the formula $\mathsf{EPHP}^{\mathsf{NOR}}$, by replacing each variable $x_{i,j}$ with $\overline{y}_{i,j} \wedge \overline{z}_{i,j}$ and each $e_{i,j}$ with $\overline{a}_{i,j} \wedge \overline{b}_{i,j}$ and rewriting it in CNF.

It was shown in [16] that Sherali-Adams without negative literals can refute PHP in polynomial size. We use this result to obtain a size upper bound for Sherali-Adams refutations with negative literals of $\mathsf{EPHP}^{\mathsf{NOR}}$.

▶ **Lemma 24** ([16]). *There is an* SA *refutation without negative literals of* PHP *of size $\mathrm{O}(n^4)$.*

▶ **Lemma 25.** *There is an* SA *refutation with negative literals of* $\mathsf{EPHP}^{\mathsf{NOR}}$ *of size $\mathrm{O}(n^5)$.*

Proof. Let $\mathcal{S}$ be the set of polynomial inequalities encoding PHP as per (6d) plus the variable axioms for each $x_{i,j}$ and let $\mathcal{S}' = \{p_1 = 0, \dots, p_m = 0; r_1 \geq 0, \dots, r_\ell \geq 0\}$ be the set of polynomial inequalities obtained from $\mathcal{S}$ by replacing each variable $x_{i,j}$ by the product $\overline{y}_{i,j}\overline{z}_{i,j}$.

We want show that there is a small, namely size $\mathrm{O}(n)$, SA derivation with negative literals from $\mathsf{EPHP}^{\mathsf{NOR}}$ of each of the inequalities $p_i \geq 0$, $-p_i \geq 0$ for $i \in [m]$ and $r_i \geq 0$ for $i \in [\ell]$. Suppose this is true. Then given a size $\mathrm{O}(n^4)$ refutation of PHP, which is guaranteed to exist by Lemma 24, we can replace each occurrence of $x_{i,j}$ by the product $\overline{y}_{i,j}\overline{z}_{i,j}$ and obtain a refutation of $\mathcal{S}'$ of exactly the same size. Composing this refutation with the derivation of $\mathcal{S}'$ from $\mathsf{EPHP}^{\mathsf{NOR}}$, we obtain, by Lemma 16, a refutation of $\mathsf{EPHP}^{\mathsf{NOR}}$ of size $\mathrm{O}(n^5)$.

We start by considering the hole axioms (29b) of PHP, that is, $\overline{x}_{i,j} \vee \overline{x}_{k,j}$, which is encoded as $(1 - x_{i,j}) + (1 - x_{k,j}) - 1 \geq 0$. After replacing the x variables in the polynomial inequality, we obtain

$$(1 - \overline{y}_{i,j}\overline{z}_{i,j}) + (1 - \overline{y}_{k,j}\overline{z}_{k,j}) - 1 \geq 0 \ , \tag{31}$$

which is in $\mathcal{S}'$. Now, the formula EPHP also contains hole axioms (29b), and thus the substituted formula $\mathsf{EPHP}^{\mathsf{NOR}}$ contains a set of inequalities encoding the formula $(\overline{y}_{i,j} \wedge \overline{z}_{i,j}) \vee (\overline{y}_{k,j} \wedge \overline{z}_{k,j})$. Since this formula, and therefore also the inequalities encoding it, semantically implies inequality (31), by Lemma 15 there is an SA derivation of (31) from $\mathsf{EPHP}^{\mathsf{NOR}}$ in constant size.

We have a similar situation for the pigeon axioms (29a) of PHP, i.e., $\bigvee_{j\in[n]} x_{i,j}$, which is encoded as $\sum_{j=1}^{n} x_{i,j} - 1 \geq 0$. Our goal is to derive the polynomial inequality

$$\sum_{j=1}^{n} \overline{y}_{i,j}\overline{z}_{i,j} - 1 \geq 0 \tag{32}$$

from $\mathsf{EPHP}^{\mathsf{NOR}}$. Again, by Lemma 15, each of the inequalities

- $(1 - \overline{a}_{i,0}\overline{b}_{i,0}) - 1 \geq 0$;
- $\overline{a}_{i,j-1}\overline{b}_{i,j-1} + \overline{y}_{i,j}\overline{z}_{i,j} + (1 - \overline{a}_{i,j}\overline{b}_{i,j}) - 1 \geq 0$, for all $j \in [n]$; and
- $\overline{a}_{i,n}\overline{b}_{i,n} - 1 \geq 0$

has an SA derivation from $\mathsf{EPHP}^{\mathsf{NOR}}$ of the constant size, since they are semantically implied by the clauses $\mathsf{EP}_{i,j}$ with variables substitute by NOR. Note that the sum of these inequalities

$$(1-\overline{a}_{i,0}\overline{b}_{i,0})-1+\sum_{j-1}^{n}(\overline{a}_{i,j-1}\overline{b}_{i,j-1}+\overline{y}_{i,j}\overline{z}_{i,j}+(1-\overline{a}_{i,j}\overline{b}_{i,j})-1)+\overline{a}_{i,n}\overline{b}_{i,n}-1 = \sum_{j=1}^{n}\overline{y}_{i,j}\overline{z}_{i,j}-1 \tag{33}$$

is a valid SA derivation of (32) in size $O(n)$.

Finally, we note that the substituted variable axioms $\overline{y}_{i,j}\overline{z}_{i,j} \geq 0$, $1 - \overline{y}_{i,j}\overline{z}_{i,j} \geq 0$, $(\overline{y}_{i,j}\overline{z}_{i,j})^2 - \overline{y}_{i,j}\overline{z}_{i,j} \geq 0$ and $-(\overline{y}_{i,j}\overline{z}_{i,j})^2 + \overline{y}_{i,j}\overline{z}_{i,j} \geq 0$ can be easily derived in constant size from the variable axioms for $y_{i,j}$ and $z_{i,j}$. ◀

We now show that any Sherali-Adams refutation of $\mathsf{EPHP}^{\mathsf{NOR}}$ without negative literals has exponential size. For this, we use the following degree lower bound.

▶ **Lemma 26** ([5]). *Any* SA *refutation of* EPHP *has a degree at least* $n - 2$.

▶ **Lemma 27.** *Any* SA *refutation of* $\mathsf{EPHP}^{\mathsf{NOR}}$ *without negative literals requires monomial size* $2^{\Omega(n)}$.

Proof. The proof is very similar to that of Lemma 22. Consider an SA refutation of $\mathsf{EPHP}^{\mathsf{NOR}}$ without negative literals, that is, a set of polynomials $\mathcal{P} = \{q_1, \ldots, q_m; s_1, \ldots, s_\ell\}$ such that

$$\sum_{j\in[m]} q_j p_j + \sum_{j\in\ell} s_j r_j = -1 \ , \tag{34}$$

where $\{p_1 = 0, \ldots, p_m = 0; r_1 \geq 0, \ldots, r_\ell \geq 0\}$ is the polynomial encoding of $\mathsf{EPHP}^{\mathsf{NOR}}$ and each s_j is a positive linear combination of generalized monomials. Let S be the monomial size of this refutation. By Lemma 6, there is a restriction ρ that sets exactly one of $\{y_{i,j}, z_{i,j}\}$ and exactly one of $\{a_{i,j}, b_{i,j}\}$ to 0 and is such that all monomials appearing in (34) when restricted by ρ have degree at most $\log S$. Note that the formula $\mathsf{EPHP}^{\mathsf{NOR}}\restriction_\rho$ is almost an isomorphic copy of EPHP, except that:

- each variable $x_{i,j}$ has been substituted by either $(1 - y_{i,j})$ or by $(1 - z_{i,j})$;
- each variable $e_{i,j}$ has been substituted by either $(1 - a_{i,j})$ or by $(1 - b_{i,j})$.

It is not hard to see that this formula $\mathsf{EPHP}^{\mathsf{NOR}}\restriction_\rho$ also requires degree $n - 2$ to be refuted in SA, since otherwise we could obtain, by substituting each variable $y_{i,j}$ and $z_{i,j}$ by $(1 - x_{i,j})$ and each variable $a_{i,j}$ and $b_{i,j}$ by $(1 - e_{i,j})$, a refutation of EPHP in degree less than $n - 2$ contradicting Lemma 19. Therefore, since $\mathcal{P}\restriction_\rho$ is an SA refutation of $\mathsf{EPHP}^{\mathsf{NOR}}\restriction_\rho$ in degree at most $\log S$, we conclude that $\log S \geq \Omega(n)$ and the size lower bound for SA follows. ◀

We collect Lemmas 25 and 27 in the following theorem.

▶ **Theorem 28.** *There is a family of constant width CNF formulas $\{F_n\}_n$ of size $\Theta(n^3)$ such that F_n has an SA refutation with negative literals of monomial size $\mathrm{O}(n^5)$, but any SA refutation of F_n without negative literals must contain $2^{\Omega(n)}$ monomials.*

6 Separating Nullstellensatz with and without negative literals

In this section we show that there are formulas that have linear size tree-like resolution refutations – and, therefore, also linear size Nullstellensatz refutations if variables for negative literals are allowed – but require nearly exponential size Nullstellensatz refutations if such variables are not allowed.

▶ **Theorem 29.** *There exists a family of constant width CNF formulas $\{F_n\}_{n\in\mathbb{N}}$ of size $\Theta(n)$ such that there are tree-like resolution refutations, and therefore also NS refutations with negative literals, of F_n in size $\mathrm{O}(n)$, but any NS refutation without negative literals of F_n must have size $2^{\Omega(n/\log n)}$.*

A formula that witnesses a size separation of $2^{\widetilde{\Omega}(n)}$ must necessarily require NS degree $\widetilde{\Omega}(n)$ since if there is a degree-d NS refutation, then there is an NS refutation without negative literals in simultaneous degree d and size $n^{O(d)}$. In this sense, the separation in Theorem 29 is nearly optimal. For smaller values of d, we can show a similar separation with the additional property that NS with negative literals presents a smooth trade-off between degree and size of refutations.

▶ **Theorem 30.** *For any $0 < \epsilon \leq 1/4$, any large enough $n \in \mathbb{N}$ and any $2 \leq k \leq n^{\epsilon/2}$, there exists a constant width CNF formula $F_{k,n}$ of size $\Theta(kn)$ such that:*

1. *there is an NS refutation with negative literals of $F_{k,n}$ in linear size $\mathrm{O}(kn)$;*
2. *for any d satisfying $2^{1+1/\epsilon}k^4 \log n \leq d \leq \sqrt{n}$, there is an NS refutation with negative literals of $F_{k,n}$ in degree d and size $n^{k(1+5\epsilon)}/d^{2k-3}$; and*
3. *any NS refutation without negative literals of $F_{k,n}$ must have size 2^k.*

The CNF formulas we consider are lifted pebbling formulas. We will also use the relation between the formulas and pebble games as defined next.

The *reversible pebble game* [10] is a single-player game that is played with a set of pebbles on a DAG G. The goal of the game is to pebble (i.e., place a pebble on) each vertex of G at least once. Initially, the graph contains no pebbles. At each round, the player is allowed to place a pebble on any vertex of G such that all its predecessors are pebbled. In particular, the player is always allowed to place a pebble on any source of G. Moreover, at any given round, a pebble on a vertex v can be removed from G if all the predecessors of v are pebbled. Again, this implies that it is always possible to remove a pebble from a source of G. A sequence of pebbling moves that pebbles each vertex of G at least once according to these rules and ends with the empty graph is called a *reversible pebbling* of G. The *time* of a reversible pebbling is the number of rounds and the *space* is the maximum number of pebbles on G at any given moment. The *reversible pebbling cost* of G is the minimum space required for any reversible pebbling of G (independent of time). We sometime refer to the *standard pebble game* where the rule for removing pebbles is relaxed so that any pebble can be removed at any point.

For our purpose, we note that pebbling formulas always have linear size NS refutations (even without negative literals), while for some "hard" graphs the NS degree is necessarily large. In order to prove the separations in this section, we use the following characterization of NS degree and size, when negative literals are not allowed, in terms of reversible pebbling space and time [19]. We would like to point out that for Theorem 29 the degree characterization of [18], or even the not-so-tight bound of [12], would have be enough.

▶ **Lemma 31** ([19]). *Let G be a single-sink DAG. There is a Nullstellensatz degree d and size t refutation without negative literals of Peb_G if and only if there is a reversible pebbling of G in space d and time $t-1$.*

By this characterisation, it is easy to see that pebbling formulas always have linear size NS refutations without negative literals. In order to obtain NS size lower bounds when negative literals are not allowed we compose pebbling formulas with the not-or function NOR, that is, we substitute each variable x_i by $\neg(y_i \vee z_i)$. This is useful for proving NS lower bounds since formulas lifted with NOR satisfy the following property.

▶ **Lemma 32.** *Let F be an unsatisfiable CNF formula. If NS requires degree d to refute F, then NS without negative literals requires size 2^d to refute F^{NOR}.*

Proof. Let n be the number of variables of F, and let $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$ be the variables of F^{NOR}. Let $\mathcal{S} = \{p_1 = 0, \ldots, p_m = 0\}$ be the set of polynomial equations encoding F^{NOR} (plus the variable axioms). Let $\{q_1, \ldots, q_m\}$ be an NS refutation without negative literals of $\mathcal{S}$, that is,

$$\sum_{j \in [m]} q_j p_j = 1 \ , \tag{35}$$

and let s be its monomial size. By Lemma 6, there is a restriction ρ that for all $i \in [n]$ sets exactly one of $\{y_i, z_i\}$ to 0 and such that all monomials in $q_j p_j$ for $p_j = 0 \in \mathcal{S}$ when restricted by ρ have degree less than $\log s$. Note that $F^{\mathsf{NOR}}\restriction_\rho$ is almost an isomorphic copy of F, except that variables x_i have been substituted by either $(1 - y_i)$ or $(1 - z_i)$. It is not hard to see that $F^{\mathsf{NOR}}\restriction_\rho$ also requires degree d refutations, since otherwise substituting every y_i or z_i appearing in the refutation by $(1 - x_i)$ would give a refutation of F in degree less than d. This implies that $\log s \geq d$. ◀

While substituting variables in a formula with NOR can give NS size lower bounds if negative literals are not allowed, for pebbling formulas this substitution does not make the formula harder for NS if negative literals are allowed, and not even for tree-like resolution.

▶ **Lemma 33.** *Let G be a DAG with n vertices. There is a tree-like resolution refutation of* $\mathrm{Peb}_G^{\mathsf{NOR}}$ *in size $4n+1$.*

Proof. We describe a decision tree that solves the falsified clause search problem of $\mathrm{Peb}_G^{\mathsf{NOR}}$. The idea is to query the variables in topological order, from the sources to the sink. Let $x_1, \ldots, x_n$ be the variables of Peb_G, ordered topologically according to G from the sources to the sink, and for $i \in [n]$, let y_i, z_i be the lifted variables so that $x_i = \neg(y_i \vee z_i)$. The decision tree queries y_i and z_i, from $i = 1$ to n: if the result of the query is 0 it proceeds to the next query, if it is 1 it has found a falsified axiom (since this implies there is a false variable whose predecessors are true). Finally, if all vertices are 0, then the sink clause of $\mathrm{Peb}_G^{\mathsf{NOR}}$, which states the sink is false, is falsified. This gives a decision tree of size $4n+1$ (and depth $2n$). ◀

We also observe that if a CNF formula has small NS refutations without negative literals in degree d, then the formula composed with NOR has small NS refutations with negative literals in degree $2d$.

▶ **Lemma 34.** *Let F be a constant-width unsatisfiable CNF formula. If there is an* NS *refutation without negative literals of F in size s and degree d then there is an* NS *refutation with negative literals of F^{NOR} in size $\mathrm{O}(s)$ and degree $2d$.*

Proof. Let $x_1, \ldots, x_n$ be the variables of F, and for $i \in [n]$, let y_i, z_i be the lifted variables so that $x_i = \neg(y_i \vee z_i)$. For a clause (or a CNF) C, let C^* be the polynomial translation of C without negative literals, as per (6b). Moreover, for a polynomial p over x variables, let $p[\bar{y}\bar{z}]$ be the polynomial obtained by substituting in p each variable x_i by the product $\bar{y}_i\bar{z}_i$.

Consider a clause C of F and denote by C^{NOR} the CNF that is obtained by substituting variables x_i by $\neg(y_i \vee z_i)$. Since C has constant width, there is an NS derivation (with negative literals) of $C^*[\bar{y}\bar{z}]$ from the set of polynomials $(C^{\mathsf{NOR}})^*$ in constant size and without increasing the degree.

Let $\mathcal{S} = \{p_1 = 0, \ldots, p_m = 0\}$ be the set of polynomial equations encoding F (plus the variable axioms). Let $\{q_1, \ldots, q_m\}$ be an NS refutation without negative literals of $\mathcal{S}$, that is,

$$\sum_{j \in [m]} q_j p_j = 1 \ , \tag{36}$$

in degree d and monomial size s. If we substitute every variable x_i in $\sum_{j \in [m]} q_j p_j$ by $\bar{y}_i\bar{z}_i$, we have a polynomial that is syntactically equal to 1, has degree $2d$ and has monomials size s. The lemma follows by the observation above that implies that there is an NS derivation (with negative literals) of $p[\bar{y}\bar{z}]$ from $(F^{\mathsf{NOR}})^*$ in constant size and without increasing the degree. ◀

The family of graphs we consider for the proof of Theorem 29 is the one defined by Paul et al. [34] and also used by Gilbert and Tarjan [24]. It was shown in [34] that these graphs have large standard pebbling cost, and thus also have large reversible pebbling cost.

▶ **Theorem 35** ([34]). *For every $N \in \mathbb{N}$, there is a DAG of size $\Theta(N)$ that has reversible pebbling cost $\Omega(N/\log N)$.*

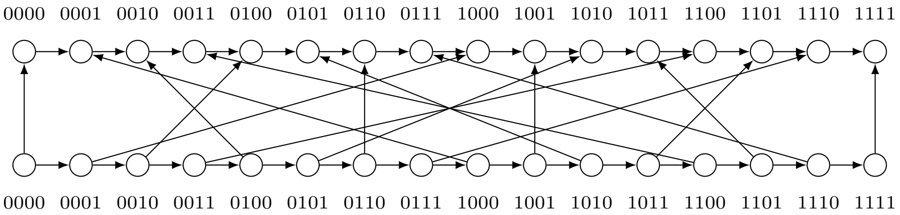


Figure 1 A 2-layer 4-bit-reversal permutation graph.

