

# 33rd Computational Complexity Conference

CCC 2018, June 22–24, 2018, San Diego, California, USA

Edited by

Rocco A. Servedio



*Editor*

Rocco A. Servedio  
Department of Computer Science  
Columbia University  
500 West 120 Street  
New York, New York 10027  
USA  
rocco@cs.columbia.edu

*ACM Classification 2012*

Theory of computation

**ISBN 978-3-95977-069-9**

*Published online and open access by*

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <http://www.dagstuhl.de/dagpub/978-3-95977-069-9>.

*Publication date*

June, 2018

*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 <http://dnb.d-nb.de>.

*License*

This work is licensed under a Creative Commons Attribution 3.0 Unported license (CC-BY 3.0): <http://creativecommons.org/licenses/by/3.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.CCC.2018.0

ISBN 978-3-95977-069-9

ISSN 1868-8969

<http://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.

### *Editorial Board*

- Luca Aceto (*Chair*, Gran Sasso Science Institute and Reykjavik University)
- Susanne Albers (TU München)
- Chris Hankin (Imperial College London)
- Deepak Kapur (University of New Mexico)
- Michael Mitzenmacher (Harvard University)
- Madhavan Mukund (Chennai Mathematical Institute)
- Anca Muscholl (University Bordeaux)
- Catuscia Palamidessi (INRIA)
- Raimund Seidel (Saarland University and Schloss Dagstuhl – Leibniz-Zentrum für Informatik)
- Thomas Schwentick (TU Dortmund)
- Reinhard Wilhelm (Saarland University)

**ISSN 1868-8969**

**<http://www.dagstuhl.de/lipics>**





## ■ Contents

Preface	
<i>Rocco A. Servedio</i> .....	0:vii
<b>Papers</b>	
Pseudorandom Generators from Polarizing Random Walks	
<i>Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett</i> .....	1:1–1:21
A PRG for Boolean PTF of Degree 2 with Seed Length Subpolynomial in $\epsilon$ and Logarithmic in $n$	
<i>Daniel Kane and Sankeerth Rao</i> .....	2:1–2:24
A New Approach for Constructing Low-Error, Two-Source Extractors	
<i>Avraham Ben-Aroya, Eshan Chattopadhyay, Dean Doron, Xin Li, and Amnon Ta-Shma</i> .....	3:1–3:19
Lossless Dimension Expanders via Linearized Polynomials and Subspace Designs	
<i>Venkatesan Guruswami, Nicolas Resch, and Chaoping Xing</i> .....	4:1–4:16
NP-hardness of Minimum Circuit Size Problem for OR-AND-MOD Circuits	
<i>Shuichi Hirahara, Igor C. Oliveira, and Rahul Santhanam</i> .....	5:1–5:31
Limits on Representing Boolean Functions by Linear Combinations of Simple Functions: Thresholds, ReLUs, and Low-Degree Polynomials	
<i>Richard Ryan Williams</i> .....	6:1–6:24
The Power of Natural Properties as Oracles	
<i>Russell Impagliazzo, Valentine Kabanets, and Ilya Volkovich</i> .....	7:1–7:20
Linear Sketching over $\mathbb{F}_2$	
<i>Sampath Kannan, Elchanan Mossel, Swagato Sanyal, and Grigory Yaroslavtsev</i> ..	8:1–8:37
Communication Complexity with Small Advantage	
<i>Thomas Watson</i> .....	9:1–9:17
Algebraic Dependencies and PSPACE Algorithms in Approximative Complexity	
<i>Zeyu Guo, Nitin Saxena, and Amit Sinhababu</i> .....	10:1–10:21
Unbalancing Sets and an Almost Quadratic Lower Bound for Syntactically Multilinear Arithmetic Circuits	
<i>Noga Alon, Mrinal Kumar, and Ben Lee Volk</i> .....	11:1–11:16
Hardness Amplification for Non-Commutative Arithmetic Circuits	
<i>Marco L. Carmosino, Russell Impagliazzo, Shachar Lovett, and Ivan Mihajlin</i> ....	12:1–12:16
Hardness vs Randomness for Bounded Depth Arithmetic Circuits	
<i>Chi-Ning Chou, Mrinal Kumar, and Noam Solomon</i> .....	13:1–13:17
On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product	
<i>Lijie Chen</i> .....	14:1–14:45

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio



Leibniz International Proceedings in Informatics  
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Hardness of Function Composition for Semantic Read once Branching Programs <i>Jeff Edmonds, Venkatesh Medabalimi, and Toniann Pitassi</i> .....	15:1–15:22
Reordering Rule Makes OBDD Proof Systems Stronger <i>Sam Buss, Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov</i> .....	16:1–16:24
Testing Linearity against Non-Signaling Strategies <i>Alessandro Chiesa, Peter Manohar, and Igor Shinkar</i> .....	17:1–17:37
Earthmover Resilience and Testing in Ordered Structures <i>Omri Ben-Eliezer and Eldar Fischer</i> .....	18:1–18:35
New Hardness Results for the Permanent Using Linear Optics <i>Daniel Grier and Luke Schaeffer</i> .....	19:1–19:29
Two-Player Entangled Games are NP-Hard <i>Anand Natarajan and Thomas Vidick</i> .....	20:1–20:18
Complexity Classification of Conjugated Clifford Circuits <i>Adam Bouland, Joseph F. Fitzsimons, and Dax Enshan Koh</i> .....	21:1–21:25
Efficient Batch Verification for UP <i>Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum</i> .....	22:1–22:23
A Tight Lower Bound for Entropy Flattening <i>Yi-Hsiu Chen, Mika Göös, Salil P. Vadhan, and Jiapeng Zhang</i> .....	23:1–23:28
Worst-Case to Average Case Reductions for the Distance to a Code <i>Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf</i> .....	24:1–24:23
On the Complexity of the Cayley Semigroup Membership Problem <i>Lukas Fleischer</i> .....	25:1–25:12
Small Normalized Boolean Circuits for Semi-disjoint Bilinear Forms Require Logarithmic Conjunction-depth <i>Andrzej Lingas</i> .....	26:1–26:10
Lower Bounds on Non-Adaptive Data Structures Maintaining Sets of Numbers, from Sunflowers <i>Sivaramakrishnan Natarajan Ramamoorthy and Anup Rao</i> .....	27:1–27:16
Dimension Reduction for Polynomials over Gaussian Space and Applications <i>Badih Ghazi, Pritish Kamath, and Prasad Raghavendra</i> .....	28:1–28:37

## ■ Preface

The papers in this volume were accepted for presentation at the 33rd Computational Complexity Conference (CCC 2018), held June 22–24, 2018 in San Diego, California. The conference is organized by the Computational Complexity Foundation in cooperation with the European Association for Theoretical Computer Science (EATCS) and the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT). CCC 2018 is sponsored by Microsoft Research.

The call for papers sought original research papers in all areas of computational complexity theory. Of the 74 submissions the program committee selected 28 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 Dieter van Melkebeek for extensive advice and assistance; Ryan O’Donnell and David Zuckerman for sharing their knowledge as prior PC chairs for CCC; Andrei Krokhin for contributing two one-hour tutorials on the topic of “Constraints, Consistency and Complexity”; and Michael Wagner for coordinating the production of these proceedings.

Rocco A. Servedio

Program Committee Chair, on behalf of the Program Committee





## ■ Awards

The program committee of the 33rd Computational Complexity Conference is very pleased to present the **Best Student Paper Award** to Lukas Fleischer for his paper

*The Complexity of the Cayley Semigroup Membership Problem.*

Funding for the best student paper award is provided by the European Association for Theoretical Computer Science (EATCS).