We are now ready to prove the first theorem of this section.

Proof of Theorem 29. Let G_N be the DAG of size $\Theta(N)$ and pebbling cost $\Omega(N/\log N)$ given by Theorem 35. We define $F_N = \mathrm{Peb}^{\mathsf{NOR}}_{G_N}$. The upper bound follows directly from Lemma 33. For the lower bound, note that Theorem 35 together with Lemma 31 imply that NS requires degree $\Omega(N/\log N)$ to refute Peb_{G_N} and, therefore, by Lemma 32, any NS refutation of F_N must have size $2^{\Omega(N/\log N)}$. ◀

To prove Theorem 30 we consider another family of graphs, based on the so-called *bit-reversal permutation graphs.* Let n be an integer. Given $j \in \{0, 1, \ldots, 2^n - 1\}$ and $\ell \in [n]$, we denote by j_ℓ the ℓth bit of j. Now let $\mathsf{reverse}(j) = \sum_{\ell\in[n]} 2^{n-\ell} j_\ell$ be the integer in $\{0, 1, \ldots, 2^n - 1\}$ obtained by reversing the bit representation of j.

The *k-layer n-bit-reversal permutation graph* consists of k directed path graphs of length 2^n, where we consider vertices in each path to be numbered from 0 to $2^n - 1$, and in between consecutive layers i and $i+1$, for $i \in [k-1]$, there are edges from vertex j in layer i to vertex $\mathsf{reverse}(j)$ in layer $i+1$, for all $j \in \{0, 1, \ldots, 2^n - 1\}$. See Figure 1 for an illustration.

It was shown in [3] that these graphs exhibit a certain smooth time-space trade-off for standard pebbling.

▶ **Proposition 36** ([3]). *Let G be a k-layer n-bit-reversal permutation graph, and let $N = 2^n$. For any s such that $k+1 \le s \le \sqrt{N}/4$ there exists a standard pebbling of G in space $2k^2s+2$ and time $2^{k/2}(N^k/s^{2k-3})$. Furthermore, every standard pebbling of G in space s requires time $2^{-3k}(N^k/s^{2k-3})$.*

By a classical result of [10], which is analysed precisely in [32], we can translate, with some loss both in time and in space, the upper bound in this trade-off to the reversible pebble game.

▶ **Proposition 37** ([10, 32]). *Let G be an arbitrary DAG. If G has a standard pebbling in space s and time $t \ge 2s$, then for any $\epsilon > 0$, G can be reversibly pebbled in simultaneous time $t^{1+\epsilon}/s^\epsilon$ and space $\epsilon(2^{1/\epsilon} - 1)\, s \log(t/s)$.*

▶ **Corollary 38.** *Let $0 < \epsilon \le 1/4$, let G be a k-layer n-bit-reversal permutation graph and let $N = 2^n$. For any s such that $k+1 \le s \le \sqrt{N}/4$ there exists a reversible pebbling of G in space $\frac{s}{k+1}2^{1/\epsilon}k^4 \log N$ and time $2^k(N^{k(1+\epsilon)}/s^{2k-3})$.*

We are now ready to prove Theorem 30.

Proof of Theorem 30. Let $0 < \epsilon \le 1/4$, let G be a k-layer n-bit-reversal permutation graph for $k < 2^{\epsilon n/2}$, and let $N = 2^n$. We define $F_{k,N}$ to be $\mathrm{Peb}^{\mathsf{NOR}}_G$. Item 1 follows from Lemma 34 and the fact that any pebbling formula has linear size NS refutations.

We argue that from Corollary 38 it follows that for any d such that $2^{1/\epsilon}k^4 \log N \leq d \leq \sqrt{N}$, the graph G can be reversibly pebbled in space d and time $n^{k(1+5\epsilon)}/d^{2k-3}$. Item 2 then follows from the correspondence between reversible pebbling and NS refutations (Lemma 31) and Lemma 34. To see why the claim above holds, let

$$s := \frac{d(k+1)}{2^{1/\epsilon}k^4 \log N} \ , \tag{37}$$

which is at least $k+1$ and at most $\sqrt{N}/4$ by the bounds of d. Note, moreover, that

$$s \geq \frac{d}{2^{1/\epsilon}k^3 \log N} \geq \frac{2d}{N^{2\epsilon}} \ , \tag{38}$$

where the last inequality holds for N large enough since $k \leq N^{\epsilon/2}$. By Corollary 38 it then follows that there is a reversible pebbling of G in space d and time

$$2^k \cdot \frac{N^{k(1+\epsilon)}}{s^{2k-3}} \leq 2^k \cdot \left(\frac{N^{2\epsilon}}{2}\right)^{2k-3} \cdot \frac{N^{k(1+\epsilon)}}{d^{2k-3}} \leq \frac{n^{k(1+5\epsilon)}}{d^{2k-3}} \ , \tag{39}$$

as claimed.

Item 3 follows by applying Lemma 32 with $d = k$. ◀

7 Concluding Remarks

Algebraic and semi-algebraic proof systems become more powerful when they can succinctly represent negation of variables using additional formal variables. In some cases this advantage results in exponentially smaller proofs. To witness these separations we built rather artificial formulas. It would be interesting to understand whether this phenomenon occurs for formulas encoding natural problems as well.

More importantly, is this just a theoretical advantage? Practical approaches based on the naive computation of a Gröbner basis nullify any additional expressive power. Since the polynomials $\overline{x}_i = 1 - x_i$ are in the ideal, any such computation eliminates one variable in each pair, potentially causing an exponential blow-up in size along the way. In algebraic circuit verification this is a concrete problem. Some works indeed use new variables for negated literals and have either to avoid or to mitigate such blow-up [36, 27]. Any algorithm that tests ideal membership and wants to make good use of negative literals should be more adaptive than, say, the standard Buchberger's algorithm. It should figure out when to reduce between x_i and $\overline{x}_i$, depending on the context.

Back to the theoretical aspects of this work, the separation formula for sums-of-squares has unbounded width. Since we manage to get formulas of constant width for the others proof systems, we would like to do the same for sums-of-squares. Is this possible? The issue here is not so much our proof techniques, which has been more than enough for all the other proof systems discussed in this paper, but the not so surprising fact that the lower bound technology for sums-of-squares is quite behind the one for NS, PC and SA. It seems fair to say that due to research progress that has happened during the last few years we now have a situation where many of the open problems regarding algebraic proof system and how they relate to one another have been resolved (see for example [11]). We know how different complexity measures relate [26, 2, 23, 33, 22, 4] and whether these systems admit efficient proof search [6, 17]. Yet the situation for sums-of-squares is far from being so positive. We still do not understand the complexity of many important formulas in this proof systems.

References

1 Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version in *STOC '00*.

2 Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. *Proceedings of the Steklov Institute of Mathematics*, 242:18–35, 2003. Preliminary version in *FOCS '01*.

3 Joël Alwen, Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. Cumulative space in black-white pebbling and resolution. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS '17)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:21, 2017.

4 Albert Atserias and Tuomas Hakoniemi. Size-degree trade-offs for Sums-of-Squares and Positivstellensatz proofs. In *Proceedings of the 34th Annual Computational Complexity Conference (CCC '19)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:20, July 2019.

5 Albert Atserias, Massimo Lauria, and Jakob Nordström. Narrow proofs may be maximally long. *ACM Transactions on Computational Logic*, 17(3):19:1–19:30, May 2016. Preliminary version in *CCC '14*.

6 Albert Atserias and Moritz Müller. Automating resolution is NP-hard. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS '19)*, pages 498–509, November 2019.

7 Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS '94)*, pages 794–806, 1994.

8 Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.

9 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

10 Charles H. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766–776, August 1989.

11 Christoph Berkholz. The relation between polynomial calculus, Sherali-Adams, and sum-of-squares proofs. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS '18)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:14, 2018.

12 Joshua Buresh-Oppenheim, Matthew Clegg, Russell Impagliazzo, and Toniann Pitassi. Homogenization and the polynomial calculus. *Computational Complexity*, 11(3-4):91–108, 2002. Preliminary version in *ICALP '00*.

13 Samuel R. Buss and Jakob Nordström. Proof complexity and SAT solving. In Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, chapter 7, pages 233–350. IOS Press, 2nd edition, February 2021.

14 Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 174–183, 1996.

15 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979. Preliminary version in *STOC '74*.

16 Stefan S. Dantchev, Barnaby Martin, and Martin Rhodes. Tight rank lower bounds for the Sherali–Adams proof system. *Theoretical Computer Science*, 410(21–23):2054–2063, May 2009.

17 Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is NP-hard. In *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing (STOC '21)*, June 2021. To appear.

18 Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS '20)*, pages 24–30, November 2020.

19 Susanna F. de Rezende, Jakob Nordström, Or Meir, and Robert Robere. Nullstellensatz size-degree trade-offs from reversible pebbling. In *Proceedings of the 34th Annual Computational Complexity Conference (CCC '19)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:16, 2019.

20 Yuval Filmus, Massimo Lauria, Jakob Nordström, Noga Ron-Zewi, and Neil Thapen. Space complexity in polynomial calculus. *SIAM Journal on Computing*, 44(4):1119–1153, August 2015. Preliminary version in *CCC '12*.

21 Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends in Theoretical Computer Science*, 14(1–2):1–221, December 2019.

22 Nicola Galesi, Leszek Kołodziejczyk, and Neil Thapen. Polynomial calculus space and resolution width. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS '19)*, pages 1325–1337, November 2019.

23 Nicola Galesi and Massimo Lauria. Optimality of size-degree trade-offs for polynomial calculus. *ACM Transactions on Computational Logic*, 12(1):4:1–4:22, November 2010.

24 John R. Gilbert and Robert Endre Tarjan. Variations of a pebble game on graphs. Technical Report STAN-CS-78-661, Stanford University, 1978. Available at `http://infolab.stanford.edu/TR/CS-TR-78-661.html`.

25 Dima Grigoriev and Nicolai Vorobjov. Complexity of Null- and Positivstellensatz proofs. *Annals of Pure and Applied Logic*, 113(1–3):153–160, 2001.

26 Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

27 Daniela Kaufmann, Armin Biere, and Manuel Kauers. From DRUP to PAC and back. In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE '20)*, pages 654–657, March 2020.

28 Jan Krajíček. *Proof Complexity*, volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, March 2019.

29 Balakrishnan Krishnamurthy. Short proofs for tricky formulas. *Acta Informatica*, 22(3):253–275, 1985.

30 Jean B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO '01)*, volume 2081 of *Lecture Notes in Computer Science*, pages 293–303. Springer, 2001.

31 Massimo Lauria and Jakob Nordström. Tight size-degree bounds for sums-of-squares proofs. *Computational Complexity*, 26(3):911–948, December 2017. Preliminary version in *CCC '15*.

32 Robert Y. Levin and Alan T. Sherman. A note on Bennett's time-space tradeoff for reversible computation. *SIAM Journal on Computing*, 19(4):673–677, August 1990. `doi:10.1137/0219046`.

33 Mladen Mikša and Jakob Nordström. A generalized method for proving polynomial calculus degree lower bounds. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 467–487, June 2015.

34 Wolfgang J. Paul, Robert Endre Tarjan, and James R. Celoni. Space bounds for a game on graphs. *Mathematical Systems Theory*, 10:239–251, 1977.

35 Aaron Potechin. Sum of squares bounds for the ordering principle. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 38:1–38:37. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.CCC.2020.38`.

36 Amr Sayed-Ahmed, Daniel Große, Mathias Soeken, and Rolf Drechsler. Equivalence checking using Gröbner bases. In *Proceedings of the 16th Conference on Formal Methods in Computer-Aided Design (FMCAD '16)*, pages 169–176, 2016.

37 Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM Journal on Computing*, 33(5):1171–1200, 2004. Preliminary version in *FOCS '02*.

38 Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411–430, 1990.

39 Gunnar Stålmarck. Short resolution proofs for a sequence of tricky formulas. *Acta Informatica*, 33(3):277–280, May 1996.

Matrix Rigidity Depends on the Target Field

László Babai
University of Chicago, IL, USA

Bohdan Kivva
University of Chicago, IL, USA

Abstract

The *rigidity* of a matrix A for target rank r is the minimum number of entries of A that need to be changed in order to obtain a matrix of rank at most r (Valiant, 1977).

We study the dependence of rigidity on the target field. We consider especially two natural regimes: when one is allowed to make changes only from the field of definition of the matrix ("strict rigidity"), and when the changes are allowed to be in an arbitrary extension field ("absolute rigidity").

We demonstrate, apparently for the first time, a separation between these two concepts. We establish a *gap of a factor of* $3/2 - o(1)$ between strict and absolute rigidities.

The question seems especially timely because of recent results by Dvir and Liu (*Theory of Computing*, 2020) where important families of matrices, previously expected to be rigid, are shown not to be absolutely rigid, while their strict rigidity remains open. Our lower-bound method combines elementary arguments from algebraic geometry with "untouched minors" arguments.

Finally, we point out that more families of long-time rigidity candidates fall as a consequence of the results of Dvir and Liu. These include the incidence matrices of projective planes over finite fields, proposed by Valiant as candidates for rigidity over $\mathbb{F}_2$.

2012 ACM Subject Classification Theory of computation; Theory of computation → Complexity theory and logic

Keywords and phrases Matrix rigidity, field extension

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.41

Funding The work of the authors has been partially supported by L. Babai's NSF Grant CCF 1718902. All statements made are solely due to the authors and have not been evaluated or endorsed by the NSF.

Acknowledgements The senior author thanks Zeev Dvir and Allen Liu for clarifications regarding their results. The authors thank an anomymous reviewer for pointing out the work of Samorodnitsky et al.

1 Introduction

1.1 Matrix rigidity. Dependence on the field

Matrix rigidity was introduced by Leslie Valiant in his seminal paper [16] as a tool to prove lower bounds on the complexity of linear arithmetic circuits (where each gate computes a linear combination of its inputs). Such circuits compute linear functions $x \mapsto Ax$ for some matrix A. Razborov [12] linked the rigidity concept to separating the polynomial hierarchy in communication complexity.

▶ **Definition 1** (Matrix rigidity). *Let $\mathbb{L}/\mathbb{K}$ be a field extension ($\mathbb{K}$ is a subfield of $\mathbb{L}$) and let $A \in \mathbb{K}^{n\times m}$. Denote by $R_{\mathbb{L}}(A, r)$ the minimum number of non-zero entries in a matrix $Z \in \mathbb{L}^{n\times m}$ for which $A + Z$ has rank at most r. The function $R_{\mathbb{L}}(A, \cdot)$ is called the* matrix rigidity function *of A over $\mathbb{L}$.*

The definition of rigidity depends on a pair of fields: $\mathbb{K}$, the field in which the matrix lives, and the extension field $\mathbb{L} \supseteq \mathbb{K}$, over which the changes to A are to be made. There are two natural regimes in which we especially propose to study matrix rigidity.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 41; pp. 41:1–41:26
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We say that $\mathbb{K}$ is the *field of definition* of a matrix $A \in \mathbb{F}^{n\times m}$ (where $\mathbb{F}$ is a field) if $\mathbb{K}$ is the smallest subfield of $\mathbb{F}$ containing all elements of A.

▶ **Definition 2** (Strict rigidity). *Let $\mathbb{K}$ denote the field of definition of the matrix A. We call the function $R_{\mathbb{K}}(A, \cdot)$ the* strict rigidity *function of A.*

▶ **Definition 3** (Absolute rigidity). *We define the* absolute rigidity *of A as*

$$R^*(A, r) = \min_{\mathbb{L}} R_{\mathbb{L}}(A, r),$$

where the minimum ranges over all extension fields $\mathbb{L}$ of the field of definition of A.

The main result of this paper shows, apparently for the first time, that the notions of strict and absolute rigidity are indeed different. We establish a *gap of a factor of* $3/2 - o(1)$ between these quantities.

The *degree* of a field extension $\mathbb{L}/\mathbb{K}$ is the dimension of $\mathbb{L}$ over $\mathbb{K}$. Extensions of degree 2 are called *quadratic extensions.*

▶ **Theorem 4.** *Let $\mathbb{K}$ be a field of characteristic zero and let $\mathbb{L}$ be a quadratic extension of $\mathbb{K}$. For every r there exists a $2r \times 2r$ matrix A_r over $\mathbb{K}$ such that $R_{\mathbb{L}}(A_r, r) \le 2r$ while $R_{\mathbb{K}}(A_r, r) \ge 3r - 2$.*

Note that the second bound requires a lower-bound technique for rigidity.

We expect much larger gaps; indeed, larger gaps will be needed to show the depedence of Valiant-rigidity on the field (see below).

We also point out that for any matrix A over a field $\mathbb{K}$ we have $R^*(A, r) = R_{\overline{\mathbb{K}}}(A, r)$, where $\overline{\mathbb{K}}$ denotes the algebraic closure of $\mathbb{K}$ (Sec. 4). In other words, for every matrix A, absolute rigidity can be achieved over a finite extension of the field of definition of A (see Cor. 55). However, effective bounds on the degree of this extension remain an open question.

A similar result holds for linear arithmetic circuits (Prop. 56).

1.2 Valiant-rigidity, non-rigidity results

While the distinction between strict and absolute rigidity seems natural and we find it somewhat surprising that apparently it has not previously been addressed, unexpected recent non-rigidity results give particular timeliness to this question.

To discuss these results, we need some asymptotic terminology.

We say that the *order* of an $n \times n$ matrix is n. We use the term *"family of square matrices"* to mean a set of square matrices of unbounded order.

▶ **Definition 5** (Valiant-rigid). *Let $\mathcal{F}$ be a family of square matrices. For $A \in \mathcal{F}$, let $\mathbb{K}(A)$ denote the field of definition of A, and let $\mathbb{L}(A)$ be an extension field of $\mathbb{K}(A)$. We say that the family $\mathcal{F}$ is* Valiant-rigid over the extension fields $\mathbb{L}(A)$ *if there exists $\epsilon > 0$ such that for every function $r(n) = O(n/\log\log n)$, for all matrices A in the family, $R_{\mathbb{L}(A)}(A, r(n_A)) = \Omega(n_A^{1+\epsilon})$, where n_A denotes the order of A.*

It seems the term "Valiant-rigid" was introduced in [3] (but their definition did not consider the effect of the field).

The terms "strictly" and "absolutely" Valiant-rigid should now be self-explanatory.

For one of the families, long believed to be rigid, the family of Walsh–Hadamard matrices, Alman and Williams [2] proved that it is in fact not strictly Valiant-rigid.

Recently, Dvir and Liu [4] proved that no family of Discrete Fourier Transform (DFT) matrices for abelian groups G and no family of G-circulant matrices (see Def. 57) is absolutely Valiant-rigid. However, strict Valiant-rigidity of these families remains an open problem.

Note that the Walsh–Hadamard matrices are the DFT matrices for elementary abelian 2-groups, yet the Dvir–Liu result does not fully reproduce the Alman–Williams result for these matrices precisely because of the target field: while Dvir and Liu prove that these matrices are not absolutely Valiant-rigid, Alman and Williams proved the stronger result that these matrices are not strictly rigid.

In Section 5 we point out that the following families of long-time rigidity candidates also fall as a consequence of the results of Dvir and Liu.

1. No family of Paley–Hadamard matrices is absolutely Valiant-rigid. (Note: The orders of these Hadamard matrices are exponentially denser than the orders of the Walsh–Hadamard matrices, shown not to be strictly Valiant-rigid by Alman and Williams [2].)
2. No family of point–hyperplane incidence matrices of Galois geometries (projective geometries over finite fields) is strictly Valiant-rigid over any fixed finite field. (Note: The incidence matrices of finite projective planes were proposed by Valiant [16] as candidates for rigidity over $\mathbb{F}_2$.)
3. No family of point–hyperplane incidence matrices of Galois geometries is absolutely Valiant-rigid in characteristic zero.
4. No family of Vandermonde matrices whose generators form a geometric progression is absolutely Valiant-rigid.

We should remind the reader that absolute rigidity is a stronger property than strict rigidity; and therefore the statement that a matrix is "not strictly rigid" is stronger than the statement that it is "not absolutely rigid."

We mention that Samorodnitsky *et al.* [13] proved rigidity lower bounds for the point-hyperplane incidence matrices of Galois geometries (projective spaces over finite fields), conditional on their conjecture that the set of normalized $\{0,1\}$-vectors arising from an arbitrary low-dimensional subspace of $\mathbb{F}_2^n$ admits non-trivial approximation by a low-dimensional Euclidean space. They show that if their conjecture is true, then there exists $\delta > 0$, such that $R_{\mathbb{F}_2}(V_d, n^{(2/d)+\delta}) \geq n^{1-2/d}$, where V_d is the $n \times n$ point–hyperplane incidence matrix of the d-dimensional Galois geometry $PG(d,q)$. Our results do not refute their conjecture, as we prove upper bounds for the target rank of an order $n \cdot \exp(-(\log n)^c)$, while [13] aims at a much smaller target rank, $n^{(2/d)+\delta}$.

1.3 Implications to complexity theory

This line of work may lead to peculiar consequences in complexity theory. Gaps between strict and absolute rigidity raise the prospect that rational linear functions may be easier to compute by arithmetic circuits over larger fields than over $\mathbb{Q}$.

▶ **Problem 6.** Does there exist a family of square matrices A over $\mathbb{Q}$ such that the linear functions $x \mapsto Ax$ can be computed by logarithmic-depth, linear-size circuits over $\mathbb{C}$ but not over $\mathbb{Q}$?

While $\mathbb{C}$ can be replaced by a finite extension of $\mathbb{Q}$ without changing the topology of the circuit (Prop. 56), a field extension of bounded degree will not create a gap in circuit complexity. Indeed, if the degree of the extension $\mathbb{L}/\mathbb{K}$ is k then operations in $\mathbb{L}$ can be simulated by operations on vectors of length k over $\mathbb{K}$. So our belief that strong separation of rigidity may exist already for quadratic extensions (Conj. 9), if true, will not help.

1.4 Our construction

We make the following "standard assumption."

($*$) Let $\mathbb{K}$ be a field of characteristic zero and let $\mathbb{L}/\mathbb{K}$ be a quadratic extension.

We prove that under this assumption, the rigidity with respect to $\mathbb{K}$ in general does not equal the rigidity with respect to $\mathbb{L}$. In order to show this, for some $A \in \mathbb{K}^{n\times n}$ and $r, k \geq 1$ one needs to establish both an upper bound and a lower bound,

$$\text{(UB)}\quad R_{\mathbb{L}}(A, r) \leq k \qquad\qquad \text{(LB)}\quad R_{\mathbb{K}}(A, r) > k.$$

It is clear that for all $A \in \mathbb{K}^{n\times n}$, the inequality $R_{\mathbb{K}}(A, r) \leq (n-r)^2$ is satisfied. In [16], Valiant showed that for an infinite field $\mathbb{K}$, almost all matrices $A \in \mathbb{K}^{n\times n}$ have maximal possible absolute rigidity $R^*(A, r) = (n-r)^2$. In particular, this means that we should not expect (UB) to hold, unless A is selected in some special way.

We take the following approach. In order to automatically satisfy (UB) we start with a matrix $M \in \mathbb{L}^{n\times n}$ of rank r that has at most k entries not in $\mathbb{K}$. Then every matrix A, obtained from M by replacing these entries with elements from $\mathbb{K}$, satisfies (UB). Hence, we only need to show that for a proper choice of M and for a proper choice of changes for elements not in $\mathbb{K}$, A satisfies (LB).

By our standard assumption ($*$), we can write $\mathbb{L} = \mathbb{K}[\omega]$ for some $\omega \in \mathbb{L}$ with $\omega^2 \in \mathbb{K}$. We focus on the following (algebraic) sets of matrices:

$$\mathcal{D}_r(\mathbb{K}, \omega) = \{M \in \mathbb{K}^{2r\times 2r} \mid \operatorname{rank}(M + \omega I) \leq r\}, \tag{1}$$

$$\mathcal{C}_r(\mathbb{K}, \omega) = \{M + D \in \mathbb{K}^{2r\times 2r} \mid M \in \mathcal{D}_r(\mathbb{K}, \omega),\ D \in \mathbb{K}^{2r\times 2r} \text{ is diagonal}\}. \tag{2}$$

By definition, for every $A \in \mathcal{C}_r(\mathbb{K}, \omega)$, $R_{\mathbb{L}}(A, r) \leq 2r$. Our main result is the following.

▶ **Theorem 7.** *Let $r \geq 3$. There exists a matrix $A \in \mathcal{C}_r(\mathbb{K}, \omega)$ with $R_{\mathbb{K}}(A, r) \geq 3r - 2$.*

As an immediate corollary, we establish the promised gap between the strict and the absolute rigidities.

▶ **Theorem 8.** *Let $\mathbb{K}$ and $\mathbb{L}$ satisfy the standard assumption ($*$). Then, for every $\varepsilon > 0$ and all sufficiently large r there exists a square matrix $M \in \mathbb{K}^{2r\times 2r}$ satisfying $R_{\mathbb{K}}(M, r) \geq (3/2 - \varepsilon)R_{\mathbb{L}}(M, r)$.*

We conjecture that much larger separation is possible.

▶ **Conjecture 9.** *Let $\mathbb{L} = \mathbb{Q}[\sqrt{2}]$. There exist $\varepsilon > 0$ and matrices M of arbitrarily large order $n = 2r$ such that $R_{\mathbb{Q}}(M, r) \geq n^{1+\varepsilon}$, while $R_{\mathbb{L}}(M, r) \leq O(n)$.*

In particular, we expect that such matrices M can be found in $\mathcal{C}_r(\mathbb{Q}, \sqrt{2})$.

We also ask whether the maximum possible rigidity can be achieved for matrices in $\mathcal{C}_r$.

▶ **Problem 10.** Is it true that for infinitely many r there exists a matrix $A \in \mathcal{C}_r(\mathbb{Q}, \sqrt{2})$ with $R_{\mathbb{Q}}(A, r) = r^2$?

1.5 Known lower bounds on rigidity: untouched minors

Despite decades of effort, progress on proving lower bounds on rigidity for explicit families of matrices has been limited. The best known general lower bound for a family of explicit $n \times n$ matrices A has the form $R^*(A, r) = \Omega((n^2/r)\log(n/r))$ [5, 14]. This lower bound is achieved through the "untouched minors argument": If all $(r+1) \times (r+1)$ minors of a matrix A are non-singular, then to reduce the rank of A to r, one needs to change at least one entry in every such minor. However, as discussed in [10], that is the best bound one can achieve through this argument.

For some semi-explicit families of matrices, stronger lower bounds are known. For $n \times n$ matrices whose entries are square roots of distinct prime numbers, Lokam [11] gives optimal, $\Omega(n^2)$ absolute rigidity for rank $r \leq n/17$. This result uses an algebraic dimension concept introduced by Shoup and Smolensky [15].

In the domain of reduced randomness, Goldreich and Tal [6] show that for random $n \times n$ Toepliz matrices A over $\mathbb{F}_2$ the bound $R_{\mathbb{F}_2}(A, r) = \Omega(n^3/(r^2 \log n))$ holds for $r \geq \sqrt{n}$.

1.6 Key steps of the proof of Theorem 7. Organization of the paper

First, observe that untouched minors arguments alone cannot answer our question; they do not distinguish between entries from $\mathbb{K}$ and $\mathbb{L}$. In order to prove the lower bound in Theorem 7 we use a combination of the untouched minors argument and arguments based on elements of algebraic geometry about the structure of $\mathcal{D}_r(\mathbb{K}, \omega)$.

We begin by noticing that for almost all matrices $A \in \mathcal{C}_r(\mathbb{K}, \omega)$, all $(r+1) \times (r+1)$ minors are non-singular (Lemma 32). So, an untouched minors argument can be used to show that if $R_{\mathbb{K}}(A, r) \leq 3r - 3$, then the entries that are being changed have a "nice" layout inside $[2r] \times [2r]$. More precisely, we argue that then there are $(r+2)$ columns with at most 1 element changed in each of them (see Section 3.1).

Next, assume that for some $M \in \mathcal{D}_r(\mathbb{K}, \omega)$ and every diagonal matrix D we have $R_{\mathbb{K}}(M + D, r) \leq 3r - 3$. We can argue that since there are only finitely many choices for $3r - 3$ entries in $[2r] \times [2r]$, there should be a fixed set π of $3r - 3$ cells in $[2r] \times [2r]$, such that for a "large" set of diagonal matrices D, the rank of $M + D$ can be made $\leq r$ by only changing entries inside π (see Section 3.2).

Finally, we exploit the geometry of the set $\mathcal{D}_r = \mathcal{D}_r(\mathbb{K}, \omega)$ to show that for almost all matrices $M \in \mathcal{D}_r$ no such fixed π exists. In order to do this, we show that among $2r^2$ entries in arbitrary r columns of $M \in \mathcal{D}_r$ there is no algebraic dependence imposed by $\mathcal{D}_r$ (see Section 2). Next, we consider a properly chosen set of $r + 2$ columns with at most one entry from π in each of them, and exploit the fact that we have sufficiently many algebraic degrees of freedom for the entries in these columns so that changing entries in π typically is not sufficient to make the rank of these columns to be r (see Sections 3.3 - 3.5). This last step is the hardest part of the proof and requires us to consider several cases.

We present the parts of the proof in a slightly different order than described above. The geometry of the set $\mathcal{D}_r$ is studied in Section 2. Other parts of the proof are contained in Section 3. We combine these parts into a complete proof of Theorem 7 in Section 3.5.

In Section 4 we show that finite extensions suffice for absolute rigidity. In Section 5 we prove the refutation of rigidity candidates mentioned in Sec. 1.2.

We review some basic concepts from algebraic geometry over arbitrary fields in Appendix A. The proofs omitted in Section 3.4 are provided in Appendix B. The model-theoretic reduction to countable fields is outlined in Appendix C. In Appendix D we exhibit a concrete 5×5 matrix of strict rigidity 9 and absolute rigidity 8. Some open problems are raised in Sec. 1.7.

1.7 Open problems

The most intriguing question to come out of this work is Problem 6, the separation of the linear arithmetic complexity of linear functions by the extension field permitted in the circuit.

Strong separation between strict and absolute rigidities is suggested in Conjecture 9.

For a matrix over a field $\mathbb{K}$, absolute rigidity can be achieved over a finite extension of $\mathbb{K}$ (Prop. 49). However, that result is not effective.

▶ **Problem 11.** Is there a computable function f that maps rational matrices to positive integers, such that the absolute rigidity of any rational matrix A can be achieved over an extension of $\mathbb{Q}$ of degree $\leq f(A)$? Can such an f be made a function of the dimensions of the matrix A?

Recent non-rigidity results [2, 3, 4] inspire the following problems.

We remind the reader that by a *family* of square matrices we mean a set of square matrices of unbounded order.

In our submission to this conference (Feb. 15, 2021) we proposed the following conjecture.

▶ **Conjecture 12.** *Let $\mathcal{F}$ be a finite set of matrices over $\mathbb{C}$. Let $\mathcal{A}$ denote the set of all possible Kronecker products of these matrices (taking each member of $\mathcal{F}$ any number of times). Then no subfamily of $\mathcal{A}$ is Valiant-rigid over $\mathbb{C}$.*

We stated that this would generalize the result that the DFT matrices for abelian groups of bounded exponent are not absolutely Valiant rigid [2, 4].

On Feb. 24, 2021, a paper by Josh Alman appeared on arXiv [1] that raises the same question and answers it in the positive in the case that all matrices in the family $\mathcal{F}$ have the same order. This restriction was subsequently removed by one of us, confirming Conjecture 12 [8]. That paper also exponentially improves Alman's non-rigidity exponent. Like Alman's, our result establishes strict non-rigidity. We state the main result of [8].

▶ **Theorem 13** (Kivva [8]). *Given $d \geq 2$ and $\varepsilon > 0$, there exists $\gamma > 0$ such that the following holds for any sequence of matrices $M_1, \ldots, M_n$ of respective orders $d_i \leq d$ over the field $\mathbb{F}$. Let $M = \otimes_{i=1}^{n} M_i$ and $N = \prod_{i=1}^{n} d_i$. If $N \geq d^{1/\gamma}$ then $R_{\mathbb{F}}(M, N^{1-\gamma}) \leq N^{1+\epsilon}$. Here γ can be chosen to be $\gamma = \Omega\left(\frac{1}{d^{3/2}\log^3(d)} \cdot \frac{\varepsilon^2}{\log^2(1/\varepsilon)}\right)$.*

The following problem remains open.

▶ **Problem 14.** Does there exist a strictly Valiant-rigid family of rational circulant matrices?

No such family is absolutely rigid by Dvir and Liu [4].

2 Basic properties of $\mathcal{D}_r$

We continue to make our standard assumption $(*)$. Let $\mathbb{L} = \mathbb{K}[\omega]$ be a quadratic extension, where $\omega^2 \in \mathbb{K}$. $\mathbb{F}$ denotes an arbitrary infinite field (not necessarily of characteristic zero).

Additionally, we assume that $\mathbb{L}$ is a subfield of $\mathbb{C}$. This assumption can be made without loss of generality. Indeed, a simple model-theoretic argument shows that we can assume that $\mathbb{K}$ is countable (Prop. 91). The proof of Prop. 91 can also be adapted to reducing Theorem 7 to countable fields.

Since $\mathbb{K}$ and ω are fixed, we use the notation $\mathcal{D}_r = \mathcal{D}_r(\mathbb{K}, \omega)$ and $\mathcal{C}_r = \mathcal{C}_r(\mathbb{K}, \omega)$. Recall that these are algebraic sets in $\mathbb{K}^{2r} \times \mathbb{K}^{2r}$.

2.1 Matrices over $\mathbb{L}$ of low rank and with few entries outside $\mathbb{K}$

We start by giving a short motivation for the family $\mathcal{D}_r$. Recall the following elementary fact from linear algebra.

▶ **Fact 15.** Let $M \in \mathbb{F}^{n\times n}$ be a matrix of rank r. Let $L \in \mathbb{F}^{n\times r}$ be a matrix consisting of r linearly independent columns of M. Then there exists $R \in \mathbb{F}^{r\times n}$ such that $M = LR$.

Let $L \in \mathbb{K}[\omega]^{n\times r}$ and $R \in \mathbb{K}[\omega]^{r\times n}$. Denote the i-th row of L by $a_i + b_i\omega$ and j-th column by $x_j + y_j\omega$, where $a_i, b_i, x_j, y_j \in \mathbb{K}^r$.

▶ **Definition 16.** *For $x, y \in \mathbb{K}^n$ define $\langle x, y\rangle = \sum\limits_{i=1}^{n} x_i \cdot y_i$.*

▶ **Observation 17.** *$(LR)_{ij} \in \mathbb{K}$ if and only if $\langle x_j, b_i\rangle + \langle y_j, a_i\rangle = 0$.*

▶ Remark 18. For a field extension of degree k, a similar criterion consists of $k-1$ linear equations to be satisfied by components of R.

▶ **Corollary 19.** *Take $n = 2r$. Then for every choice of $2r$ linearly independent vectors $(a_i, b_i) \in \mathbb{K}^{2r}$ there exists a unique choice of $2r$ vectors (x_i, y_i) such that LR is in $\mathcal{D}_r + \omega I$.*

Note that if $n \geq 2r$, and L is a generic $n \times r$ matrix, we should expect at least $n(n-2r+1)$ entries of LR to be from $\mathbb{L} \setminus \mathbb{K}$. At the same time, $R_{\mathbb{K}}(LR, r) \leq (n-r)^2$. We prefer the quotient of these numbers to be as small as possible, which is achieved for $n = 2r$ (if $n \geq 2r$).