## ■ Conference Organization

### Program Committee

Eric Allender, Rutgers University  
Paul Beame, University of Washington  
Eric Blais, University of Waterloo  
Mark Braverman, Princeton University  
Michael A. Forbes, University of Illinois Urbana-Champaign  
Shafi Goldwasser, Massachusetts Institute of Technology and Weizmann Institute  
Rocco A. Servedio (Chair), Columbia University  
Srikanth Srinivasan, Indian Institute of Technology Bombay  
Thomas Thierauf, Aalen University  
Madhur Tulsiani, Toyota Technological Institute at Chicago  
Henry Yuen, University of California, Berkeley and University of Toronto

### Local Arrangements Committee

Sam Buss, University of California, San Diego  
Shachar Lovett (chair), University of California, San Diego

### Board of Trustees

Boaz Barak, Harvard University  
Sevag Gharibian, University of Paderborn and Virginia Commonwealth University  
Shachar Lovett, University of California, San Diego  
Dieter van Melkebeek (President), University of Wisconsin-Madison  
Ryan O'Donnell, Carnegie Mellon University  
Rahul Santhanam, Oxford University  
Rocco A. Servedio, Columbia University







## ■ External Reviewers

Amir Abboud	Divesh Aggarwal	Robert Andrews
Nikhil Balaji	Jess Banks	David Barrington
Niel de Beaudrap	Alexander Belov	Shalev Ben-David
Amev Bhangale	Arnab Bhattacharyya	Pranav Bisht
Andrej Bogdanov	Adam Bouland	Karl Bringmann
Joshua Brody	Jonah Brown-Cohen	Amit Chakrabarti
Richard Chang	Arkadev Chattopadhyay	Eshan Chattopadhyay
Gil Cohen	Samir Datta	Anindya De
Jian Ding	Dean Doron	Andrew Drucker
Lior Eldar	Shai Evra	Omar Fawzi
Bill Fefferman	Stephen Fenner	Eldar Fischer
Venkata Gandikota	Sumegha Garg	Dmitry Gavinsky
Sumanta Ghosh	Michael Goodrich	Elena Grigorescu
Ofer Grossman	Tom Gur	Iftach Haitner
Dhiraj Holden	Justin Holmgren	Pavel Hrubes
Pavel Hubacek	Christian Ikenmeyer	Fernando Jeronimo
Zhengfeng Ji	Stasys Jukna	Valentine Kabanets
Yael Tauman Kalai	Neeraj Kayal	Swastik Kopparty
Robin Kothari	Alexander Kulikov	Akash Kumar
Alex Lubotzky	Guillaume Malod	Pasin Manurangsi
Jieming Mao	Pierre McKenzie	Or Meir
Stefan Mengel	Dor Minzer	Ashley Montanaro
Anand Natarajan	Joe Neeman	Huy Nguyen
Chinmay Nirkhe	Jerri Nummenpalo	Igor Carboni Oliveira
Eran Omri	Ori Parzanchevski	Ramamohan Paturi
Aduri Pavan	Supartha Podder	Aaron Potechin
Manoj Prabhakaran	Youming Qiao	Jaikumar Radhakrishnan
Nicolas Resch	Robert Robere	Cristobal Rojas
Ron Rothblum	Aviad Rubinstein	Michael Saks
Rahul Santhanam	Nitin Saxena	Luke Schaeffer
Dominik Scheder	Jon Schneider	Uwe Schöning
Matthias Schröder	Igor Sergeev	Ronen Shaltiel
Suhail Sherif	Igor Shinkar	Amir Shpilka
Amit Sinhababu	Nick Spooner	Noah Stephens-Davidowitz
Avishay Tal	Li-Yang Tan	Justin Thaler
Robin Thomas	Jacobo Toran	Prashant Vasudevan
Thomas Vidick	Marc Vinyals	Ben Lee Volk
Erik Waingarten	Thomas Watson	Omri Weinstein
Ryan Williams	Karl Wimmer	John Wright
Grigory Yaroslavtsev	Amir Yehudayoff	Yuichi Yoshida
Shengyu Zhang	Standa Zivny	

33rd Computational Complexity Conference (CCC 2018).  
Editor: Rocco A. Servedio



Leibniz International Proceedings in Informatics  
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



# Pseudorandom Generators from Polarizing Random Walks

Eshan Chattopadhyay<sup>1</sup>

Cornell University and IAS, Princeton, USA  
eshanc@ias.edu

Pooya Hatami<sup>2</sup>

University of Texas at Austin, USA  
pooyahat@gmail.com

Kaave Hosseini<sup>3</sup>

University of California, San Diego, USA  
skhossei@ucsd.edu

Shachar Lovett<sup>4</sup>

University of California, San Diego, USA  
slovett@ucsd.edu

---

## Abstract

We propose a new framework for constructing pseudorandom generators for  $n$ -variate Boolean functions. It is based on two new notions. First, we introduce fractional pseudorandom generators, which are pseudorandom distributions taking values in  $[-1, 1]^n$ . Next, we use a fractional pseudorandom generator as steps of a random walk in  $[-1, 1]^n$  that converges to  $\{-1, 1\}^n$ . We prove that this random walk converges fast (in time logarithmic in  $n$ ) due to polarization. As an application, we construct pseudorandom generators for Boolean functions with bounded Fourier tails. We use this to obtain a pseudorandom generator for functions with sensitivity  $s$ , whose seed length is polynomial in  $s$ . Other examples include functions computed by branching programs of various sorts or by bounded depth circuits.

**2012 ACM Subject Classification** Theory of computation → Pseudorandomness and derandomization

**Keywords and phrases** AC0, branching program, polarization, pseudorandom generators, random walks, Sensitivity

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.1

**Acknowledgements** We thank Avi Wigderson and David Zuckerman for various stimulating discussions during the course of our work.

## 1 Introduction

Pseudorandom generators (PRG) are widely studied in complexity theory. There are several general frameworks used to construct PRGs. One is based on basic building blocks, such as small bias generators [15, 2],  $k$ -wise independence, or expander graphs [10]. Another

---

<sup>1</sup> Supported by NSF grant CCF-1412958 and the Simons foundation.

<sup>2</sup> Supported by a Simons Investigator Award (#409864, David Zuckerman).

<sup>3</sup> Supported by NSF grant CCF-1614023.

<sup>4</sup> Supported by NSF grant CCF-1614023.



approach is based on hardness vs randomness paradigm, which was introduced by Nisan and Wigderson [17] and has been very influential. Many of the hardness results used in the latter framework are based on random restrictions, and the analysis of how they simplify the target class of functions. The number of papers in these lines of work is on the order of hundreds, so we do not even attempt to give a comprehensive survey of them all.

The purpose of this paper is to introduce a new framework for constructing PRGs based on polarizing random walks. We develop the theory in this paper and give a number of applications; perhaps the most notable one is a PRG for functions of sensitivity  $s$  whose seed length is polynomial in  $s$ . But, as this is a new framework, there are many questions that arise, both technical and conceptual, and we view this paper as mostly preliminary, with the hope that many more applications would follow.

## 1.1 PRGs and fractional PRGs

Let  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  be a Boolean function. The standard definition of a PRG for  $f$  with error  $\varepsilon > 0$ , is a random variable  $X \in \{-1, 1\}^n$  such that

$$|\mathbb{E}_X[f(X)] - \mathbb{E}_U[f(U)]| \leq \varepsilon,$$

where  $U$  denotes a random variable with the uniform distribution in  $\{-1, 1\}^n$ . We relax this definition by introducing a new object called a *fractional PRG*, defined in the next paragraph.