2.2 Geometry of $\mathcal{D}_r$

In this section we study the geometry of the set $\mathcal{D}_r$. See Appendix A for some basic definitions and facts from algebraic geometry that are used in this paper.

▶ **Definition 20** (proj_S)**.** *For a matrix $A \in \mathbb{F}^{n\times n}$ and $S \subseteq [n]$ define $\mathrm{proj}_S(A)$ to be the matrix consisting of columns of A with indices in S.*

▶ **Definition 21** (Small set)**.** *We say that a set $A \subseteq \mathbb{K}^n$ is* small *(in $\mathbb{K}^n$) if it is contained in a proper algebraic subset in $\mathbb{K}^n$.*

▶ **Definition 22** (${}^\sigma M^\tau$)**.** *For permutations $\sigma \in S_n$, $\tau \in S_m$ and an $n \times m$ matrix M, define ${}^\sigma M^\tau$ to be the matrix obtained from M by permuting rows by σ and columns by τ.*

Our first goal is to show that for $S \subseteq [2r]$ with $|S| = r$ only a small set of matrices from $\mathbb{K}^{2r\times r}$ is not in the image of $\mathrm{proj}_S : \mathcal{D}_r \to \mathbb{K}^{2r\times r}$. Note that for every permutation $\sigma \in S_{2r}$ and $M \in \mathcal{D}_r$ we have ${}^\sigma M^\sigma \in \mathcal{D}_r$. Thus, it is sufficient to study $\mathrm{proj}_{[r]}$.

▶ **Lemma 23.** *Let $A_1, A_2 \in \mathbb{K}^{r\times r}$. Assume that A_2 is invertible. Then*

$$\begin{pmatrix} A_1 & -A_1^2A_2^{-1} + \omega^2 A_2^{-1} \\ A_2 & -A_2A_1A_2^{-1} \end{pmatrix} \in \mathcal{D}_r. \tag{3}$$

Proof. Observe that

$$\begin{pmatrix} A_1 + I\omega & -A_1^2A_2^{-1} + \omega^2 A_2^{-1} \\ A_2 & -A_2A_1A_2^{-1} + I\omega \end{pmatrix} = \begin{pmatrix} A_1 + I\omega \\ A_2 \end{pmatrix} \cdot \begin{pmatrix} I, & -A_1A_2^{-1} + A_2^{-1}\omega \end{pmatrix}. \qquad ◀$$

Next, we observe that a simple condition on $M \in \mathcal{D}_r$ guarantees that $\mathrm{proj}_{[r]}$ is injective.

▶ **Lemma 24.** *Let $A_1, A_2 \in \mathbb{K}^{r\times r}$ and $M \in \mathcal{D}_r$. Assume that $\mathrm{proj}_{[r]}(M+\omega I) = \begin{pmatrix} A_1+\omega I \\ A_2 \end{pmatrix}$ has rank r (over $\mathbb{L}$). Then, A_2 is invertible, and M is uniquely determined by $\mathrm{proj}_{[r]}(M)$, and we have*

$$M = \phi_{[r]}\begin{pmatrix} A_1 \\ A_2 \end{pmatrix} := \begin{pmatrix} A_1 & -A_1^2A_2^{-1}+\omega^2A_2^{-1} \\ A_2 & -A_2A_1A_2^{-1} \end{pmatrix}.$$

Proof. Denote $L = \mathrm{proj}_{[r]}(M+\omega I)$. Since $\mathrm{rank}(L) = r$, $M+\omega I = LR$ for some $R \in \mathbb{L}^{r\times 2r}$. Let $R = (X_1+Y_1\omega, X_2+Y_2\omega)$ for $X_1, Y_1, X_2, Y_2 \in \mathbb{K}^{r\times r}$. The inclusion $M \in \mathcal{D}_r$ imposes the following constraints.

$$A_1Y_1+X_1 = I, \qquad A_1Y_2+X_2 = 0, \qquad A_2Y_1 = 0 \quad \text{and} \quad A_2Y_2 = I. \tag{4}$$

The last equality implies that A_2 is invertible. Therefore $Y_1 = 0$, $Y_2 = A_2^{-1}$, $X_1 = I$ and $X_2 = -A_1A_2^{-1}$. ◀

Define $U_{[r]}$ to be the set of $X \in \mathbb{K}^{2r\times r}$ such that the matrix formed by the last r rows of X is non-singular. Note that $\phi_{[r]} : U_{[r]} \to \mathbb{K}^{2r\times r}$ defined in the lemma above is a regular map according to Def. 82 (see Lemma 83).

Due to Lemma 24, it will be convenient to work with the following subset of $\mathcal{D}_r$.

$$\mathcal{D}'_r = \{M \in \mathcal{D}_r \mid \forall S \subset [2r],\ |S| = r:\ \mathrm{rank}(\mathrm{proj}_S(M+\omega I)) = r\}. \tag{5}$$

Let $I_{2r,r} \in \mathbb{K}^{2r\times r}$ be the identity matrix padded with r zero rows. Define

$$\mathcal{L} = \{L \in \mathbb{K}^{2r\times r} \mid \text{all } r\times r \text{ minors of } L+\omega I_{2r,r} \text{ are non-singular}\}. \tag{6}$$

▶ **Observation 25.** *$\mathcal{L}$ is an irreducible Zariski-open subset of $\mathbb{K}^{2r\times r}$.*

Proof. The set of L for which $L+\omega I_{2r,r}$ has a singular $r\times r$ minor is a finite union of proper Zariski-closed subsets of $\mathbb{K}^{2r\times r}$. Since, $\mathbb{K}^{2r\times r}$ is irreducible, this union is a proper Zariski-closed subset. Hence $\mathcal{L}$ is Zariski-open and it is irreducible, as a Zariski-open subset of an irreducible set. ◀

Then, by Lemmas 23 – 24, for every $L \in \mathcal{L}$ there exists a unique matrix $M \in \mathcal{D}_r$ with $\mathrm{proj}_{[r]}(M) = L$. For $\phi_{[r]}$ as in Lemma 24, define

$$\mathcal{D}^*_r = \{\phi_{[r]}(L)^T \mid L \in \mathcal{L}\}. \tag{7}$$

▶ **Lemma 26.** *The set $\mathcal{D}^*_r$ is an irreducible quasi-affine variety. Moreover, $\mathcal{D}^*_r \subseteq \mathcal{D}'_r$, and for every $S \subseteq [2r]$ with $|S| = r$ only a small set of matrices in $\mathbb{K}^{2r\times r}$ is not in $\mathrm{proj}_S(\mathcal{D}^*_r) \subseteq \mathbb{K}^{2r\times r}$.*

Proof. Observe, that for $L \in \mathcal{L}$ and $M = \phi_{[r]}(L)$, any r distinct rows of $M+\omega I$ are linearly independent. Therefore, $M^T \in \mathcal{D}'_r$, and so $\mathcal{D}^*_r \subseteq \mathcal{D}'_r$. The set $\mathcal{L} \subseteq \mathbb{K}^{2r\times r}$ is a non-empty Zariski-open irreducible set. Recall that $\mathcal{D}_r$ is an affine algebraic set. By Lemma 24, the set $\mathcal{D}^*_r$ is equal to $\psi^{-1}_{[r]}(\mathcal{L}) \cap \mathcal{D}_r$, where $\psi_{[r]} : \mathbb{K}^{2r\times 2r} \to \mathbb{K}^{2r\times r}$ is defined by $M \mapsto \mathrm{proj}_{[r]}(M^T)$. Since $\psi_{[r]}$ is regular, $\mathcal{D}^*_r$ is a quasi-affine algebraic set. The map $(\phi_{[r]})^T : \mathcal{L} \to \mathbb{K}^{2r\times 2r}$ is regular, so $\mathcal{D}^*_r$ is irreducible (see Obs. 76).

For every $S \subseteq [2r]$ with $|S| = r$, let $\phi_S : U_S \to \mathbb{K}^{2r\times 2r}$ (defined similarly as in Lemma 24) be an inverse function to proj_S, where U_S is a Zariski-open subset of $\mathbb{K}^{2r\times r}$ where ϕ_S is well-defined. The map ϕ_S is regular and injective. Therefore, by Lemma 24, $\mathrm{proj}_S(\mathcal{D}^*_r) = (\phi_S \circ \psi_{[r]})^{-1}(\mathcal{L})$, and so it is a Zariski-open subset of $\mathbb{K}^{2r\times r}$. Hence, only a small set of matrices in $\mathbb{K}^{2r\times r}$ is not in $\mathrm{proj}_S(\mathcal{D}^*_r) \subseteq \mathbb{K}^{2r\times r}$. ◀

Def. 78 defines the notion of "almost all elements" of an irreducible quasi-variety. Since we have not proved that $\mathcal{D}_r$ is irreducible, we need a special definition to formalize our references to "almost all elements of $\mathcal{D}_r$."

▶ **Definition 27** (Almost all elements of $\mathcal{D}_r$). *We shall say that some property holds for* almost all *matrices in $\mathcal{D}_r$ if it holds for almost all elements of $\mathcal{D}_r^*$.*

We believe that, in fact, $\mathcal{D}_r$ is irreducible. If that is the case, Def. 27 remains consistent with Def. 78.

By Obs. 79, if each of a finite number of properties holds for almost all elements of $\mathcal{D}_r$, then they all hold simultaneously for almost all elements of $\mathcal{D}_r$.

▶ Remark 28. If $\mathcal{A} \subseteq \mathcal{D}_r^*$ is such that for some $S \subseteq [2r]$ with $|S| = r$ the set $\operatorname{proj}_S(\mathcal{A})$ is small in $\mathbb{K}^{2r\times r}$, then by Lemma 26, almost all matrices in $\mathcal{D}_r$ are not in $\mathcal{A}$.

▶ **Definition 29** ($\mathcal{D}_r^{\#}$). *Let $\mathcal{D}_r^{\#}$ denote the set of matrices $M \in \mathcal{D}_r^*$ such that for all $k \le r$ every $k \times k$ minor of M is non-singular.*

▶ **Corollary 30.** *$\mathcal{D}_r^{\#}$ is a non-empty Zariski-open subset of $\mathcal{D}_r^*$.*

Proof. Let X be the Zariski-open subset of $\mathbb{K}^{2r\times r}$ consisting of matrices with all $k \times k$ minors being non-singular for all $k \le r$. Then, $\mathcal{D}_r^{\#} = \bigcap_{S\subseteq[2r],\ |S|=r} \left(\mathcal{D}_r^* \cap \operatorname{proj}_S^{-1}(X)\right)$. ◀

▶ **Definition 31** ($\operatorname{Diag}(\mathbb{F}^{n\times m})$). *Define $\operatorname{Diag}(\mathbb{F}^{n\times m})$ to be the set of matrices in $\mathbb{F}^{n\times m}$ that have non-zero entries only in the cells with indices $\{(i,i) \mid 1 \le i \le \min(n,m)\}$.*

▶ **Lemma 32.** *For every $M \in \mathcal{D}_r^{\#}$ let $\mathcal{L}_M$ be the set of $D \in \operatorname{Diag}(\mathbb{K}^{2r\times 2r}) \cong \mathbb{K}^{2r}$ such that some $(r+1)\times(r+1)$ minor of $M + D$ is singular. Then $\mathcal{L}_M$ is a proper Zariski-closed subset of $\operatorname{Diag}(\mathbb{K}^{2r\times 2r})$.*

Proof. Let X be an $(r+1)\times(r+1)$ minor of $M + D$ that involves k diagonal entries of D: $x_1, x_2, \dots, x_k$. Then $k > 0$. Moreover, $\det(X)$ is a polynomial over $\mathbb{K}$ in variables $\{x_i \mid i \in [k]\}$ and the coefficient in front of $x_1x_2\dots x_k$ is the determinant of a minor formed by rows and columns of X that have no diagonal entries of D. Since $M \in \mathcal{D}_r^{\#}$, this coefficient is non-zero. Hence, the set of D for which $\det(X) = 0$ is a proper Zariski-closed set in $\operatorname{Diag}(\mathbb{K}^{2r\times 2r})$. Since $\operatorname{Diag}(\mathbb{K}^{2r\times 2r}) \cong \mathbb{K}^{2r}$ is irreducible, the finite union (over all $(r+1)\times(r+1)$ minors) of proper Zariski-closed subsets is a proper Zariski-closed subset. ◀

3 A lower bound on the strict rigidity for a matrix in $\mathcal{C}_r$

In this section we prove the following stronger version of Theorem 7.

▶ **Theorem 33.** *Let $r \ge 3$. For almost all matrices $M \in \mathcal{D}_r$ there exists a diagonal matrix $D \in \operatorname{Diag}(\mathbb{K}^{2r\times 2r})$ such that $R_{\mathbb{K}}(M + D, r) \ge 3r - 2$.*

▶ **Definition 34** ($\mathbb{F}^{(\pi)}$). *For $\pi \subseteq [n] \times [m]$ denote by $\mathbb{F}^{(\pi)}$ the subset of matrices in $\mathbb{F}^{n\times m}$ with zero entries in every cell outside of π (we assume that n and m are clear from the context).*

Assume $R_{\mathbb{K}}(M + D, r) \le 3r - 3$ for all diagonal matrices D. Intuitively, since there are only finitely many subsets $\pi \subset [2r] \times [2r]$ of size $3r - 3$, there should exist π such that for a "large set" of diagonal matrices D there exists a corresponding $Z \in \mathbb{K}^{(\pi)}$ with $\operatorname{rank}(M + D + Z) \le r$. In Section 3.2, we are going to make this intuitive argument precise.

Then, in order to prove Theorem 33 it is sufficient to show that for an arbitrary fixed π of size $3r - 3$ for almost all matrices $M \in \mathcal{D}_r$ there is no "large set" of diagonal matrices D such that $\operatorname{rank}(M + D + Z) \le r$ for all D is this set and all $Z \in \mathbb{K}^{(\pi)}$.

3.1 Structure of the subsets of $[2r] \times [2r]$ with at most $3r - 3$ elements

We start the discussion towards the proof of Theorem 33 with the study of the structure of the subsets of $[2r] \times [2r]$ with at most $3r - 3$ elements.

▶ **Definition 35** (Well-distributed). *Let $m \geq r+1$. We say that $\pi \subseteq [2r] \times [m]$ is* well-distributed*, if every $(r+1) \times (r+1)$ minor contains at least one element of π.*

Note that if π is not well-distributed and all $(r+1) \times (r+1)$ minors of $A \in \mathbb{C}^{2r \times 2r}$ are non-singular, then for every $Z \in \mathbb{C}^{(\pi)}$ we have $\operatorname{rank}(A+Z) \geq r+1$. By Lemma 32, for all $M \in \mathcal{D}_r^{\#}$, for a Zariski-open (so, "large") set of diagonal matrices D all $(r+1) \times (r+1)$ minors of $M + D$ are non-singular.

Hence, we mainly need to concentrate on well-distributed sets π.

▶ **Observation 36.** *Let $\pi \subseteq [2r] \times [m]$ be well-distributed. Then for any set of $r+1$ columns, π contains elements in at least r distinct rows.*

Proof. If not, we immediately find an $(r+1) \times (r+1)$ minor with no elements from π. ◀

▶ **Lemma 37.** *Let $\pi \subseteq [2r] \times [2r]$ be well-distributed. For each $i \in [2r]$, let t_i be the number of elements of π in the i-th column. Let $t_{(j)}$ be the j-th smallest number among $\{t_i \mid i \in [2r]\}$.*

1. *Then, either $t_{(r+2)} = 1$, or $|\pi| \geq 3r - 2$.*
2. *If $t_{(1)} = 1$, then either $t_{(r+3)} = 1$, or $|\pi| \geq 3r - 2$.*

Proof. Assume $t_{(r+2)} \geq 2$. By Observation 36, $r+1$ columns that contain the least number of elements from π have at least r elements from π. The other $r-1$ columns contain at least $2(r-1)$ elements from π. Thus, in this case, $|\pi| \geq 3r - 2$.

Finally, if $t_{(1)} = t_{(r+2)} = 1$, but $t_{(r+3)} \geq 2$, then $|\pi| \geq 2(r-2) + r + 2 = 3r - 2$. ◀

▶ **Definition 38** (Matching). *We say that $\pi \subseteq [n] \times [m]$ is a* matching *if the projections of π on each of its two coordinates are injective.*

▶ **Lemma 39.** *Let $r \geq 3$. Assume that $\pi \subseteq [2r] \times [r+3]$ has precisely one element in every column and is well-distributed, then π contains a matching of size $r+2$.*

Proof. Note that $|\pi| = r + 3$ and Observation 36 implies that π has elements in at least r rows. Since $3r > r + 3$ for $r \geq 3$ there is at least one row with ≤ 2 elements. Considering $r+1$ columns that do not contain these elements, by Observation 36, we get that π has elements in at least $r+1$ distinct rows. Since $2r + 1 > r + 3$ for $r \geq 3$ there are at least two rows with precisely one element in each. We match each of these rows to the unique available column. Consider the set of the other $r+1$ columns. By Observation 36, there are at least r rows that have elements in these columns. Since every column has precisely 1 element, by picking one element in each row we will get a matching of size $r+2$. ◀

3.2 Reduction to a fixed well-distributed π

▶ **Definition 40** (Unbounded). *For a subfield $\mathbb{F} \subseteq \mathbb{C}$ we say that a set of points $\{x_i\}_{i \in I} \subseteq \mathbb{F}$ is unbounded, if it is unbounded as a set in $\mathbb{C}$.*

▶ **Definition 41** ($\mathcal{C}_{r,\pi}$, $\mathcal{C}'_{r,\pi}$). *For $\pi \subseteq [2r] \times [2r]$ define*

$$\begin{aligned} \mathcal{C}_{r,\pi} &= \{A \in \mathcal{C}_r \mid \exists Z \in \mathbb{C}^{(\pi)} : \operatorname{rank}(A+Z) \leq r\}, \text{ and} \\ \mathcal{C}'_{r,\pi} &= \{A \in \mathcal{C}_r \mid \exists Z \in \mathbb{K}^{(\pi)} : \operatorname{rank}(A+Z) \leq r\}. \end{aligned} \tag{8}$$

▶ **Lemma 42.** *Let $M \in \mathcal{D}_r$ and P be a finite collection of subsets of $[2r] \times [2r]$. Let Ω_M be a non-empty Zariski-open set in $\mathrm{Diag}(\mathbb{K}^{2r\times 2r})$. Assume that for every diagonal matrix $D \in \Omega_M$ we have $M + D \in \bigcup_{\pi \in P} \mathcal{C}_{r,\pi}$. Then there exist unbounded sets $E_1, E_2, \dots E_{2r} \subseteq \mathbb{K}$ and $\pi \in P$ such that for all diagonal D with $D_{ii} \in E_i$ for all $i \in [2r]$ we have $M + D \in \mathcal{C}_{r,\pi}$.*

Proof. For $\pi \in P$, consider the algebraic set

$$W_M(\pi) = \{(D, Z) \mid D \in \mathrm{Diag}(\mathbb{C}^{2r\times 2r}), Z \in \mathbb{C}^{(\pi)},\ \mathrm{rank}(M + D + Z) \le r\}.$$

Since $\mathrm{Diag}(\mathbb{C}^{2r\times 2r}) \cong \mathbb{C}^{2r}$, we can treat $W_M(\pi)$ as a subvariety of $\mathbb{C}^{2r} \times \mathbb{C}^{|\pi|}$. The projection on the first coordinate $p : (D, Z) \mapsto D$ is regular, so by Chevalley's Theorem (Theorem 87) the image $p(W_M(\pi))$ under this projection is a constructible set for every π. Since a constructible set is an intersection of a closed and an open set, for every π, either $p(W_M(\pi))$ is Zariski-open, or there exists a non-trivial polynomial f_π that completely vanishes on $p(W_M(\pi))$. If neither of $p(W_M(\pi))$ is Zariski-open, then there exists a nontrivial polynomial (e.g., $\prod_{\pi\in P} f_\pi$) that vanishes on $\bigcup_{\pi\in P} p(W_M(\pi))$, and so vanishes on Ω_M. This is a contradiction, as $\mathrm{Diag}(\mathbb{K}^{2r\times 2r}) \cong \mathbb{K}^{2r}$ is irreducible and Ω_M is non-empty Zariski-open.

Hence, there exists $\pi \in P$, such that $p(W_M(\pi))$ is Zariski-open in $\mathrm{Diag}(\mathbb{C}^{2r\times 2r})$, and so $\Omega'_M = \Omega_M \cap p(W_M(\pi))$ is Zariski-open in $\mathrm{Diag}(\mathbb{K}^{2r\times 2r})$. Hence, the claim of the lemma follows from Lemma 85. ◀

Note that in the definition of $\mathcal{C}_{r,\pi}$ we allow the entries of Z to be from $\mathbb{C}$ instead of $\mathbb{K}$. So if P contains a superset of $\{(i, i) \mid i \in [2r]\}$ the lemma above gives a trivial statement. Thus we shall consider two different regimes, when π is "close to containing the diagonal" and when it is not.

More precisely, as we saw in Lemma 37, there exists a subset $S' \subset [2r]$ of size $r + 2$ such that every column with index in S' has at most one element of π. We will discuss how to pick S' in Section 3.5, if for π this choice is not unique. We distinguish two cases: (a) when π restricted to columns in S' is a subset of the diagonal and (b) when it is not.

In the case (a) we will show that for almost all $M \in D_r^{\#}$ for all diagonal $D \in \mathrm{Diag}(\mathbb{K}^{2r\times 2r})$ we have $A = M + D \notin \mathcal{C}'_{r,\pi}$.

By applying Lemma 42 to the collection of all π from the case (b), we get that there exists a π from case (b) and a "large" set of diagonal matrices D such that $A = M + D \in \mathcal{C}_{r,\pi}$. We will argue that this does not happen for almost all $M \in D_r^{\#}$.

Both in case (a) and in case (b) we only study the matrix $B = \mathrm{proj}_{S'}(A)$ and show that for almost all M there is no change of entries inside π that allows to get a matrix of rank $\le r$ from B.

3.3 Case when π in columns S' coincides with the diagonal

In the next lemma we show that for almost all $M \in \mathcal{D}_r$ there is no $Z' \in \mathrm{Diag}(\mathbb{K}^{2r\times(r+2)})$ such that $\mathrm{rank}(\mathrm{proj}_{[r+2]}(M) + Z') \le r$.

In this and next section we use e_i to denote the vector in $\mathbb{K}^r$ with entry 1 in coordinate i and 0 in all other coordinates.

▶ **Lemma 43.** *Let $r \ge 3$. Consider $A_1 \in \mathbb{K}^{r\times r}$ and an invertible matrix $A_2 \in \mathbb{K}^{r\times r}$. Define*

$$v_i = -A_1^2 A_2^{-1} e_i + \omega^2 A_2^{-1} e_i \quad \textit{and} \quad w_i = -A_2 A_1 A_2^{-1} e_i.$$

For a diagonal matrix $Z \in \mathbb{K}^{r\times r}$ and $z_1, z_2 \in \mathbb{K}$ consider

$$T(Z, z_1, z_2) = \begin{pmatrix} A_1 + Z & v_1 & v_2 \\ A_2 & w_1 + z_1 e_1 & w_2 + z_2 e_2 \end{pmatrix}.$$

The set of matrices $(A_1, A_2) \in \mathbb{K}^{2r^2}$ *for which there exist* $Z \in \mathrm{Diag}(\mathbb{K}^{r\times r})$, $z_1, z_2 \in \mathbb{K}$ *s.t.* $\mathrm{rank}(T(Z, z_1, z_2)) \le r$ *is small in* $\mathbb{K}^{2r^2}$*.*

Proof. Assume $\mathrm{rank}(T(Z, z_1, z_2)) \le r$. Since A_2 is invertible, the last two columns of $T(Z, z_1, z_2)$ can be expressed as a linear combination of the first r columns. Let $y_i \in \mathbb{K}^r$ satisfy

$$\begin{pmatrix} A_1 + Z \\ A_2 \end{pmatrix} y_i = \begin{pmatrix} v_i \\ w_i + z_i e_i \end{pmatrix}.$$

Then

$$y_i = A_2^{-1}(w_i + z_i e_i) \quad \Rightarrow \quad (A_1 + Z)A_2^{-1}(w_i + z_i e_i) = v_i,$$
$$-A_1^2 A_2^{-1} e_i + z_i A_1 A_2^{-1} e_i + Z(-A_1 A_2^{-1} e_i + z_i A_2^{-1} e_i) = -A_1^2 A_2^{-1} e_i + \omega^2 A_2^{-1} e_i.$$

Let $\alpha_i = A_1 A_2^{-1} e_i$ and $\beta_i = A_2^{-1} e_i$. Then for all $k \in [r]$ we have

$$Z_{kk} = \frac{\omega^2 \beta_{ik} - z_i \alpha_{ik}}{-\alpha_{ik} + z_i \beta_{ik}}.$$

Hence, for all $k \in [r]$,

$$\frac{\omega^2 \beta_{1k} - z_1 \alpha_{1k}}{-\alpha_{1k} + z_1 \beta_{1k}} = \frac{\omega^2 \beta_{2k} - z_2 \alpha_{2k}}{-\alpha_{2k} + z_2 \beta_{2k}}. \tag{9}$$

This can be rewritten as

$$\alpha_{2k} \left(\frac{\omega^2 \beta_{1k} - z_1 \alpha_{1k}}{-\alpha_{1k} + z_1 \beta_{1k}} - z_2 \right) = z_2 \beta_{2k} \left(\frac{\omega^2 \beta_{1k} - z_1 \alpha_{1k}}{-\alpha_{1k} + z_1 \beta_{1k}} \right) - \omega^2 \beta_{2k}. \tag{10}$$

The coefficient in front of α_{2k} is 0 if and only if

$$\omega^2 \beta_{1k} - z_1 \alpha_{1k} = -z_2 \alpha_{1k} + z_2 z_1 \beta_{1k} \quad \Leftrightarrow \quad \beta_{1k} = \frac{z_1 - z_2}{\omega^2 - z_1 z_2} \alpha_{1k}.$$

Unless $z_1 = z_2 = \pm\omega$, such equation can hold for at most one index k, or the set $(A_1, A_2) \in \mathbb{K}^{2r^2}$ is small. If the coefficient in front of α_{2k} is non-zero for some k, then α_{2k} can be expressed as a rational function of α_{1k}, β_{1k}, β_{2k} and z_1, z_2. Hence, for every k either α_{2k} is a function of α_{1k}, β_{1k}, β_{2k} and two parameters z_1, z_2, or β_{1k} is a function of α_{1k} and z_1, z_2. In any case, for $r \ge 3$ we see that the set $(A_1, A_2) \in \mathbb{K}^{2r^2}$ that satisfy Eq. (9) is small. ◀

3.4 Case when π in columns S' does not coincide with the diagonal

In the lemmas below we think of T as of a matrix obtained by permuting rows and columns of $\mathrm{proj}_{S'}(M + D + Z')$ for $M \in \mathcal{D}_r^{\#}$, $D \in \mathrm{Diag}(\mathbb{K}^{2r\times 2r})$ and $Z' \in \mathbb{C}^{\pi}$. The variables x_i correspond to selected diagonal entries of D and the variables Z, z_i correspond to entries of Z'.

Let $\widehat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ be the Riemann sphere, i. e., the one-point compactification of $\mathbb{C}$ with respect to the usual complex norm.

▶ **Observation 44.** *Let* $f : \widehat{\mathbb{C}} \to \widehat{\mathbb{C}}$ *be defined as* $f(x) = \dfrac{ax + b}{cx + d}$ *for* $a, b, c, d \in \mathbb{C}$*. If* $\{f(x_k)\}_{k=1}^{\infty}$ *converges to* y *in* $\widehat{\mathbb{C}}$*, then there exists* $x \in \widehat{\mathbb{C}}$ *such that* $f(x) = y$*.*

Proof. If $ad - bc \ne 0$, take x to be the limit of $\{x_k\}_{k=1}^{\infty}$ in $\widehat{\mathbb{C}}$. Else, pick an arbitrary $x \in \widehat{\mathbb{C}}$. ◀

▶ **Lemma 45.** *Let $r \geq 3$. Let $j_1 \notin \{1,2\}$ and $j_2 \notin \{2, j_1\}$ be elements of $[r]$. Let $E_1, E_2 \subseteq \mathbb{K}$ be unbounded sets. For $v_1, v_2, w_1, w_2 \in \mathbb{K}^r$, $A_1 \in \mathbb{K}^{r\times r}$, an invertible matrix $A_2 \in \mathbb{K}^{r\times r}$, $x_1, x_2 \in \mathbb{K}$, $z_1, z_2 \in \mathbb{C}$ and a diagonal matrix $Z \in \mathbb{C}^{r\times r}$ consider*

$$T(x_1, x_2, Z, z_1, z_2) = \begin{pmatrix} A_1 + Z & v_1 & v_2 \\ A_2 & w_1 + z_1e_1 + x_1e_{j_1} & w_2 + z_2e_2 + x_2e_{j_2} \end{pmatrix}.$$

The set of $(A_1, A_2) \in \mathbb{K}^{2r^2}$, for which there exist v_1, v_2, w_1, w_2, s.t. for all $x_1 \in E_1$, $x_2 \in E_2$ there exist $Z \in \mathrm{Diag}(\mathbb{C}^{r\times r})$, $z_1, z_2 \in \mathbb{C}$ s.t. $\mathrm{rank}(T(x_1, x_2, Z, z_1, z_2)) \leq r$, is small in $\mathbb{K}^{2r^2}$.

Proof. Since A_2 is invertible and the rank of $T(x_1, x_2, Z, z_1, z_2)$ is $\leq r$, $\forall i \in \{1,2\}$ we have

$$(A_1 + Z)A_2^{-1}(w_i + z_ie_i + x_ie_{j_i}) = v_i,$$

$$A_1A_2^{-1}(z_ie_i + x_ie_{j_i}) + ZA_2^{-1}(w_i + z_ie_i + x_ie_{j_i}) = v_i - A_1A_2^{-1}w_i.$$

Denote $\gamma_i = v_i - A_1A_2^{-1}w_i$, $\alpha_i = A_1A_2^{-1}e_i$, $\beta_i = A_2^{-1}e_i$ and $\phi_i = A_2^{-1}w_i$. Then

$$Z_{kk} = \frac{\gamma_{ik} - z_i\alpha_{ik} - x_i\alpha_{j_ik}}{\phi_{ik} + z_i\beta_{ik} + x_i\beta_{j_ik}}, \qquad \forall k \in [r], \forall i \in \{1,2\}, \quad \text{so} \tag{11}$$

$$\frac{\gamma_{1k} - z_1\alpha_{1k} - x_1\alpha_{j_1k}}{\phi_{1k} + z_1\beta_{1k} + x_1\beta_{j_1k}} = \frac{\gamma_{2k} - z_2\alpha_{2k} - x_2\alpha_{j_2k}}{\phi_{2k} + z_2\beta_{2k} + x_2\beta_{j_2k}} \qquad \forall k \in [r]. \tag{12}$$

Fix $x_2 \in E_2$. By passing to a subsequence for $x_1 \in E_1$ we may assume that $\lim\limits_{E_1 \ni x_1 \to \infty} z_1(x_1, x_2)/x_1 = c \in \widehat{\mathbb{C}}$ is well-defined. For this subsequence,

$$\lim_{E_1 \ni x_1 \to \infty} \frac{\gamma_{2k} - z_2\alpha_{2k} - x_2\alpha_{j_2k}}{\phi_{2k} + z_2\beta_{2k} + x_2\beta_{j_2k}} = -\frac{\alpha_{1k}c + \alpha_{j_1k}}{\beta_{1k}c + \beta_{j_1k}} \qquad \forall k \in [r].$$

Hence, using Observation 44, there exists $z_2 = z_2(x_2)$ such that

$$\frac{\gamma_{2k} - z_2\alpha_{2k} - x_2\alpha_{j_2k}}{\phi_{2k} + z_2\beta_{2k} + x_2\beta_{j_2k}} = -\frac{\alpha_{1k}c + \alpha_{j_1k}}{\beta_{1k}c + \beta_{j_1k}} \qquad \forall k \in [r].$$

By passing to a subsequence for $x_2 \in E_2$ we may assume that $\lim\limits_{E_2 \ni x_2 \to \infty} z_2(x_2)/x_2 = c' \in \widehat{\mathbb{C}}$. Then

$$\lim_{E_2 \ni x_2 \to \infty} \frac{\alpha_{1k}c(x_2) + \alpha_{j_1k}}{\beta_{1k}c(x_2) + \beta_{j_1k}} = \frac{c'\alpha_{2k} + \alpha_{j_2k}}{c'\beta_{2k} + \beta_{j_2k}} \qquad \forall k \in [r].$$

Hence, using Observation 44, there exists c'' such that

$$\frac{\alpha_{1k}c'' + \alpha_{j_1k}}{\beta_{1k}c'' + \beta_{j_1k}} = \frac{c'\alpha_{2k} + \alpha_{j_2k}}{c'\beta_{2k} + \beta_{j_2k}} \qquad \forall k \in [r]. \tag{13}$$

Since $j_2 \neq 2$, this gives a dependence for α_{j_2k} on other variables with last index k and 2 parameters c', c'', if $c' \neq \infty$. If $c' = \infty$, we get a dependence for α_{2k} on other variables with last index k and a parameter c''. Hence for $r \geq 3$ the set of matrices $(A_1, A_2) \in \mathbb{K}^{2r^2}$ that satisfy Eq. (13) is small in $\mathbb{K}^{2r^2}$. ◀

The next two lemmas can be proved in a similar fashion, so we defer their proofs to Appendix B.

▶ **Lemma 46.** *Let $r \geq 3$. Consider $A_1 \in \mathbb{K}^{r\times r}$ and an invertible matrix $A_2 \in \mathbb{K}^{r\times r}$. Define*

$$v_i = -A_1^2 A_2^{-1} e_i + \omega^2 A_2^{-1} e_i \quad \text{and} \quad w_i = -A_2 A_1 A_2^{-1} e_i$$

Let $E_2 \subseteq \mathbb{K}$ be an unbounded set. For a diagonal matrix $Z \in \mathbb{C}^{r\times r}$ and $z_1, z_2 \in \mathbb{C}$, $x_2 \in \mathbb{K}$, consider

$$T(x_2, Z, z_1, z_2) = \begin{pmatrix} A_1 + Z & v_1 & v_2 \\ A_2 & w_1 + z_1 e_1 & w_2 + z_2 e_3 + x_2 e_2 \end{pmatrix}.$$

The set of matrices $(A_1, A_2) \in \mathbb{K}^{2r^2}$, such that for all $x_2 \in E_2$ there exist $Z \in \operatorname{Diag}(\mathbb{C}^{r\times r})$, and $z_1, z_2 \in \mathbb{C}$ such that $\operatorname{rank}(T(x_2, Z, z_1, z_2)) \leq r$, is small in $\mathbb{K}^{2r^2}$.

Proof. See Appendix B, Lemma 88. ◀

▶ **Lemma 47.** *Let $r \geq 3$. Let $j_1 \notin \{1, 2\}$ be an element of $[r]$. Let $E_1, E_2 \subseteq \mathbb{K}$ be unbounded sets. For $v_1, v_2, w_1, w_2 \in \mathbb{K}^r$, $A_1 \in \mathbb{K}^{r\times r}$, an invertible matrix $A_2 \in \mathbb{K}^{r\times r}$, $x_1, x_2 \in \mathbb{K}$, $z_1, z_2 \in \mathbb{C}$ and a diagonal matrix $Z \in \mathbb{C}^{r\times r}$ consider*

$$T(x_1, x_2, Z, z_1, z_2) = \begin{pmatrix} A_1 + Z & v_1 & v_2 + x_2 e_1 \\ A_2 & w_1 + z_1 e_1 + x_1 e_{j_1} & w_2 + z_2 e_2 \end{pmatrix}.$$

The set of matrices $(A_1, A_2) \in \mathbb{K}^{2r^2}$, for which there exist $v_1, v_2, w_1, w_2 \in \mathbb{K}^r$, s.t. for all $x_1 \in E_1$ and $x_2 \in E_2$ there exist $Z \in \operatorname{Diag}(\mathbb{C}^{r\times r})$, $z_1, z_2 \in \mathbb{C}$ s.t. $\operatorname{rank}(T(x_1, x_2, Z, z_1, z_2)) \leq r$, is small in $\mathbb{K}^{2r^2}$.