To prepare the notation for the definition, identify  $f$  with a real multi-linear polynomial, namely its Fourier expansion. This extends  $f$  to  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , although, we would only be interested in inputs from  $[-1, 1]^n$ . Observe that if  $x \in [-1, 1]^n$  then  $f(x) = \mathbb{E}_X[f(X)]$  where  $X \in \{-1, 1\}^n$  is a random variable sampled as follows: for every  $i \in [n]$  sample  $X_i \in \{-1, 1\}$  independently with  $\mathbb{E}[X_i] = x_i$ . In particular,  $f$  on  $[-1, 1]^n$  is bounded, namely  $f : [-1, 1]^n \rightarrow [-1, 1]$ . Also,  $f(\bar{0}) = \mathbb{E}_U[f(U)]$ . The following is a key definition.

► **Definition 1** (Fractional PRG). Let  $f : [-1, 1]^n \rightarrow [-1, 1]$  be multilinear. A fractional PRG for  $f$  is a random variable  $X \in [-1, 1]^n$  such that

$$|\mathbb{E}_X[f(X)] - f(\bar{0})| \leq \varepsilon.$$

One trivial construction of a fractional PRG is  $X \equiv \bar{0}$  but this is not going to be useful for our purpose of constructing PRGs. To disallow such examples, we require each coordinate of  $X$  to be far from zero with some noticeable probability. Formally,  $X \in [-1, 1]^n$  is called  $p$ -noticeable if  $\mathbb{E}[X_i^2] \geq p$  for all  $i = 1, \dots, n$ .

A good example to keep in mind is the following. Let  $G : \{-1, 1\}^r \rightarrow \{-1, 1\}^n$  be a (Boolean valued) function, and set  $X = pG(U)$ , where  $U \in \{-1, 1\}^r$  is uniform. Notice that  $X$  is  $p^2$ -noticeable. In this case we say  $X$  has seed length  $r$ . More generally,  $X$  has seed length  $r$  if  $X = G(U)$  where  $G : \{-1, 1\}^r \rightarrow [-1, 1]^n$ .

Fractional PRGs are easier to construct than standard PRGs, as they can take values in  $[-1, 1]^n$ . For example, assume that  $f$  has Fourier tails bounded in  $L_1$ . That is, there exist parameters  $a, b \geq 1$  for which

$$\sum_{S \subset [n]: |S|=k} |\hat{f}(S)| \leq a \cdot b^k \quad \forall k = 1, \dots, n.$$

We show (in Lemma 22) that if  $X \in [-1, 1]^n$  is small-biased, then  $pX$  is a fractional PRG for  $f$  with  $p \approx 1/b$ . The reason is that this choice of  $p$  controls all the Fourier coefficients of  $f$  with large Hamming weight, while  $X$  controls the ones with small weight. (In fact, to optimize parameters one can choose  $X$  to be almost  $k$ -wise independent; see Lemma 22 for details). In any case, note that  $pX$  is  $p^2$ -noticeable as  $pX$  takes values in  $\{-p, p\}^n$ .

## 1.2 Fractional PRG as steps in a random walk

Let  $X \in [-1, 1]^n$  be a fractional PRG for  $f$  with error  $\varepsilon$ . That is,

$$|\mathbb{E}_X[f(X)] - f(\bar{0})| \leq \varepsilon.$$

The goal is to construct a random variable  $Y \in \{-1, 1\}^n$  such that  $\mathbb{E}_Y[f(Y)] \approx f(\bar{0})$ , where the fractional PRG  $X$  provides a “small step” towards this approximation. If we can combine these small steps in a way that they converge fast to  $\{-1, 1\}^n$ , then we would be done. To be a bit more precise, consider a random walk starting at  $\bar{0}$  with the following properties:

1. The value of  $f$  at each step typically does not change by too much.
2. The random walk converges fast to  $\{-1, 1\}^n$ .

Observe that if we take  $X$  as the first step, then property 1 is satisfied for the first step. Considering later steps leads to the following question: Given a point  $\alpha \in [-1, 1]^n$ , can we find a random variable  $A = A(\alpha, X)$  such that

$$|\mathbb{E}[f(A)] - f(\alpha)| \leq \varepsilon,$$

and such that  $A$  takes values closer to Boolean values? We show that this is indeed the case if we assume that  $X$  not only fools  $f$ , but also fools any possible restriction of  $f$ .

To formalize this, let  $\mathcal{F}$  be a family of  $n$ -variate Boolean functions  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ . We say that  $\mathcal{F}$  is closed under restrictions if for any  $f \in \mathcal{F}$ , if we fix some inputs of  $f$  to constants  $\{-1, 1\}$ , then the new restricted function is still in  $\mathcal{F}$ . Most natural families of Boolean functions studied satisfy this condition. Some examples are functions computed by small-depth circuits, functions computed by bounded width branching programs, and functions of low sensitivity.

We show that if  $X$  is a fractional PRG for such  $\mathcal{F}$ , then it can be used to approximate  $f(\alpha)$  for any  $\alpha \in [-1, 1]^n$ . Define  $\delta_\alpha \in [0, 1]^n$  by  $(\delta_\alpha)_i = 1 - |\alpha_i|$ . For  $x, y \in [-1, 1]^n$  define  $x \circ y \in [-1, 1]^n$  to be their coordinate-wise product,  $(x \circ y)_i = x_i y_i$ . Note that under this definition, the sub-cube  $\{\alpha + \delta_\alpha \circ y : y \in [-1, 1]^n\}$  is the largest symmetric sub-cube of  $[-1, 1]^n$  centered at  $\alpha$ .

We show (Claim 15) that if  $X \in [-1, 1]^n$  is a fractional PRG for  $\mathcal{F}$  which is closed under restrictions, then for any  $f \in \mathcal{F}$  and any  $\alpha \in [-1, 1]^n$  it holds that

$$|\mathbb{E}[f(\alpha + \delta_\alpha \circ X)] - f(\alpha)| \leq \varepsilon.$$

Technically, we need to also assume that  $X$  is *symmetric*, which means that  $\Pr[X = x] = \Pr[X = -x]$  for all  $x$ . This is easy to achieve from any  $X$  which is not symmetric, for example by multiplying  $X$  with a uniform bit (thus, increasing its seed length by 1 bit).

## 1.3 Polarization and fast convergence

Our next goal is to show fast convergence of the random walk to  $\{-1, 1\}^n$ . To that end, we need to analyze the following martingale:

$$\begin{aligned} Y_1 &= X_1 \\ Y_i &= Y_{i-1} + \delta_{Y_{i-1}} \circ X_i \end{aligned}$$

where  $X_1, X_2, \dots$  are independent copies of a fractional PRG. We show that for some  $t$  not too large,  $Y_t$  is close to a point in  $\{-1, 1\}^n$ . But why would that be true? This turns out to

be the result of *polarization* in the random walk. It suffices to show this for every coordinate individually.

So, let  $Z_1, Z_2, \dots \in [-1, 1]$  be independent random variables (which are the  $i$ -th coordinate of  $X_1, X_2, \dots$  for some fixed  $i$ ), and define the following one-dimensional martingale:

$$\begin{aligned} W_1 &= Z_1 \\ W_i &= W_{i-1} + (1 - |W_{i-1}|)Z_i. \end{aligned}$$

Claim 17 shows that if (i)  $Z_i$  is symmetric, and (ii)  $\mathbb{E}[Z_i^2] \geq p$  (which follows from our assumption that the fractional PRG is  $p$ -noticeable), then it holds that

$$\Pr[|W_t| \geq 1 - \delta] \geq 1 - \delta$$

for  $t = O(\log(1/\delta)/p)$ . Setting  $\delta = \varepsilon/n$  guarantees that with probability  $1 - \varepsilon$  all the coordinates of  $Y_t$  are  $\varepsilon/n$  close to  $\{-1, 1\}$ . Then a simple argument shows that rounding the coordinates gives a PRG with error  $O(\varepsilon)$ , as desired.

We now state our main theorem.

► **Theorem 2** (Main theorem, informal version of Theorem 12). *Let  $\mathcal{F}$  be a family of  $n$ -variate Boolean functions that is closed under restrictions. Let  $X \in [-1, 1]^n$  be a symmetric  $p$ -noticeable fractional PRG for  $\mathcal{F}$  with error  $\varepsilon$ . Set  $t = O(\log(n/\varepsilon)/p)$  and let  $X_1, \dots, X_t$  be i.i.d. copies of  $X$ . Define the following random variables taking values in  $[-1, 1]^n$ :*

$$Y_0 = \bar{0}; \quad Y_i = Y_{i-1} + \delta_{Y_{i-1}} \circ X_i \quad i = 1, \dots, t.$$

*Let  $G = \text{sign}(Y_t) \in \{-1, 1\}^n$  obtained by taking the sign of the coordinates in  $Y_t$ . Then  $G$  is a PRG for  $\mathcal{F}$  with error  $(t + 1)\varepsilon$ .*

## 1.4 PRG for functions with bounded Fourier tails

As mentioned above, the families of Boolean functions that are fooled by our PRG include ones that satisfy the following two properties: (i) being closed under restrictions; (ii) having bounded  $L_1$  Fourier tails. Tal [20] showed that the latter condition follows from a widely studied condition, that of bounded  $L_2$  Fourier tails. Thus, using existing bounds for  $L_2$  Fourier tails, we get that our PRG fools several classes of Boolean functions. Below we list the results for error  $\varepsilon = O(1)$ , and refer the reader to the corresponding claims for the details of the full range of parameters:

1. **Functions of sensitivity  $s$** : seed length  $O(s^3 \log \log n)$ . The best previous construction [9] required seed length sub-exponential in  $s$  (concretely, their dependence on  $s$  is  $\exp(\sqrt{s})$ ). See Corollary 24 for details.
2. **Unordered read-once branching programs of width  $w$** : seed length  $O(\log^{2w+1} n \cdot \log \log n)$ . This is quadratically worse than the best known PRG [5]. However, our PRG construction does not utilize the branching program structure at all, except to obtain the Fourier tail bounds. See Corollary 25 for details.
3. **Permutation unordered read-once branching programs of width  $w$** : seed length  $O(w^4 \log n \cdot \log \log n)$ . This improves the dependence on  $n$  quadratically compared to the previous best PRG [18]. See Corollary 26 for details.
4. **Bounded depth circuits**: if  $f$  is computed by  $\text{AC}^0$  circuits of depth  $d$  and size  $\text{poly}(n)$ , our PRG has seed length  $O(\log^{2d-1} n \cdot \log \log n)$ . This is quadratically worse than the best known PRG [20]. See Corollary 27 for details.

Other than the PRG for functions of low sensitivity, all the other PRGs are comparable to the best known tailored PRG. However, the main message is that **they are all the same PRG**. Our general theorem is the following.

► **Theorem 3** (PRG for functions of bounded  $L_1$  Fourier tail, informal version of Theorem 23). *Let  $\mathcal{F}$  be a family of  $n$ -variate Boolean functions closed under restrictions. Assume that there exist  $a, b \geq 1$  such that for every  $f \in \mathcal{F}$ ,*

$$\sum_{S \subset [n]: |S|=k} |\hat{f}(S)| \leq a \cdot b^k.$$

*Then, for any  $\varepsilon > 0$  there exists an explicit PRG  $X \in \{-1, 1\}^n$  which fools  $\mathcal{F}$  with error  $\varepsilon > 0$ , whose seed length is  $O(\log(n/\varepsilon)(\log \log n + \log(a/\varepsilon))b^2)$ .*

We note again that by [20], Theorem 3 holds also if we instead assume a bound on the  $L_2$  Fourier tails (which are more common), namely if we assume that for every  $f \in \mathcal{F}$  it holds that

$$\sum_{S \subset [n]: |S| \geq k} \hat{f}(S)^2 \leq a \cdot 2^{-k/b}.$$

## 1.5 PRG for functions which simplify under random restriction

A major component in prior constructions of PRGs that are based on random restrictions is finding a much smaller set of ‘pseudorandom restrictions’. Ajtai and Wigderson [1] proposed such a PRG for low depth circuits based on Håstad’s switching lemma [8]. Many follow-up works are based on this framework to build PRGs for various classes of functions including low depth circuits, branching programs, low-sensitivity functions [21, 6, 18, 5, 9], and a major component of the analysis is proving that the derandomized random restrictions work.

Our framework for constructing PRGs directly applies to function families that simplify under random restrictions without the need to derandomize the restrictions. Let  $\mathcal{F}$  be a family of functions  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  which are extended multilinearly to  $[-1, 1]^n$ . Fix a parameter  $0 < p < 1$  and define the  $p$ -averaged function of  $f$ , denoted  $f_p : \{-1, 1\}^n \rightarrow [-1, 1]$ , as follows: sample  $A \subset [n]$  where  $\Pr[i \in A] = p$  independently for  $i \in [n]$ , and define

$$f_p(x) = \mathbb{E}_{A, U} [f(x_A, U_{A^c})]$$

where  $x_A \in \{-1, 1\}^A$  is the restriction of the input  $x$  to the coordinates in  $A$ , and  $U \in \{-1, 1\}^n$  is independently and uniformly chosen. The crucial observation (Claim 28) is that for every  $x \in \{-1, 1\}^n$  it holds that

$$f(px) = f_p(x).$$

Suppose now we have a standard PRG  $X$  for the class of  $p$ -averaged functions  $\mathcal{F}_p = \{f_p : f \in \mathcal{F}\}$ . Note a PRG for the  $p$ -random restriction of functions in  $\mathcal{F}$  would do, as  $f_p$  is a convex combination of  $p$ -random restrictions of  $f$  (namely, averaging over  $U$ ). Then, using our observation above, this implies that  $X' = pX$  is a fractional PRG for the class  $\mathcal{F}$ . Now by using our framework of viewing this fractional PRG as a random walk step, one can derive a standard PRG for  $\mathcal{F}$  using  $O(\log(1/\varepsilon)/p^2)$  independent copies of  $X$ .

## 1.6 Fourier tails of low degree $\mathbb{F}_2$ polynomials

Viola [22] gave a construction of a pseudorandom generator which fools  $n$ -variate polynomials over  $\mathbb{F}_2$ . The construction is the XOR of  $d$  independent small-bias generators. We wonder whether our framework can be used to achieve similar bounds. In particular, we raise the following problem: does the class of low-degree polynomials over  $\mathbb{F}_2$  have bounded  $L_1$  Fourier tails? It's trivially true for  $d = 1$  and it can be shown to hold for  $d = 2$ . However, to the best of our knowledge nothing was known for  $d \geq 3$ .

We show (see Theorem 29 for more details) that for any Boolean function  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  computed by a  $\mathbb{F}_2$ -polynomial of degree at most  $d$ , the following  $L_1$  Fourier tail bound holds:

$$\sum_{|S|=k} |\widehat{f}(S)| \leq k^k 2^{3dk} \quad \forall k = 1, \dots, n.$$

This bound however falls short of implying a PRG using our techniques, and we conjecture that the correct bound is  $c_d^k$ , for some constant  $c_d = 2^{O(d)}$ .

## 1.7 PRGs with respect to arbitrary product distributions

We note the following interesting generalization of our results that is almost direct from our techniques. Consider the problem of ‘fooling’ a family of functions with respect to an arbitrary product distribution  $D$  on  $\{-1, 1\}^n$  (the uniform distribution being a special case). More formally, given a distribution  $D$  on  $\{-1, 1\}^n$  and a family of functions  $\mathcal{F}$ , we say that a random variable  $X$  is a PRG for  $\mathcal{F}$  (with respect to  $D$ ) if  $|\mathbb{E}[f(D)] - \mathbb{E}[f(X)]| \leq \epsilon$ .