Proof. See Appendix B, Lemma 89. ◀

▶ Remark 48. Note that in Lemmas 43 and 46 we assume that v_1, v_2, w_1 and w_2 have the specific form given by Lemma 24, while in Lemmas 45 and 47 we cannot make such assumption. The reason is that Lemmas 45 and 47 treat matrices obtained from $M \in \mathcal{D}_r$ after its rows and columns are permuted in the way that does not respect the diagonal. And so, in this case, Lemma 24 cannot be applied.

3.5 Proof of Theorem 33

Finally, we are ready to prove Theorem 33.

Proof of Theorem 33. Let $\mathcal{P}$ denote the collection of the subsets of $[2r] \times [2r]$ with precisely $3r - 3$ elements. Let $\mathcal{P}_0 \subset \mathcal{P}$ denote the set of well-distributed $\pi \in \mathcal{P}$.

Recall that $\mathcal{D}_r^{\#}$ denotes the set of matrices $M \in \mathcal{D}_r^*$ such that for all $k \leq r$ every $k \times k$ minor of M is non-singular.

Fix $M \in \mathcal{D}_r^{\#}$. Assume that for every $D \in \operatorname{Diag}(\mathbb{K}^{2r\times 2r})$ we have $R_{\mathbb{K}}(M + D, r) \leq 3r - 3$. This means that for every $D \in \operatorname{Diag}(\mathbb{K}^{2r\times 2r})$ there exists $\pi \in \mathcal{P}$ and $Z \in \mathbb{K}^{(\pi)}$ such that

$$\operatorname{rank}(M + D + Z) \leq r.$$

Let $\mathcal{L}_M$ be the set of $D \in \operatorname{Diag}(\mathbb{K}^{2r\times 2r}) \cong \mathbb{K}^{2r}$ such that some $(r+1) \times (r+1)$ minor of $M + D$ is singular. By Lemma 32, $\mathcal{L}_M$ is a proper Zariski-closed subset of $\operatorname{Diag}(\mathbb{K}^{2r\times 2r})$. Define $\Omega_M = \operatorname{Diag}(\mathbb{K}^{2r\times 2r}) \setminus \mathcal{L}_M$.

Observe that for all $D \in \Omega_M$, for all $\pi \in \mathcal{P} \setminus \mathcal{P}_0$ and for all $Z \in \mathbb{C}^{(\pi)}$ we have

$$\operatorname{rank}(M + D + Z) \geq r + 1.$$

From now on we restrict ourself to taking $D \in \Omega_M$. Hence, in the rest of the proof we may assume that π is well-distributed.

Let $S \subseteq [2r]$ denote the set of indices of columns that have at most 1 element of π. Then, by Lemma 37, $|S| \geq r+2$, and if there is no column with 0 elements, then $|S| \geq r+3$.

Now we want to pick a subset S' of $r+2$ indices from S. We use the following rules.

1. If there is a column with index in S that contains 0 elements of π, select S' to be an arbitrary subset of S of size $r+2$ that contains this index.
2. Otherwise, every column with index in S has precisely 1 element of π and $|S| \geq r+3$.
   a. If $\pi = \{(i,i) \mid i \in S\}$ pick S' to be an arbitrary subset of S of size $r+2$.
   b. If π disagrees with the diagonal in precisely 1 position, choose S' to consist of columns where π agrees with the diagonal.
   c. If π disagrees with the diagonal in precisely 2 positions, choose S' to contain only one column where they disagree. Moreover, using Lemma 39, we can pick such S' so that π restricted to columns in S' defines a matching.
   d. Else, π has at least 3 elements not on the diagonal. We claim that it is always possible to pick S' so that
      - π defines a matching, when it is restricted to the columns in S'.
      - π disagrees with the diagonal in at least 2 positions when it is restricted to the columns with indices in S'.
      - π contains an element with column index in S' and row index not in S'.

      To justify that, first shrink S to be of size $r+3$ by preserving the condition that π disagrees with the diagonal in at least 3 columns. If π is a matching on S, choose any $(i,j) \in \pi$ with $i \neq j$ and define $S' = S \setminus \{i\}$. Otherwise, by Lemma 39, π contains a matching of size $r+2$, so there is precisely one row i with 2 elements in columns j_1 and j_2. Moreover, there is $j \in S$ such that the row with index j has no element of π. To get S' delete from S any of the elements j_1, j_2 that is different from j. Such S' satisfies all the desired properties.

Let $\pi' \subseteq [2r] \times S'$ denote the restriction of π to the columns in S' and define a matrix $B = \mathrm{proj}_{S'}(M)$. If there is a column of B with no element of π' we add an element to π' in this column to the row that has no element of π', and if possible, with an index not in S'. Thus, we may assume that every column of B contains precisely one element of π', and π' defines a matching.

By permuting the rows and columns of M in a way that preserves the diagonal, we may assume that $S' = [r+2]$. We also assume that coordinates in Ω_M are permuted accordingly.

We want to show that for almost all M and all $\pi \in \mathcal{P}_0$ there is no "large set" (in the sense of Lemma 42) of diagonal matrices $D \in \mathrm{Diag}(\mathbb{K}^{2r\times 2r})$ such that for arbitrary D in this set $\mathrm{rank}(M+D+Z) \leq r$ for some $Z \in \mathbb{K}^{(\pi)}$.

To do this, we show how to permute rows and columns of B in order to apply one of the lemmas proved in Sections 3.3 and 3.4. We have three cases for π'.

(A) If π' coincides with the diagonal, then by Lemma 43 for almost all matrices $M \in \mathcal{D}_r^{\#}$ there is no diagonal matrix $Y \in \mathrm{Diag}(\mathbb{K}^{2r\times(r+2)})$ such that $B+Y$ has rank at most r.

(B) If π' disagrees with the diagonal in precisely one column, then by the choice of S', π' has an element in a row that is not in $S' = [r+2]$. We may permute the rows and the columns of B, so that the diagonal is preserved and $\pi' = \{(i,i) \mid i \in [r+1]\} \cup \{(r+3, r+2)\}$. Then it follows from Lemma 46 that for almost all matrices $M \in \mathcal{D}_r^{\#}$ there are no unbounded sets $E_1, E_2, \dots, E_{r+2}$ such that for every matrix $D \in \mathrm{Diag}(\mathbb{C}^{2r\times r})$ with $D_{ii} \in E_i$ for $i \in [r+2]$ there exists $Y \in \mathbb{C}^{(\pi')}$ for which $B+D+Y$ has rank at most r.

(C) If π' disagrees with the diagonal in at least 2 columns, then, by the choice of S', there is a row with index $j_1 > r+2$ that has an element of π' in the column $i_1 \in S' = [r+2]$. Moreover, there is at least one other column i_2 with an element of π not on a diagonal. Since π' defines a matching, we can permute the rows and the columns of B so that π' becomes the diagonal and columns i_1, i_2 are mapped to columns $r+1$ and $r+2$. Let $\sigma \subseteq [2r] \times [r+2]$ be the the image of the diagonal after such permutation. By the construction of π', σ has the entry in column $r+1$ in the row with index $\geq r+3$ and the entry in column $r+2$ in the row distinct from $r+2$. If the entry of σ in the last column is in the row with index $\leq r$ we can further permute the first r columns and rows in the way that preserves the diagonal, so that the entry of σ in the last column becomes in the first row.

Let $E_1, E_2, \dots, E_{r+2}$ be unbounded subsets of $\mathbb{K}$ (which may depend on M) and let $\sigma' = \sigma \setminus \{(i,i) \mid i \in [r]\}$. Let $D \in \mathbb{K}^{(\sigma)}$ be such that $D_{ij} \in E_j$ for all $(i,j) \in \sigma$ and let D' be a part of D supported on σ'. Then, by Lemma 45 and Lemma 47, if for every D there exists $Y \in \operatorname{Diag}(\mathbb{C}^{2r\times(r+2)})$ such that $B + D + Y$ has rank at most r, then for every D, $\operatorname{proj}_{[r]}(B) + D'$ belongs to a proper Zariski-closed subset $\mathcal{B}_{\pi'}$ of $\mathbb{K}^{2r\times r}$, which depends only on π'.

This means that there exists a non-trivial polynomial $f \in \mathbb{K}[x_{ij}]_{i\in[2r],\ j\in[r]}$ such that $f(\operatorname{proj}_{[r]}(B) + D') = 0$. Consider this as a polynomial with variables d_{ij}, which are the (i,j)-th entries of D' with $(i,j) \in \sigma'$. Since every variable d_{ij} independently can take infinitely many values we get that this is a trivial polynomial in variables d_{ij}. Since f is non-trivial, we get that entries of B satisfy some non-trivial polynomial.

Therefore, for almost all $M \in \mathcal{D}_r^{\#}$ there are no unbounded sets $E_1, E_2, \dots, E_{r+2}$ such that for every matrix $D \in \operatorname{Diag}(\mathbb{C}^{2r\times r})$ with $D_{ii} \in E_i$ for $i \in [r+2]$ there exists $Y \in \mathbb{C}^{(\pi')}$ for which $B + D + Y$ has rank at most r.

We see from (A), that there is a set $\mathcal{M}$ of almost all matrices $M \in \mathcal{D}_r^{\#}$, such that for all $D \in \Omega_M$ and all well-distributed π, for which π' coincides with the diagonal, there is no $Z \in \mathbb{K}^{(\pi)}$ with $\operatorname{rank}(M + D + Z) \leq r$.

Let $\mathcal{P}_1 \subseteq \mathcal{P}_0$ be the set of well-distributed $\pi \subseteq [2r] \times [2r]$ for which the π', constructed by the rules above, does not end up in case (A), i.e. π' does not coincide with the diagonal.

Assume that $M \in \mathcal{M}$. Then for any $D \in \Omega_M$ there should exists a $\pi \in \mathcal{P}_1$ and $Z \in \mathbb{K}^{(\pi)}$ such that $\operatorname{rank}(M + D + Z) \leq r$. Using Lemma 42, applied with $P = \mathcal{P}_1$, we deduce that there exists a set $\pi \in \mathcal{P}_1$ and unbounded sets $E_1, E_2, \dots, E_{2r} \subseteq \mathbb{K}$, such that for any $D \in \operatorname{Diag}(\mathbb{K}^{2r\times 2r})$ with $D_{ii} \in E_i$ for every $i \in [2r]$, there exists $Z \in \mathbb{C}^{(\pi)}$ with $\operatorname{rank}(M+D+Z) \leq r$. Let S' be as above and $B = \operatorname{proj}_{S'}(M)$. Then $\operatorname{rank}(B + \operatorname{proj}_{S'}(D) + \operatorname{proj}_{S'}(Z)) \leq r$. However, for almost all matrices M this gives a contradiction with (B) or (C).

Thus, for almost all $M \in \mathcal{D}_r^{\#}$ there is $D \in \operatorname{Diag}(\mathbb{K}^{2r\times 2r})$ with $R_{\mathbb{K}}(M + D, r) \geq 3r - 2$. ◀

4 Field extension: avoiding transcendentals

In this section we prove that absolute rigidity can always be achieved over a finite extension. Recall that a field extension $\mathbb{L}/\mathbb{K}$ is *finite* if $\dim_{\mathbb{K}} \mathbb{L}$ is finite. Recall also that we wrote $R^*(A, r)$ to denote the absolute rigidity of A for target rank r.

▶ **Proposition 49.** *Let A be a matrix over the field $\mathbb{K}$. Then there exists a finite extension $\mathbb{L}/\mathbb{K}$ such that for all $r \geq 0$ we have $R^*(A, r) = R_{\mathbb{L}}(A, r)$.*

▶ **Notation 50** (weight). For a matrix A, let $w(A)$, the *weight* of A, denote the number of nonzero entries of A.

We begin with some simple observations.

▶ **Observation 51.** *If $\mathbb{L}/\mathbb{K}$ is a field extension and A is a matrix over $\mathbb{K}$, then, for all $r \geq 0$, we have $R_{\mathbb{K}}(A, r) \geq R_{\mathbb{L}}(A, r)$.* ◀

▶ **Definition 52.** *For a field $\mathbb{K}$ let $\mathrm{cl}(\mathbb{K})$ denote the algebraic closure of the pure transcendental extension of $\mathbb{K}$ of countably infinite transcendence degree.*

▶ **Observation 53.** *Let A be a matrix over the field $\mathbb{K}$. Then for all $r \geq 0$, we have $R^*(A, r) = R_{\mathrm{cl}(\mathbb{K})}(A, r)$.*

Proof. Fix r. By definition, $R^*(A, r) \leq R_{\mathrm{cl}(\mathbb{K})}(A, r)$. We need to prove the reverse inequality.

Let $\mathbb{L}$ be an extension of $\mathbb{K}$ such that $R^*(A, r) = R_{\mathbb{L}}(A, r)$. So there exists a matrix D over $\mathbb{L}$, of weight $R^*(A, r)$, such that $\mathrm{rank}(A - D) \leq r$. Let $\mathbb{M} \subseteq \mathbb{L}$ be the subfield generated by $\mathbb{K}$ and the elements of D. Then $R_{\mathbb{L}}(A, r) = R_{\mathbb{M}}(A, r)$. But $\mathbb{M}$ can be embedded in $\mathrm{cl}(\mathbb{K})$ and therefore, by Obs. 51, $R_{\mathbb{M}}(A, r) \geq R_{\mathrm{cl}(\mathbb{K})}(A, r)$. So $R^*(A, r) = R_{\mathbb{L}}(A, r) = R_{\mathbb{M}}(A, r) \geq R_{\mathrm{cl}(\mathbb{K})}(A, r)$. ◀

We shall need the following well-known result, which is often the first step in the proof of Hilbert's Nullstellensatz. See, e. g., Cor. 1.2 in Chap. 9, §1 of [9].

▶ **Fact 54.** Let $\mathbb{L}/\mathbb{K}$ be a field extension. Assume $\mathbb{L}$ is a finitely generated $\mathbb{K}$-algebra. Then the extension $\mathbb{L}/\mathbb{K}$ is finite.

Proof of Proposition 49. We need to achieve $R_{\mathbb{L}}(A, r) = R_{\mathrm{cl}(\mathbb{K})}(A, r)$ for all r. For each r we have a matrix Z_r over $\mathrm{cl}(\mathbb{K})$ of weight $\leq R^*(A, r)$ such that $\mathrm{rank}_{\mathrm{cl}(\mathbb{K})}(A - Z_r) \leq r$. Let $\mathcal{Z}$ denote the set of elements of the matrices Z_r, $r \geq 0$. This is a finite set. (If $r \geq \mathrm{rk}(A)$ then $Z_r = 0$.) Let $\mathcal{B} = \mathbb{K}[\mathcal{Z}]$ denote the $\mathbb{K}$-algebra generated by $\mathcal{Z}$. Let $\mathcal{M}$ be a maximal ideal of $\mathcal{B}$, and let $\mathbb{L} = \mathcal{B}/\mathcal{M}$. So $\mathbb{L}$ is an extension field of $\mathbb{K}$.

Let $\varphi : \mathcal{B} \to \mathbb{L}$ denote the natural epimorphism. So φ fixes all elements of $\mathbb{K}$. Moreover, for every matrix B we have $w(\varphi(B)) \leq w(B)$ and $\mathrm{rank}(\varphi(B)) \leq \mathrm{rank}(B)$ (because singular minors are mapped to singular minors). Therefore $\mathrm{rank}(A - \varphi(Z_r)) = \mathrm{rank}\,\varphi(A - Z_r) \leq \mathrm{rank}(A - Z_r) \leq r$, and $w(\varphi(Z_r)) \leq w(Z_r)$. This proves that $R_{\mathbb{L}}(A, r) \leq R^*(A, r)$. The reverse inequality holds by definition.

Finally we need to show that the extension $\mathbb{L}/\mathbb{K}$ is finite. This is immediate from Fact 54, given that $\mathbb{L} = \mathbb{K}[\varphi(\mathcal{Z})]$ is a finitely generated $\mathbb{K}$-algebra which is a field. ◀

We observe that Prop. 49 is equivalent to saying that absolute rigidity is achieved over the algebraic closure of the field of definition of the matrix.

▶ **Corollary 55.** *Let A be a matrix over the field $\mathbb{K}$. Then $R^*(A, r) = R_{\overline{\mathbb{K}}}(A, r)$, where $\overline{\mathbb{K}}$ denotes the algebraic closure of $\mathbb{K}$. Moreover, this statement is equivalent to Prop. 49.*

Proof. Assume Prop. 49. Let $\mathbb{L}$ be a finite extension of $\mathbb{K}$ such that $R^*(A, r) = R_{\mathbb{L}}(A, r)$. Then $\mathbb{L}$ can be embedded in $\overline{\mathbb{K}}$, so a reference to Obs. 51 proves the Corollary.

Now suppose the Corollary is true. For every r, let B_r be the matrix over $\overline{\mathbb{K}}$ such that $\mathrm{rank}(B_r) \leq r$ and $w(A - B_r) = R^*(A, r)$. Let $S \subset \overline{\mathbb{K}}$ be the (finite) set of elements of the matrices B_r. Then, for all r, we have $R^*(A, r) = R_{\mathbb{K}[S]}(A, r)$. But $\mathbb{K}[S]$ is a finite extension, proving Prop. 49. ◀

A similar result holds for linear arithmetic circuits.