We show a way to fool functions with respect to arbitrary product distributions.

► **Corollary 4.** *Let  $\mathcal{F}$  be a family of  $n$ -variate Boolean functions which is closed under restrictions and let  $D$  be any product distribution on  $\{-1, 1\}^n$ . Let  $X \in [-1, 1]^n$  be a symmetric  $p$ -noticeable fractional PRG for  $\mathcal{F}$  with error  $\epsilon$  and seed length  $\ell$ . Let  $t = O(\log(n/\epsilon)/p)$ . Then there exists an explicit PRG for  $\mathcal{F}$  with respect to  $D$  with error  $t\epsilon$  and seed length  $t\ell$ .*

**Proof sketch.** If  $D$  is a product distribution on  $\{-1, 1\}^n$ , then  $\mathbb{E}[f(D)] = f(\alpha)$ , where  $\alpha = \mathbb{E}[D] \in [-1, 1]^n$ . Thus, we now start our random walk (defined by the fractional PRG) from the point  $\alpha$  instead of from  $\bar{0}$ , and the convergence follows from polarization in exactly the same way. ◀

Thus all our PRG results in fact generalize to PRGs with respect to arbitrary product distributions. To the best of our knowledge, we are not aware of any non-trivial PRGs against arbitrary product distributions for the classes of functions we study. We wonder if this notion of fooling arbitrary product distributions has interesting applications.

## 1.8 Related works

The line of research closest in spirit to our work, and which motivated our work, is that of using random and pseudo-random restrictions to construct PRGs. A good example is [6] which uses pseudo-random restrictions to construct PRGs. Our framework can be seen as extending this, as we do not need to analyze pseudo-random restrictions; instead, we analyze fractional PRGs, where the restriction happens automatically from the fractional PRG structure, and no derandomization is necessary.



Another line of work is the use of random walks in combinatorial optimization, for example in the algorithmic versions of Spencer’s theorem [3, 12] and follow up works. It would be interesting to see if polarization can be used to speed up random walks in combinatorial optimization as well.

## 1.9 Open problems

As we give a new framework for constructing PRGs, there are many open problems that arise, both conceptual and technical.

### 1.9.1 Early termination

Our analysis requires a random walk with  $t = O(\log(n/\varepsilon)/p)$  steps, each coming from a  $p$ -noticeable fractional PRG. We believe that for some natural families of functions shorter random walks might also suffice, but we do not know how to show this. We discuss this further in Section 7.

► **Open problem 5.** *Find conditions on classes of Boolean functions so that short random walks can be used to construct PRGs. In particular, are there nontrivial classes where the number of steps is independent of  $n$ ?*

### 1.9.2 Less independence

Our analysis of Theorem 12 currently requires to assume  $t$  independent copies of a fractional PRG  $X$ . It might be possible that they copies can be chosen in a less independent form, where the analysis still holds.

► **Open problem 6.** *Can the fractional PRGs  $X_1, \dots, X_t$  in Theorem 12 be chosen not independently, such that the conclusion still holds? Concrete examples to consider are  $k$ -wise independence for  $k \ll t$ , or using an expander random walk.*

### 1.9.3 More applications

Our current applications follow from the construction of a fractional PRG for functions with bounded Fourier tails. The fractional PRG itself follows from standard constructions in pseudo-randomness (almost  $k$ -wise independent) adapted to our scenario. It will be interesting to try and find other classes of Boolean functions for which different constructions of fractional PRG work.

### 1.9.4 Gadgets

We can view the random walk as a “gadget construction”. Given independent  $p$ -noticeable fractional PRGs  $X_1, \dots, X_t \in [-1, 1]^n$ , view them as the rows of a  $t \times n$  matrix, and then apply a gadget  $g : [-1, 1]^t \rightarrow \{-1, 1\}$  to each column to obtain the outcome in  $\{-1, 1\}^n$ . We show that the random walk gives such a gadget which converges for  $t = O(\log(n/\varepsilon)/p)$ . Many constructions of PRGs can be viewed in this framework, where typically  $X_i \in \{-1, 1\}^n$ . Ours is the first construction which allows  $X_i$  to take non-Boolean values. It is interesting whether other gadgets can be used instead of the random walk gadget, and whether there are general properties of gadgets that would suffice.

### 1.9.5 Low degree polynomials

As discussed above, we wonder if our techniques can be used to construct a PRG for low degree  $\mathbb{F}_2$  polynomials. In particular, we ask if one could improve the bounds we obtain (see Theorem 29) on the  $L_1$  Fourier tails of low degree  $\mathbb{F}_2$  polynomials.

► **Open problem 7.** Let  $f = (-1)^p$  where  $p : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  is a polynomial of degree  $d$ . Is there a constant  $c_d$  such that  $\sum_{S:|S|=k} |\hat{f}(S)| \leq c_d^k$  which is independent of  $n$ ? In particular, we conjecture that  $c_d = 2^{O(d)}$  should work.

Note that the exponential dependence on  $k$  is needed, as witnessed from the following example: consider the quadratic  $\mathbb{F}_2$  polynomial  $q(x) = \sum_{i=1}^{n/2} x_{2i-1}x_{2i}$ . Then  $(-1)^q$  has Fourier  $L_1$  weight  $\binom{n}{n/2} \cdot 2^{-n/2} = 2^{\Omega(n)}$  on the  $(n/2)$ -th level.

### 1.10 Paper organization

We describe the general framework in detail in Section 2. We prove Theorem 12 in Section 3. We describe applications in Section 4. Our framework also applies to function families that simplify under random restrictions. We describe this in Section 5. We prove  $L_1$  Fourier tail bounds for low degree  $\mathbb{F}_2$  polynomials in Section 6. We try to partially answer the question related to early termination of the random walk in Section 7.

## 2 General framework

### 2.1 Boolean functions

Let  $f : \{-1, 1\}^n \rightarrow [-1, 1]$  be an  $n$ -variate Boolean function, identified with its multilinear extension, also known as its Fourier expansion. For  $x \in [-1, 1]^n$  define  $f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{i \in S} x_i$ . As  $f$  is multilinear, a convenient viewpoint is to view  $f(x)$  as computing the expected value of  $f$  on a product distribution on  $\{-1, 1\}^n$ . That is, let  $W = W(x) \in \{-1, 1\}^n$  be a random variable, where  $W_1, \dots, W_n$  are independently chosen so that  $\mathbb{E}[W_i] = x_i$ . Then  $f(x) = \mathbb{E}f(W)$ . In particular,  $f(\bar{0}) = \mathbb{E}f(U)$ , where  $U \in \{-1, 1\}^n$  is uniformly chosen.

A family  $\mathcal{F}$  of  $n$ -variate Boolean functions is said to be *closed under restrictions* if for any  $f \in \mathcal{F}$  and any function  $f' : \{-1, 1\}^n \rightarrow \{-1, 1\}$  obtained from  $f$  by fixing some of its inputs to  $\{-1, 1\}$  it holds that also  $f' \in \mathcal{F}$ .

### 2.2 Pseudorandom generators

Let  $\mathcal{F}$  be a family of  $n$ -variate Boolean functions. The following is the standard definition of a pseudorandom generator (PRG) for  $\mathcal{F}$ , adapted to our notation.

► **Definition 8 (PRG).** A random variable  $X \in \{-1, 1\}^n$  is a PRG for  $\mathcal{F}$  with error  $\varepsilon$ , if for any  $f \in \mathcal{F}$  it holds that  $|f(\bar{0}) - \mathbb{E}f(X)| \leq \varepsilon$ .