▶ **Proposition 56.** *Let $\mathbb{E}/\mathbb{K}$ be a field extension. Let $\mathcal{A}$ be a linear arithmetic circuit over the field $\mathbb{E}$ that computes a linear function $x \mapsto Ax$ over $\mathbb{K}$ (so A is a matrix over $\mathbb{K}$). Then $\mathcal{A}$ can be simulated by a linear arithmetic circuit $\mathcal{A}'$ over a finite extension of $\mathbb{K}$ such that $\mathcal{A}'$ has the same set of nodes and wires as $\mathcal{A}$.*

Proof. Each node of $\mathcal{A}$ computes an $\mathbb{E}$-linear combination of its inputs. Let $\mathcal{Z}$ denote the set of all the coefficients occurring at nodes. Let $\mathcal{B} = \mathbb{K}[\mathcal{Z}]$ denote the $\mathbb{K}$-algebra generated by $\mathcal{Z}$. Let $\mathcal{M}$ be a maximal ideal of $\mathcal{B}$, and let $\mathbb{L} = \mathcal{B}/\mathcal{M}$. So $\mathbb{L}$ is an extension field of $\mathbb{K}$. We shall define the linear arithmetic circuit $\mathcal{A}'$ over $\mathbb{L}$.

Let $\varphi : \mathcal{B} \to \mathbb{L}$ denote the natural epimorphism. So φ fixes all elements of $\mathbb{K}$. Now keep all nodes and links in $\mathcal{A}$ but replace each scalar $a \in \mathcal{Z}$ involved in $\mathcal{A}$ (as a coefficient of a linear combination at a gate) by $\varphi(a)$. So this circuit will compute the transformation $x \mapsto \varphi(A)x$. But $\varphi(A) = A$ (since φ fixes $\mathbb{K}$ pointwise), so the simulation is complete.

Finally, as before, the extension $\mathbb{L}/\mathbb{K} = \mathbb{K}[\varphi(\mathcal{Z})]/\mathbb{K}$ is finite by Fact 54. ◀

5 Refutation of more candidates for rigidity

In this section we show that, as corollaries to the results of Dvir and Liu [4], more long-running candidates for rigidity fail.

▶ **Definition 57** (G-circulants)**.** *Let G be a finite abelian group of order n, and let $A = (a_{ij})$ be an $n \times n$ matrix over a domain D. Let the rows and columns of A be labeled by the elements of G. We say that A is a G-circulant if there is a function $f : G \to D$ such that for all $i, j \in G$ we have $a_{ij} = f(i - j)$. A* circulant *matrix is a G-circulant where G is the cyclic group of order n.*

Recall that by a *family* of square matrices we mean a set of square matrices of unbounded order.

▶ **Theorem 58** (Dvir–Liu)**.**

(a) *No family of G-circulants over $\mathbb{C}$ (for variable G) is Valiant-rigid over $\mathbb{C}$.*

(b) *No family of circulants over a fixed finite field is strictly Valiant-rigid.*

Part (a) is stated in [4, Theorem 1.5]. Part (b) is stated in [4, Theorem 7.27].

5.1 Point–hyperplane incidence matrices

Finite projective geometries of dimension d are defined by geometric axioms. "Desargues' Theorem" is not one of the axioms; geometries satisfying this additional axiom are called Desarguesian. The Desarguesian finite projective geometries are precisely the *Galois geometries* $\mathrm{PG}(d, q)$ constructed from finite fields (q is the order of the field).

In fact, for $d \geq 3$, all projective spaces are Desarguesian. However, this is not the case for $d = 2$ (finite projective planes), so we need to make this distinction. Here we are interested only in Galois geometries.

Let q be a prime power and $d \geq 2$. The points as well as the hyperplanes of the d-dimensional Galois geometry $\mathrm{PG}(d, q)$ can be represented by equivalence classes of nonzero vectors in $\mathbb{F}_q^{d+1}$, where the equivalence relation is defined by scaling (one vector is a scalar multiple of the other). In particular, there are $N := (q^{d+1} - 1)/(q - 1)$ points and the same number of hyperplanes in this geometry. Let a be a point represented by a vector $x \in \mathbb{F}_q^{d+1} \setminus \{0\}$ and let b be a hypeplane represented by a vector $y \in \mathbb{F}_q^{d+1} \setminus \{0\}$. Then a and b are incident if and only if $x^T y = 0$ (x and y are "orthogonal"). (We view x, y as column vectors.)

The incidence matrix of this geometry is the $N \times N$ $(0,1)$ matrix of which the rows are labeled by the points, the columns are labeled by the hyperplanes, and an entry of 1 represents incidence.

▶ **Lemma 59.** *Let q be a prime power and $d \geq 2$. Under appropriate numbering of the points and hyperplanes, the point–hyperplane incidence matrix of the Galois geometry* $\mathrm{PG}(d,q)$ *is a circulant matrix.*

Proof. This is a consequence of the existence of a *Singer cycle* in $\mathrm{GL}(d+1,q)$, i. e., a linear transformation σ of $\mathbb{F}_q^{d+1}$ that cyclically permutes the nonzero vectors. The existence of such a transformation follows from the fact that the muliplicative group of $\mathbb{F}_{q^{d+1}}$ is cyclic: View $\mathbb{F}_q^{d+1}$ as the additive group of $\mathbb{F}_{q^{d+1}}$ and let σ be the multiplication by a generator of the multiplicative group of $\mathbb{F}_{q^{d+1}}$; this is a linear transformation of $\mathbb{F}_q^{d+1}$.

Any linear transformation of $\mathbb{F}_q^{d+1}$ preserves the "scaling" equivalence relation, so σ also gives a cyclic permutation of the points and the hyperplanes.

Let $A \in GL(d+1,q)$ be an invertible matrix and let let B denote its inverse-transpose. Then, for any $x, y \in \mathbb{F}_q^{d+1}$ we have $x^T y = 0$ if and only if $(Ax)^T(By) = 0$. So in this sense, the pair (A,B) preserves orthogonality.

Let now A be the matrix of a Singer cycle and let B denote its inverse-transpose. Let a_0 be a point represented by the vector $x \neq 0$ and b_0 a hyperplane represented by the vector $y \neq 0$. For $k \in \mathbb{Z}$, let a_k be the point represented by $A^k x$ and let b_k be the hyperplane represented by the vector $B^k y$. So $a_i = a_j$ if and only if $i \equiv j \pmod{N}$, and the same holds for the b_i. We also note by the foregoing that a_i and b_j are incident if and only if a_{i+1} and b_{j+1} are incident. This means that arranging the points in the order $a_0, \ldots, a_{N-1}$ and the hyperplanes in the order $b_0, \ldots, b_{N-1}$, the incidence matrix becomes a circulant. ◀

We obtain the following two corollaries from Theorem 58.

▶ **Corollary 60.** *For no family of Galois geometries is the corresponding family of point–hyperplane incidence matrices absolutely Valiant-rigid in characteristic zero.*

▶ **Corollary 61.** *For no family of Galois geometries is the corresponding family of point–hyperplane incidence matrices strictly Valiant-rigid over any fixed finite field.*

Galois planes are the Galois geometries $\mathrm{PG}(2,q)$. It follows from Corollary 61 that for no family of Galois planes is the corresponding family of point-line incidence matrices Valiant-rigid over $\mathbb{F}_2$. This is noteworthy because Valiant [16] suggested (without making a distinction between Desarguesian and non-Desarguesian planes) that the incidence matrices of finite projective planes might be candidates for rigidity over $\mathbb{F}_2$.

5.2 Vandermonde matrices

In this section we show that Vandermonde matrices of which the generators form a geometric progression are not absolutely Valiant-rigid.

▶ **Definition 62** (G-Hankel matrices)**.** *Let G be a finite abelian group of order n. Let $f : G \to \mathbb{F}$ be a function from G to a field $\mathbb{F}$. We define the G-Hankel matrix corresponding to f as the $n \times n$ matrix, whose rows and columns are labeled by the elements of G, and the element in position (g,h) is $f(g+h)$.*

As pointed out in [4], by permuting the rows of a G-Hankel matrix one can get a G-circulant matrix. Therefore such a pair of matrices has the same rigidity.

The classical *Hankel matrices* are the special case of G-Hankel matrices where G is the cyclic group of order n.

▶ **Observation 63.** *Let V be a Vandermonde matrix over a field $\mathbb{K}$ with generators that form a geometric progression. Then there exist diagonal matrices D_1 and D_2 over $\mathbb{K}$ such that D_1VD_2 is a Hankel matrix.*

Proof. Assume that the generators of V are sa^{i-1} for $i = 1, 2, \ldots, n$. Then the (i, j)-th entry of V is $s^{(j-1)}a^{(i-1)(j-1)}$. Define a pair of diagonal matrices with entries

$$(D_1)_{ii} = a^{i(i-1)/2} \quad \text{and} \quad (D_2)_{jj} = s^{-(j-1)}a^{j(j-1)/2}.$$

Clearly, the entries of D_1 and D_2 belong to K. Moreover,

$$(D_1VD_2)_{ij} = a^{1+(i+j)(i+j-3)/2}.$$

Thus, D_1VD_2 is a Hankel matrix. ◀

This observation, combined with part (a) of Theorem 58 by Dvir and Liu, yields the following corollary.

▶ **Corollary 64.** *Let $\mathcal{F}$ be a family of Vandermonde matrices over fields of characteristic zero, with generators that form a geometric progression. Then $\mathcal{F}$ is not absolutely Valiant-rigid.*

Proof. Note that multiplication by a diagonal matrix with non-zero entries does not change rigidity, so for D_1 and D_2 defined as above, $R_{\mathbb{K}}(V, r) = R_{\mathbb{K}}(D_1VD_2, r)$. ◀

5.3 Paley–Hadamard matrices

Hadamard matrices have for decades been considered candidates for rigidity. To everyone's surprise, Alman and Williams [2] recently showed that the Walsh–Hadamard matrices are not strictly Valiant-rigid.

In this section we remove a lot more Hadamard matrices from the list of rigidity candidates.

▶ **Corollary 65.** *No family of Paley–Hadamard matrices is absolutely Valiant-rigid.*

While the orders of the Walsh–Hadamard matrices are the powers of 2, the Paley–Hadamard matrices are exponentially more frequent: for every prime power $q \equiv -1 \pmod 4$ there is a Paley–Hadamard matrix of order $q + 1$, and for every prime power $q \equiv 1 \pmod 4$ there is a Paley–Hadamard matrix of order $2q + 2$.

Let q be an odd prime power and let $\chi : \mathbb{F}_q \to \{0, 1, -1\} \subseteq \mathbb{C}$ denote the quadratic character over $\mathbb{F}_q$. So for $x \in \mathbb{F}_q$, we have $\chi(x) = 0$ if $x = 0$; $\chi(x) = 1$ if $x \neq 0$ is a square in $\mathbb{F}_q$ and $\chi(x) = -1$ if x is not a square.

▶ **Definition 66** (Paley–Hadamard matrices). *For an odd prime power q define a $q \times q$ matrix Q with $Q_{i,j} = \chi(i - j)$.*

- *if $q \equiv -1 \mod 4$, consider a matrix $H = I + \begin{pmatrix} 0 & \mathbf{1}^{\mathbf{T}} \\ \mathbf{1} & Q \end{pmatrix}$, where $\mathbf{1}$ is an all-ones vector.*
- *if $q \equiv 1 \mod 4$, consider a matrix H obtained by replacing each entry of $\begin{pmatrix} 0 & \mathbf{1}^{\mathbf{T}} \\ \mathbf{1} & Q \end{pmatrix}$ with a 2×2 matrix in the following way.*
  1. *Each entry 0 is replaced with $\begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$;*
  2. *Each entry ± 1 is replaced with $\pm \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.*

The matrix H is a Hadamard matrix and is called a Paley–Hadamard matrix.

▶ **Observation 67.** *Let q be an odd prime power and let Q be the corresponding Paley–Hadamard matrix.*

- *If $q \equiv -1 \mod 4$, then the lower right $q \times q$ submatrix of H is a G-circulant matrix, where G is the additive group of $\mathbb{F}_q$.*
- *If $q \equiv 1 \mod 4$, then the lower right $2q \times 2q$ submatrix of H consists of 4 blocks that are G-circulants for the additive group of $\mathbb{F}_q$.*

Proof. If $q \equiv -1 \mod 4$, the statement immediately follows from the definition of the matrix Q. If $q \equiv 1 \mod 4$, denote by H_0 the right-lower $2q \times 2q$ submatrix. Note that the matrix obtained from H_0 by looking at the entries on the intersection of odd rows and odd columns is $Q + I$, and so is a circulant. Similarly, the matrices obtained by looking at the intersection of even rows and even columns, odd rows and even columns, and even rows and odd columns are circulants. ◀

This observation, combined with Theorem 58, proves Corollary 65.

References

1 Josh Alman. Kronecker products, low-depth circuits, and matrix rigidity. In *Proc. 53rd ACM Symp. on Theory of Computing (STOC'21)*, pages 772–785, 2021. arXiv:2102.11992. doi:10.1145/3406325.3451008.

2 Josh Alman and Ryan Williams. Probabilistic rank and matrix rigidity. In *Proc. 49th STOC*, pages 17:1–17:23. ACM Press, 2017. doi:10.1145/3055399.3055484.

3 Zeev Dvir and Benjamin Edelman. Matrix rigidity and the Croot-Lev-Pach lemma. *Theory of Computing*, 15(8):1–7, 2019. doi:10.4086/toc.2019.v015a008.

4 Zeev Dvir and Allen Liu. Fourier and circulant matrices are not rigid. *Theory of Computing*, 16(20):1–48, 2020. doi:10.4086/toc.2020.v016a020.

5 Joel Friedman. A note on matrix rigidity. *Combinatorica*, 13(2):235–239, 1993. doi:10.1007/BF01303207.

6 Oded Goldreich and Avishay Tal. Matrix rigidity of random Toeplitz matrices. *Comput. Complexity*, 27(2):305–350, 2018. Preliminary version in STOC'16. doi:10.1007/s00037-016-0144-9.

7 Robin Hartshorne. *Algebraic geometry.* Graduate texts in mathematics (52) Springer, 1977.

8 Bohdan Kivva. Improved upper bounds for the rigidity of Kronecker products. *arXiv*, 2021. arXiv:2103.05631.

9 Serge Lang. *Algebra*, volume 211 of *Grad. Texts in Math.* Springer, 3rd edition, 1996.

10 Satyanarayana V. Lokam. On the rigidity of Vandermonde matrices. *Theoret. Comput. Sci.*, 237(1–2):477–483, 2000. doi:10.1016/S0304-3975(00)00008-6.

11 Satyanarayana V. Lokam. Quadratic lower bounds on matrix rigidity. In *Internat. Conf. on Theory and Appl. of Models of Computation (TAMC'06)*, pages 295–307. Springer, 2006. doi:10.1007/11750321_28.

12 Alexander Razborov. On Rigid Matrices. Technical report, Steklov Mathematical Institute, 1989.

13 Alex Samorodnitsky, Ilya Shkredov, and Sergey Yekhanin. Kolmogorov width of discrete linear spaces: an approach to matrix rigidity. *Computational Complexity*, 25(2):309–348, 2016.

14 Mohammad Amin Shokrollahi, Daniel A. Spielman, and Volker Stemann. A remark on matrix rigidity. *Inform. Process. Lett.*, 64(6):283–285, 1997. doi:10.1016/S0020-0190(97)00190-7.

15 Victor Shoup and Roman Smolensky. Lower bounds for polynomial evaluation and interpolation. *Computational Complexity*, 6(4):301–311, 1997.

16 Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Math. Found. Comp. Sci. (MFCS'77)*, pages 162–176. Springer, 1977. doi:10.1007/3-540-08353-7_135.

A Basic concepts of algebraic geometry

In this appendix we review basic notions of the algebraic geometry that are needed in this paper. Our definitions follow [7], but, critically, we do not make the assumption that a field is algebraically closed.

Let $\mathbb{F}$ be an infinite field. $\mathbb{F}[x_1, x_2, \dots, x_n]$ denotes the ring of polynomials in variables $x_1, x_2, \dots, x_n$, with coefficients in $\mathbb{F}$.

▶ **Definition 68** (Affine algebraic set [7, p.2]). *A set $V \subseteq \mathbb{F}^n$ is called an (affine)* algebraic set *if it is the set of common zeros of a set of polynomials, $P \subseteq \mathbb{F}[x_1, x_2, \dots, x_n]$.*

▶ **Theorem 69** (Hilbert basis theorem). *Every affine algebraic set in $\mathbb{F}^n$ can be defined by a finite set of polynomials in $\mathbb{F}[x_1, x_2, \dots, x_n]$.*

▶ **Definition 70** (Irreduciblility). *A topological space is* irreducible *if it is not a union of two nonempty proper closed subsets.*

▶ **Observation 71.** *The intersection of a finite number of non-empty open subsets of an irreducible topological space is non-empty (and open).*

Proof. For two sets this is equivalent to the definition of irreducibility; the full statement follows by induction. ◀

▶ **Definition 72** (Zariski topology 1 [7, p.2]). *The* Zariski topology *on $\mathbb{F}^n$ is the topology in which the closed sets are precisely the affine algebraic sets of $\mathbb{F}^n$.*

▶ **Proposition 73.** *The Zariski topology on $\mathbb{F}^n$ is a topology, and $\mathbb{F}^n$ is irreducible.*

Proof. To prove irreducibility, let A_1 and A_2 be two Zariski-closed proper subsets of $\mathbb{F}^n$. Let the nonzero polynomial f_i vanish on A_i. Let $a \in \mathbb{F}^n$ be a point at which $(f_1 f_2)(a) \neq 0$. It follows that $a \notin A_1 \cup A_2$. ◀

▶ **Definition 74** (Locally closed set). *In a topological space, a set is called* locally closed *if it can be written as an intersection of an open set and a closed set.*

▶ **Definition 75** (Zariski topology 2). *Let $V \subseteq \mathbb{F}^n$ be a locally closed set in the Zariski topology. The Zariski topology on V is the restriction of the Zariski topology on $\mathbb{F}^n$ to V.*

▶ **Observation 76.** *Let V and W be topological spaces. Let $f : V \to W$ be a continuous surjective map. If V is irreducible, then W is irreducible.*

▶ **Definition 77** ((Quasi-)affine variety [7, p.3]). *An irreducible affine algebraic set is called an* affine variety*. A Zariski-open subset of an affine variety is called a* quasi-affine variety.