We introduce the notion of a *fractional PRG*. It is the same as a PRG, except that the random variable is allowed to take values in  $[-1, 1]^n$ , instead of only Boolean values.

► **Definition 9 (Fractional PRG).** A random variable  $X \in [-1, 1]^n$  is a fractional PRG for  $\mathcal{F}$  with error  $\varepsilon$ , if for any  $f \in \mathcal{F}$  it holds that  $|f(\bar{0}) - \mathbb{E}f(X)| \leq \varepsilon$ .

Our main goal will be to “amplify” fractional PRGs for  $\mathcal{F}$  in order to obtain PRGs for  $\mathcal{F}$ . To that end, we need to enforce some non-triviality conditions on the fractional PRG. For example,  $X = \bar{0}$  is a fractional PRG for any function. We require that for any coordinate  $i \in [n]$ , the value of  $X_i$  is far from zero with noticeable probability. Formally, we require a noticeable second moment.

► **Definition 10** (*p*-noticeable random variable). A random variable  $X \in [-1, 1]^n$  is *p*-noticeable if for every  $i \in [n]$ ,  $\mathbb{E}[X_i^2] \geq p$ .

For technical reasons, we would also need  $X$  to be *symmetric*, which means that the distribution of  $-X$  is the same as the distribution of  $X$ . This is easy to achieve, for example by multiplying all elements of  $X$  with a uniformly chosen sign.

### 2.3 Polarizing random walks

The main idea is to view a fractional PRG as steps in a random walk in  $[-1, 1]^n$  that converges to  $\{-1, 1\}^n$ . To that end, we define a gadget that implements the random walk; and moreover, that allows for fast convergence. As we will see later, the fast convergence is an effect of polarization.

► **Definition 11** (Random walk gadget). For any  $t \geq 1$  define the random walk gadget  $g_t : [-1, 1]^t \rightarrow [-1, 1]$  as follows. Let  $a_1, \dots, a_t \in [-1, 1]$ . Define  $g_1(a_1) := a_1$  and for  $t > 1$ ,

$$g_t(a_1, \dots, a_t) := g_{t-1}(a_1, \dots, a_{t-1}) + (1 - |g_{t-1}(a_1, \dots, a_{t-1})|)a_t.$$

We extend the definition to act on bit-vectors. Define  $g_t^n : ([-1, 1]^n)^t \rightarrow [-1, 1]^n$  as follows. For  $x_1, \dots, x_t \in [-1, 1]^n$  define

$$g_t^n(x_1, \dots, x_t) = (g_t(x_{1,1}, \dots, x_{t,1}), \dots, g_t(x_{1,n}, \dots, x_{t,n})).$$

Equivalently, we can view  $g_t^n$  as follows: construct a  $t \times n$  matrix whose rows are  $x_1, \dots, x_t$ ; and then apply  $g_t$  to each column of the matrix to obtain a resulting vector in  $[-1, 1]^n$ .

The following theorem shows how to “amplify” fractional PRGs using the random walk gadget to obtain a PRG. Below, for  $x \in [-1, 1]^n$  we denote by  $\text{sign}(x) \in \{-1, 1\}^n$  the Boolean vector obtained by taking the sign of each coordinate (the sign of 0 can be chosen arbitrarily).

► **Theorem 12** (Amplification Theorem). *Let  $\mathcal{F}$  be a family of  $n$ -variate Boolean functions which is closed under restrictions. Let  $X \in [-1, 1]^n$  be a symmetric  $p$ -noticeable fractional PRG for  $\mathcal{F}$  with error  $\varepsilon$ . Set  $t = O(\log(n/\varepsilon)/p)$  and let  $X_1, \dots, X_t$  be iid copies of  $X$ . Define a random variable  $G \in \{-1, 1\}^n$  as follows:*

$$G := G(X_1, \dots, X_t) = \text{sign}(g_t^n(X_1, \dots, X_t)).$$

*Then  $G$  is a PRG for  $\mathcal{F}$  with error  $(t + 1)\varepsilon$ .*

## 3 Proof of Amplification Theorem

We prove Theorem 12 in this section. From here onwards, we fix a family  $\mathcal{F}$  of  $n$ -variate Boolean functions which is closed under restrictions. The proof is based on the following two lemmas. The first lemma amplifies a  $p$ -noticeable fractional PRG to a  $(1 - q)$ -noticeable fractional PRG. The second lemma shows that setting  $q = \varepsilon/n$ , the latter fractional PRG can be rounded to a Boolean-valued PRG without incurring too much error.

► **Lemma 13** (Amplification lemma). *Let  $X_1, \dots, X_t \in [-1, 1]^n$  be independent symmetric  $p$ -noticeable fractional PRGs for  $\mathcal{F}$  with error  $\varepsilon$ . Define a random variable  $Y \in [-1, 1]^n$  as*

$$Y := g_t^n(X_1, \dots, X_t).$$

*Then  $Y$  is a  $(1 - q)$ -noticeable fractional PRG for  $\mathcal{F}$  with error  $t\varepsilon$ , where  $q = 2^{-\Omega(pt)}$ .*

► **Lemma 14** (Rounding lemma). *Let  $Y \in [-1, 1]^n$  be a  $(1 - q)$ -noticeable fractional PRG for  $\mathcal{F}$  with error  $\varepsilon$ . Then  $\text{sign}(Y) \in \{-1, 1\}^n$  is a PRG for  $\mathcal{F}$  with error  $\varepsilon + qn$ .*

Theorem 12 follows directly by applying Lemma 13 with  $t = O(\log(n/\varepsilon)/p)$  to obtain  $q = \varepsilon/n$  and then applying Lemma 14.

### 3.1 Proof of Lemma 13

We prove Lemma 13 in this section. We need to prove two claims: that  $g_t^n(X_1, \dots, X_t)$  is a fractional PRG for  $\mathcal{F}$  with error  $\varepsilon t$ , and that it is  $(1 - q)$ -noticeable. This is achieved in the following sequence of claims.

First we need some notations. For  $y \in [-1, 1]^n$  define  $\delta_y \in [-1, 1]^n$  by  $(\delta_y)_i := 1 - |y_i|$ . For two vectors  $x, y \in [-1, 1]^n$  define  $x \circ y \in [-1, 1]^n$  to be their pointwise product, namely  $(x \circ y)_i := x_i y_i$ . Observe that  $\{y + \delta_y \circ x : x \in [-1, 1]^n\}$  is the largest symmetric sub-cube in  $[-1, 1]^n$  centered at  $y$ .

► **Claim 15.** *Let  $X \in [-1, 1]^n$  be a fractional PRG for  $\mathcal{F}$  with error  $\varepsilon$ . Then for any  $f \in \mathcal{F}$  and any  $y \in [-1, 1]^n$ ,*

$$|f(y) - \mathbb{E}f(y + \delta_y \circ X)| \leq \varepsilon.$$

**Proof.** Consider a distribution over  $F \in \mathcal{F}$  obtained from  $f$  by fixing the  $i$ -th input to  $\text{sign}(y_i)$  with probability  $|y_i|$ , independently for each  $i$ . That is,

$$F(x) := f(R(x)),$$

where  $R(x) \in \{-1, 1\}^n$  is a random variable obtained by sampling  $R_1, \dots, R_n$  independently where  $\Pr[R_i = \text{sign}(y_i)] = |y_i|$  and  $\Pr[R_i = x_i] = 1 - |y_i|$ . By the multi-linearity of  $f$ , and as  $R(x)$  is a product distribution,

$$\mathbb{E}_F[F(x)] = \mathbb{E}_R[f(R(x))] = f(\mathbb{E}_R[R(x)]) = f(y + \delta_y \circ x).$$