It is easy to see that a quasi-affine variety is irreducible by definition.

▶ **Definition 78** (Almost all). *We say that some property holds for* almost all *points in a (quasi-)affine variety if it holds for some non-empty Zariski-open subset of the variety.*

We now restate Obs. 71.

▶ **Observation 79.** *If each of a finite number of properties holds for almost all points of a quasi-variety V, then they all hold simultaneously for almost all points of V.*

▶ **Definition 80** (Regular function [7, p.15]). *Let V be a quasi-affine variety in $\mathbb{F}^n$. A function $f : V \to \mathbb{F}$ is* regular at a point $p \in V$ *if there exists a Zariski-open neighbourhood $p \in U \subset V$ and polynomials $g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$ such that h is nowhere zero on U and $f = g/h$ on U. We say that f is* regular *on V if it is regular at every point of V.*

▶ **Lemma 81** ([7, Lemma 3.1]). *A regular function $f : V \to \mathbb{F}$ is continuous.*

▶ **Definition 82** (Morphism [7, p.15]). *Let V, W be a pair of quasi-affine varieties. A* morphism *(or a* regular map*) $\phi : V \to W$ is a continuous map such that for every open set $U \in W$, and for every regular function $f : U \to \mathbb{F}$ the function $f \circ \phi : \phi^{-1}(U) \to \mathbb{F}$ is regular.*

Clearly, the composition of two morphisms is a morphism.

▶ **Lemma 83** ([7, Lemma 3.6]). *Let X be a quasi-affine variety and $Y \subseteq \mathbb{F}$ be an affine variety. A map (of sets) $\phi : X \to Y$ is a morphism if and only if $x_i \circ \phi$ is a regular function on X for each i, where $x_1, x_2, \dots, x_n$ are the coordinate functions on $\mathbb{F}^n$.*

▶ **Observation 84.** *Let $E_1, E_2, \dots, E_n$ be infinite subsets of $\mathbb{F}$. Suppose the polynomial $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ vanishes on the Cartesian product $E_1 \times \dots \times E_n$. Then f is the zero polynomial.*

▶ **Lemma 85.** *Let U be a non-empty Zariski-open subset of $\mathbb{F}^n$, where $\mathbb{F}$ is a subfield of $\mathbb{C}$. Then there exist $E_1, E_2, \dots E_n \subseteq \mathbb{F}$ such that $E_1 \times \dots \times E_n \subseteq U$ and each E_i is an unbounded set in $\mathbb{C}$.*

Proof. Since U is Zariski-open, there exists a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$, such that if $f(a_1, \dots a_n) \neq 0$, then $(a_1, \dots, a_n) \in U$. Let $E'_1, E'_2, \dots E'_n \subseteq \mathbb{F}$ be finite sets, such that for all $a_i \in E'_i$, $i \in [n]$ we have $f(a_1, a_2, \dots, a_n) \neq 0$. Then it is easy to see that for an arbitrary j there exists $b_j \notin E'_j$ such that the same condition holds when E'_j is replaced with $E'_j \cup \{b_j\}$. Moreover, such b_j can be taken so that its complex norm is greater than 1 plus the maximum of the norms of all elements that are currently in E'_j. Since we can in turn increment the size of each E_i, the claim follows by passing to the limit. ◀

▶ **Definition 86** (Constructible set). *In a topological space, a set is called* constructible *if it is a finite union of locally closed sets.*

▶ **Theorem 87** (Chevalley's theorem). *Let $f : V \to W$ be a regular map between algebraic sets over $\mathbb{F}$. Then $f(V)$ is a constructible set in the Zariski topology on W.*

B Omitted proofs

In this appendix we provide the proofs of Lemmas 46 and 47.

▶ **Lemma 88.** *Let $r \geq 3$. Consider $A_1 \in \mathbb{K}^{r\times r}$ and an invertible matrix $A_2 \in \mathbb{K}^{r\times r}$. Define*

$$v_i = -A_1^2 A_2^{-1} e_i + \omega^2 A_2^{-1} e_i \quad \text{and} \quad w_i = -A_2 A_1 A_2^{-1} e_i$$

Let $E_2 \subseteq \mathbb{K}$ be an unbounded set. For a diagonal matrix $Z \in \mathbb{C}^{r\times r}$ and $z_1, z_2 \in \mathbb{C}$, $x_2 \in \mathbb{K}$ consider

$$T(x_2, Z, z_1, z_2) = \begin{pmatrix} A_1 + Z & v_1 & v_2 \\ A_2 & w_1 + z_1 e_1 & w_2 + z_2 e_3 + x_2 e_2 \end{pmatrix}.$$

The set of matrices $(A_1, A_2) \in \mathbb{K}^{2r^2}$ such that for all $x_2 \in E_2$ there exist $Z \in \mathrm{Diag}(\mathbb{C}^{r\times r})$, and $z_1, z_2 \in \mathbb{C}$ such that $\mathrm{rank}(T(x_2, Z, z_1, z_2)) \leq r$ is small in $\mathbb{K}^{2r^2}$.

Proof. Assume $\operatorname{rank}(T(x_2, Z, z_1, z_2)) \leq r$. Since A_2 is invertible, the last two columns of $B(x_2, Z, z_1, z_2)$ can be expressed as a linear combination of the first r columns.

For convenience, define $x_1 = 0$, $j_1 = 1$ and $j_2 = 3$. Let $y_i \in \mathbb{K}^r$ satisfy

$$\begin{pmatrix} A_1 + Z \\ A_2 \end{pmatrix} y_i = \begin{pmatrix} v_i \\ w_i + z_i e_{j_i} + x_i e_2 \end{pmatrix}.$$

Then

$$y_i = A_2^{-1}(w_i + z_i e_{j_i} + x_i e_2) \quad \Rightarrow \quad (A_1 + Z)A_2^{-1}(w_i + z_i e_{j_i} + x_i e_2) = v_i,$$

$$-A_1^2 A_2^{-1} e_i + z_i A_1 A_2^{-1} e_{j_i} + x_i A_1 A_2^{-1} e_2 + Z(-A_1 A_2^{-1} e_i + z_i A_2^{-1} e_{j_i} + x_i A_2^{-1} e_2) = -A_1^2 A_2^{-1} e_i + \omega^2 A_2^{-1} e_i.$$

Let $\alpha_i = A_1 A_2^{-1} e_i$ and $\beta_i = A_2^{-1} e_i$. Then for all $k \in [r]$ we have

$$Z_{kk} = \frac{\omega^2 \beta_{1k} - z_1 \alpha_{1k}}{-\alpha_{1k} + z_1 \beta_{1k}} \quad \text{and} \quad Z_{kk} = \frac{\omega^2 \beta_{2k} - z_2 \alpha_{3k} - x_2 \alpha_{2k}}{-\alpha_{2k} + z_2 \beta_{3k} + x_2 \beta_{2k}}.$$

Hence, for all $k \in [r]$,

$$\frac{\omega^2 \beta_{1k} - z_1 \alpha_{1k}}{-\alpha_{1k} + z_1 \beta_{1k}} = \frac{\omega^2 \beta_{2k} - z_2 \alpha_{3k} - x_2 \alpha_{2k}}{-\alpha_{2k} + z_2 \beta_{3k} + x_2 \beta_{2k}}. \tag{14}$$

By passing to a subsequence for $x_2 \in E_2$ we may assume that $\lim\limits_{E_2 \ni x_2 \to \infty} z_2(x_2)/x_2 = c \in \widehat{\mathbb{C}}$ and $\lim\limits_{E_2 \ni x_2 \to \infty} z_1(x_2) = c' \in \widehat{\mathbb{C}}$ are well-defined. Then we must have

$$\frac{\omega^2 \beta_{1k} - c' \alpha_{1k}}{-\alpha_{1k} + c' \beta_{1k}} = -\frac{c \alpha_{3k} + \alpha_{2k}}{c \beta_{3k} + \beta_{2k}} \qquad \forall k \in [r].$$

If $c \neq \infty$, for every k this gives a non-trivial rational equation for α_{2k} in terms of other variables α_{ik}, β_{ik} and c, c'. If $c = \infty$, for every k we get a nontrivial rational equation for α_{3k} in terms of other variables and c'. In any case, for $r \geq 3$ the set of matrices $(A_1, A_2) \in \mathbb{K}^{2r^2}$ that satisfy Eq. (14) is small in $\mathbb{K}^{2r^2}$. ◀

▶ **Lemma 89.** *Let $r \geq 3$. Let $j_1 \notin \{1, 2\}$ be an element of $[r]$. Let $E_1, E_2 \subseteq \mathbb{K}$ be unbounded sets. For $v_1, v_2, w_1, w_2 \in \mathbb{K}^r$, $A_1 \in \mathbb{K}^{r \times r}$, an invertible matrix $A_2 \in \mathbb{K}^{r \times r}$, $x_1, x_2 \in \mathbb{K}$, $z_1, z_2 \in \mathbb{C}$ and a diagonal matrix $Z \in \mathbb{C}^{r \times r}$ consider*

$$T(x_1, x_2, Z, z_1, z_2) = \begin{pmatrix} A_1 + Z & v_1 & v_2 + x_2 e_1 \\ A_2 & w_1 + z_1 e_1 + x_1 e_{j_1} & w_2 + z_2 e_2 \end{pmatrix}.$$

The set of matrices $(A_1, A_2) \in \mathbb{K}^{2r^2}$, for which there exist $v_1, v_2, w_1, w_2 \in \mathbb{K}^r$, s.t. for all $x_1 \in E_1$ and $x_2 \in E_2$ there exist $Z \in \operatorname{Diag}(\mathbb{C}^{r \times r})$, $z_1, z_2 \in \mathbb{C}$ s.t. $\operatorname{rank}(T(x_1, x_2, Z, z_1, z_2)) \leq r$, is small in $\mathbb{K}^{2r^2}$.

Proof. Similarly, as in Lemma 45, Eq. (11) holds for $i = 1$. For the second column we get

$$(A_1 + Z)A_2^{-1}(w_2 + z_2 e_2) = v_2 + x_2 e_1.$$

Denote $\gamma_i = v_i - A_1 A_2^{-1} w_i$, $\alpha_i = A_1 A_2^{-1} e_i$, $\beta_i = A_2^{-1} e_i$ and $\phi_i = A_2^{-1} w_i$, then

$$Z_{kk} = \frac{\gamma_{2k} - x_2 \mathbf{1}[k = 1] - z_2 \alpha_{2k}}{\phi_{2k} + z_2 \beta_{2k}}.$$

Combining this with Eq. (11) for $i = 1$, we get

$$\frac{\gamma_{1k} - z_1\alpha_{1k} - x_1\alpha_{j_1k}}{\phi_{1k} + z_1\beta_{1k} + x_1\beta_{j_1k}} = \frac{\gamma_{2k} - x_2\mathbf{1}[k=1] - z_2\alpha_{2k}}{\phi_{2k} + z_2\beta_{2k}}.$$

Similarly, as in Lemma 45, by fixing $x_2 \in E_2$ and passing to the subsequence for $x_1 \in E_1$, we deduce that there exist $c(x_2) \in \widehat{\mathbb{C}}$ and $z_2 = z_2(x_2) \in \widehat{\mathbb{C}}$ such that

$$\frac{\gamma_{2k} - x_2\mathbf{1}[k=1] - z_2\alpha_{2k}}{\phi_{2k} + z_2\beta_{2k}} = -\frac{\alpha_{1k}c(x_2) + \alpha_{j_1k}}{\beta_{1k}c(x_2) + \beta_{j_1k}} \qquad \forall k \in [r].$$

Again, as in Lemma 45, by passing to the subsequence for $x_2 \in E_2$ we may deduce that there exist c' and c'' in $\widehat{\mathbb{C}}$ such that

$$\frac{\mathbf{1}[k=1] + c'\alpha_{2k}}{c'\beta_{2k}} = \frac{\alpha_{1k}c'' + \alpha_{j_1k}}{\beta_{1k}c'' + \beta_{j_1k}} \qquad \forall k \in [r]. \tag{15}$$

If $c'' \neq 0$, then α_{1k} can be expressed through other variables α_{ik}, β_{ik} and c', c''. Since $j_1 \neq 2$, if $c'' = 0$, then α_{j_1k} can be expressed in terms of other variables α_{ik}, β_{ik} and c'. Thus, the set of matrices (A_1, A_2) that satisfy Eq. (15) is small in $\mathbb{K}^{2r^2}$. ◀

C Reduction to countable fields

In this section we outline the basic model theory that allows us to consider countable fields only for our main result.

▶ **Proposition 90.** *Let us fix positive integers n, r, s. Let $X = (x_{ij})$ be an $n \times n$ matrix of variables. Then there is a first-order formula $\varphi(x_{ij})$ in the language of fields that expresses, over any field $\mathbb{F}$, the statement that $R_{\mathbb{F}}(X, r) = s$.*

Proof. Rank is first-order expressible (look at a finite number of determinants). There is a finite number of s-tuples where the matrix can be changed. Combine these. ◀

Let us fix positive integers n, r, s, t. We wish to prove a statement of the following form:

($**$) If $\mathbb{K}$ is a field of characteristic zero and $\mathbb{L}/\mathbb{K}$ is a quadratic extension then there exists an $n \times n$ matrix A over $\mathbb{K}$ such that $R_{\mathbb{K}}(A, r) \geq s$ and $R_{\mathbb{L}}(A, r) \leq t$.

▶ **Proposition 91.** *If statement ($**$) holds whenever $\mathbb{K}$ is countable then it always holds.*

Proof. Let $\mathbb{L} = \mathbb{K}[\omega]$ where $\omega^2 =: u \in \mathbb{K}$. Let us add a name for u as a constant to the signature of rings, so we talk about the model $(\mathbb{K}, u)$. By the downward Löwenheim–Skolem theorem, this model has a countable elementary submodel $(\mathbb{K}', u)$. Let now $\mathbb{L}' = \mathbb{K}'[\omega]$. So $\mathbb{L}'/\mathbb{K}'$ is a quadratic extension (because $u \in \mathbb{K}'$).

Let us now apply ($**$) to this extension. Let A be a matrix over $\mathbb{K}'$ with the required properties: $R_{\mathbb{K}'}(A, r) \geq s$ and $R_{\mathbb{L}'}(A, r) \leq t$.

Now $R_{\mathbb{L}}(A, r) \leq t$ follows immediately because $\mathbb{L}' \subseteq \mathbb{L}$. On the other hand, in the light of Prop. 90, $R_{\mathbb{K}'}(A, r) = R_{\mathbb{K}}(A, r)$, because $\mathbb{K}'$ is an elementary submodel of $\mathbb{K}$. ◀

D A 5×5 matrix with different strict and absolute rigidity

In this appendix we provide a concrete example of a matrix that shows a difference between strict and absolute rigidity. Specifically, we exhibit a matrix $A \in \mathbb{Q}^{5\times 5}$ such that $R_{\mathbb{Q}}(A,2) = 9$ and $R_{\mathbb{Q}[\sqrt{2}]}(A,2) = 8$. Consider the 5×2 and 2×5 matrices

$$L = \begin{pmatrix} 1 & -\sqrt{2} \\ \sqrt{2} & -1 \\ 3-\sqrt{2} & 1 \\ 12-7\sqrt{2} & 1 \\ 10-7\sqrt{2} & 1+2\sqrt{2} \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 0 & 2+\sqrt{2} & 3+2\sqrt{2} & 1 \\ \sqrt{2} & 1 & 1+2\sqrt{2} & 2-3\sqrt{2} & 3+\sqrt{2} \end{pmatrix}.$$

The product LR has 8 irrational entries:

$$L \cdot R = \begin{pmatrix} -1 & -\sqrt{2} & -2 & 9 & -1-3\sqrt{2} \\ 0 & -1 & 1 & 2+6\sqrt{2} & -3 \\ 3 & 1 & 5+3\sqrt{2} & 7 & 6 \\ 12-6\sqrt{2} & 1 & 11 & 10 & 15-6\sqrt{2} \\ 14-6\sqrt{2} & 1+2\sqrt{2} & 15 & -8 & 17 \end{pmatrix}$$

The following matrix, $A \in \mathbb{Q}^{5\times 5}$, differs from LR in only these 8 entries.

$$A = \frac{1}{16}\begin{pmatrix} -16 & 34 & -32 & 144 & 67 \\ 0 & -16 & 16 & -89 & -48 \\ 48 & 16 & 43 & 112 & 96 \\ 137 & 16 & 176 & 160 & -92 \\ 39 & 73 & 240 & -128 & 272 \end{pmatrix} \tag{16}$$

In other words,

$$A - LR = \begin{pmatrix} 0 & * & 0 & 0 & * \\ 0 & 0 & 0 & * & 0 \\ 0 & 0 & * & 0 & 0 \\ * & 0 & 0 & 0 & * \\ * & * & 0 & 0 & 0 \end{pmatrix}, \tag{17}$$

where each $*$ hides some non-zero entry. We selected every entry of A at positions marked by $*$ independently uniformly at random from $\{-135/16, -134/16, \ldots, 135/16\}$.

Note that Eq. (17) immediately implies that $R_{\mathbb{Q}[\sqrt{2}]}(A,2) \leq 8$.

We use exhaustive computer search to verify that $R_{\mathbb{Q}}(A,2) > 8$. We consider all the $\binom{25}{8} = 1{,}081{,}575$ combinations of 8 cells among the 5×5 cells. Having fixed a set of 8 cells, we introduce variables for their entries, and use Matlab to verify that the system of $\binom{5}{3}^2 = 100$ polynomial equations, saying that the determinant of every 3×3 minor is zero, has no rational solutions. In fact, we obtain the following stronger result.

▶ **Proposition 92.** *If a 5×5 complex matrix B of rank ≤ 2 differs from A in at most 8 positions then B is either LR or its algebraic conjugate (replace every occurrence of $\sqrt{2}$ by $-\sqrt{2}$).*