Setting  $x = X$  and averaging over  $X$  gives

$$|f(y) - \mathbb{E}_X[f(y + \delta_y \circ X)]| = |\mathbb{E}_F F(\bar{0}) - \mathbb{E}_{F,X}[F(X)]| \leq \mathbb{E}_F |F(\bar{0}) - \mathbb{E}_X[F(X)]| \leq \varepsilon,$$

since  $F \in \mathcal{F}$  with probability one and  $X$  is a fractional PRG for  $\mathcal{F}$  with error  $\varepsilon$ . ◀

► **Claim 16.** *Let  $X_1, \dots, X_t \in [-1, 1]^n$  be independent fractional PRGs for  $\mathcal{F}$  with error  $\varepsilon$ . Then for any  $f \in \mathcal{F}$ ,*

$$|f(\bar{0}) - \mathbb{E}_{X_1, \dots, X_t}[f(g_t^n(X_1, \dots, X_t))]| \leq t\varepsilon.$$

**Proof.** The proof is by induction on  $t$ . The base case  $t = 1$  follows by definition as  $g_1^n(X_1) = X_1$ . For  $t > 1$  we will show that

$$|\mathbb{E}[f(g_{t-1}^n(X_1, \dots, X_{t-1}))] - \mathbb{E}[f(g_t^n(X_1, \dots, X_t))]| \leq \varepsilon,$$

from which the claim follows by the triangle inequality. In fact, we will show a stronger inequality: for any fixing of  $x_1, \dots, x_{t-1} \in [-1, 1]^n$ , it holds that

$$|f(g_{t-1}^n(x_1, \dots, x_{t-1})) - \mathbb{E}_{X_t}[f(g_t^n(x_1, \dots, x_{t-1}, X_t))]| \leq \varepsilon.$$

The first inequality then follows by averaging over  $x_1 = X_1, \dots, x_{t-1} = X_{t-1}$ . To see why this latter inequality holds, set  $y = g_{t-1}^n(x_1, \dots, x_{t-1})$ . Then by definition,

$$g_t^n(x_1, \dots, x_{t-1}, X_t) = y + \delta_y \circ X_t.$$

The claim now follows from Claim 15. ◀

We have so far proved that  $g_t^n(X_1, \dots, X_t)$  is a fractional PRG for  $\mathcal{F}$  with slightly worse error. Although we do not need it, it is worth noting that it is symmetric since  $X_1, \dots, X_t$  are symmetric and  $-g_t^n(X_1, \dots, X_t) = g_t^n(-X_1, \dots, -X_t)$ . To conclude, we show that it converges fast to a value close to  $\{-1, 1\}^n$ . This is the effect of *polarization*. It will be enough to analyze this for one-dimensional random variables.

► **Claim 17.** *Let  $A_1, \dots, A_t \in [-1, 1]$  be independent symmetric random variables with  $\mathbb{E}[A_i^2] \geq p$ . For  $i = 1, \dots, t$  define*

$$B_i := g_i(A_1, \dots, A_i) = B_{i-1} + (1 - |B_{i-1}|)A_i.$$

Then  $\mathbb{E}[B_t^2] \geq 1 - q$  where  $q = 3 \exp(-tp/8)$ .

**Proof.** Let  $C_i := 1 - |B_i|$  be the distance to  $\{-1, 1\}$  at step  $i$ . We show that  $C_i$  converges to 0 exponentially fast. Observe that  $C_i$  satisfies the following recursive definition:

$$C_i = \begin{cases} C_{i-1}(1 - A_i) & \text{if } C_{i-1}(1 - A_i) \leq 1 \\ 2 - C_{i-1}(1 - A_i) & \text{if } C_{i-1}(1 - A_i) > 1 \end{cases}.$$

In either case one can verify that  $C_i \in [0, 1]$  and that

$$C_i \leq C_{i-1}(1 - A_i).$$

As  $C_{i-1}$  and  $A_i$  are independent we obtain that

$$\mathbb{E}[\sqrt{C_i}] = \mathbb{E}[\sqrt{C_{i-1}}] \mathbb{E}[\sqrt{1 - A_i}].$$

We now use the assumption that the  $A_i$  are symmetric. The Taylor expansion of  $\sqrt{1 - x}$  in  $[-1, 1]$  is

$$\sqrt{1 - x} = 1 - \frac{x}{2} - \frac{x^2}{8} - \frac{x^3}{16} - \dots$$

In particular, all the coefficients except for the constant term are negative. As  $A_i$  is symmetric,  $\mathbb{E}[A_i^k] = 0$  for any odd  $k$ , so

$$\mathbb{E}[\sqrt{1 - A_i}] \leq 1 - \frac{\mathbb{E}[A_i^2]}{8} \leq 1 - \frac{p}{8} \leq \exp(-p/8).$$

Thus

$$\mathbb{E}[\sqrt{C_t}] \leq \prod_{i=1}^t \mathbb{E}[\sqrt{1 - A_i}] \leq \exp(-tp/8).$$

By Markov's inequality,  $\Pr[C_t \geq \exp(-tp/2)] \leq \exp(-tp/8)$ . If  $C_t \leq \exp(-tp/2)$  then  $1 - B_t^2 \leq 2\exp(-tp/2)$ . If not, then we can trivially bound  $1 - B_t^2 \leq 1$ . Putting these together gives

$$\mathbb{E}[1 - B_t^2] \leq 2\exp(-tp/2) + \exp(-tp/8) \leq 3\exp(-tp/8). \quad \blacktriangleleft$$

To provide a piece of intuition explaining the fast convergence of this random walk, notice that once  $C_i$  becomes sufficiently small, it gets more and more difficult to increase the value of  $C_i$  again. This could be best explained with an example. Suppose all  $A_i$ 's take value in  $\{-0.5, 0.5\}$ . We start at  $B_0 = 0$  and take a step, say  $A_1 = 0.5$ , and therefore  $B_1 = 0.5$ . Now observe that the length of the next step would be only  $(1 - |B_1|)|A_2| = 0.25$ . So even if  $A_2 = -0.5$ , we get  $B_2 = 0.25$ , which means we still need to take one more step to become less than 0. In other words, once we get close to the boundary  $\{-1, 1\}$ , the random walk converges faster as it gets more difficult to move away from the boundary.

► **Corollary 18.** *Let  $X_1, \dots, X_t \in [-1, 1]^n$  be independent symmetric  $p$ -noticeable random variables. Define  $Y = g_t^n(X_1, \dots, X_t)$ . Then  $Y$  is  $(1 - q)$ -noticeable for  $q = 3\exp(-tp/8)$ .*

**Proof.** Apply Claim 17 to each coordinate of  $Y$ . ◀

Lemma 13 follows by combining Claim 16 and Corollary 18.

### 3.2 Proof of Lemma 14

We prove Lemma 14 in this section. Let  $x \in [-1, 1]^n$  be a potential value obtained by  $X$ . Let  $W := W(x) \in \{-1, 1\}^n$  be a random variable, where  $W_1, \dots, W_n$  are independent and  $\mathbb{E}[W_i] = x_i$ . Then  $\mathbb{E}_W[f(W)] = f(x)$ . As  $f$  takes values in  $[-1, 1]$ , we can upper bound  $|f(x) - f(\text{sign}(x))|$  by

$$|f(x) - f(\text{sign}(x))| = |\mathbb{E}_W[f(W)] - f(\text{sign}(x))| \leq \Pr[W \neq \text{sign}(x)].$$

The last term can be bounded by the union bound,

$$\Pr[W \neq \text{sign}(x)] \leq \sum_{i=1}^n \Pr[W_i \neq \text{sign}(x_i)] = \frac{1}{2} \sum_{i=1}^n (1 - |x_i|).$$

Setting  $x = X$  and averaging over  $X$  gives

$$|\mathbb{E}_X[f(X)] - \mathbb{E}_X[f(\text{sign}(X))]| \leq \mathbb{E}_X |f(X) - f(\text{sign}(X))| \leq \frac{1}{2} \sum_{i=1}^n \mathbb{E}[1 - |X_i|].$$

As  $X$  is  $(1 - q)$ -noticeable it satisfies  $\mathbb{E}[X_i^2] \geq 1 - q$  for all  $i$ . As  $1 - z \leq 1 - z^2$  for all  $z \in [0, 1]$  we have

$$\mathbb{E}[1 - |X_i|] \leq \mathbb{E}[1 - X_i^2] \leq q.$$

This concludes the proof as

$$|f(\bar{0}) - \mathbb{E}_X[f(\text{sign}(X))]| \leq |f(\bar{0}) - \mathbb{E}_X[f(X)]| + |\mathbb{E}_X[f(X)] - \mathbb{E}_X[f(\text{sign}(X))]| \leq \varepsilon + qn,$$

where the first inequality follows as  $X$  is a fractional PRG with error  $\varepsilon$ , and the second by the discussion above.

#### 4 PRGs for functions with bounded Fourier tails

Several natural families of Boolean functions have bounded Fourier tails, such as:  $AC^0$  circuits [11, 14]; functions with bounded sensitivity [7, 13]; and functions computed by branching programs of various forms [18, 5]. Our goal is to construct a universal PRG which fools any such function. We consider two variants:  $L_1$  bounds and  $L_2$  bounds.

► **Definition 19** ( $L_1$  bounds). For  $a, b \geq 1$ , we denote by  $\mathcal{L}_1^n(a, b)$  the family of  $n$ -variate Boolean functions  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  which satisfy

$$\sum_{\substack{S \subseteq [n] \\ |S|=k}} |\widehat{f}(S)| \leq a \cdot b^k \quad \forall k = 1, \dots, n.$$

► **Definition 20** ( $L_2$  bounds). For  $a, b \geq 1$ , we denote by  $\mathcal{L}_2^n(a, b)$  the family of  $n$ -variate Boolean functions  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  which satisfy

$$\sum_{\substack{S \subseteq [n] \\ |S| \geq k}} \widehat{f}(S)^2 \leq a \cdot 2^{-k/b} \quad \forall k = 1, \dots, n.$$

Tal [20] showed that  $L_2$  bounds imply  $L_1$  bounds: if  $f \in \mathcal{L}_2(a, b)$  then  $f \in \mathcal{L}_1(a, b')$  for  $b' = O(b)$ . The reverse direction is false, as can be witnessed by the PARITY function. So, the class of functions with  $L_1$  bounded Fourier tails is richer, and we focus on it.

In the following lemma, we construct a fractional PRG for this class, which we will then amplify to a PRG. We note that this lemma holds also for bounded functions, not just Boolean functions. The construction is based on a scaling of almost  $d$ -wise independent random variables, whose definition we now recall.

► **Definition 21** (Almost  $d$ -wise independence). A random variable  $Z \in \{-1, 1\}^n$  is  $\varepsilon$ -almost  $d$ -wise independent if, for any restriction of  $Z$  to  $d$  coordinates, the marginal distribution has statistical distance at most  $\varepsilon$  from the uniform distribution on  $\{-1, 1\}^d$ .

Naor and Naor [15] gave an explicit construction of an  $\varepsilon$ -almost  $d$ -wise random variable  $Z \in \{-1, 1\}^n$  with seed length  $O(\log \log n + \log d + \log(1/\varepsilon))$ . We note that this seed length is optimal, up to the hidden constants.

► **Lemma 22.** Fix  $n, a, b \geq 1$  and  $\varepsilon > 0$ . There exists a fractional PRG  $X \in [-1, 1]^n$  that fools  $\mathcal{L}_1^n(a, b)$  with error  $\varepsilon$ , such that

- (i)  $X$  is  $p$ -noticeable for  $p = \frac{1}{4b^2}$ .
- (ii) The seed length of  $X$  is  $O(\log \log n + \log(a/\varepsilon))$ .

**Proof.** Fix  $f \in \mathcal{L}_1^n(a, b)$ . Set  $d = \lceil \log 2a/\varepsilon \rceil$ ,  $\delta = \varepsilon/2a$ ,  $\beta = 1/2b$ . Let  $Z \in \{-1, 1\}^n$  be an  $\delta$ -almost  $d$ -wise independent random variable, and set  $X = \beta Z$  which takes values in  $\{-\beta, \beta\}^n$ . We claim that  $X$  satisfies the requirements of the lemma. Claim (i) clearly holds, and claim (ii) holds by the Naor-Naor construction. We thus focus on proving that  $X$  fools  $\mathcal{F}$  with error  $\varepsilon$ .

Fix  $f \in \mathcal{F}$  and consider its Fourier expansion:

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) x^S.$$

We need to show that  $\mathbb{E}[f(X)]$  is close to  $f(\bar{0})$ . Averaging over  $X$  gives

$$|\mathbb{E}[f(X)] - f(\bar{0})| \leq \sum_{|S|>0} |\widehat{f}(S)| \cdot |\mathbb{E}[X^S]| = \sum_{|S|>0} |\widehat{f}(S)| \cdot \beta^{|S|} |\mathbb{E}[Z^S]|.$$

We next bound  $|\mathbb{E}[Z^S]|$ . If  $|S| \leq d$  then by the definition of  $Z$  we have  $|\mathbb{E}[Z^S]| \leq \delta$ . If  $|S| > d$  we bound trivially  $|\mathbb{E}[Z^S]| \leq 1$ . Let  $W_k = \sum_{S:|S|=k} |\hat{f}(S)|$ , where by assumption  $W_k \leq a \cdot b^k$ . Thus

$$|\mathbb{E}[f(X)] - f(\bar{0})| \leq \delta \sum_{k=1}^d W_k \beta^k + \sum_{k>d} W_k \beta^k \leq \delta a \sum_{k=1}^d (\beta b)^k + a \sum_{k>d} (\beta b)^k \leq \delta a + 2^{-d} a$$

where we used the choice of  $\beta = 1/2b$ . The claim follows as we set  $\delta = \varepsilon/2a$  and  $2^{-d} \leq \varepsilon/2a$ .  $\blacktriangleleft$

Applying Theorem 12 using the fractional PRG constructed in Lemma 22 gives the following PRG construction. Note that we still need to require that  $\mathcal{F}$  is closed under restrictions.

► **Theorem 23.** *Let  $\mathcal{F}$  be a family of  $n$ -variate Boolean functions closed under restrictions. Assume that  $\mathcal{F} \subset \mathcal{L}_1^n(a, b)$  or that  $\mathcal{F} \subset \mathcal{L}_2^n(a, b)$ . Then, for any  $\varepsilon > 0$  there exists an explicit PRG  $X \in \{-1, 1\}^n$  which fools  $\mathcal{F}$  with error  $\varepsilon > 0$ , whose seed length is  $O(\log(n/\varepsilon)(\log \log n + \log(a/\varepsilon)b^2))$ .*

## 4.1 Applications

We apply our PRG from Theorem 23 to several well studied classes of Boolean functions that are known to satisfy a Fourier tail bound.

### 4.1.1 Functions of bounded sensitivity

Let  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  be a Boolean function. Its sensitivity at an input  $x \in \{-1, 1\}^n$  is the number of neighbors  $x'$  of  $x$  (that is,  $x'$  and  $x$  differ at exactly one coordinate) such that  $f(x') \neq f(x)$ . The (max) sensitivity of  $f$  is  $s(f) = \max_x s(f, x)$ . The sensitivity conjecture speculates that functions of sensitivity  $s$  can be computed by decision trees of depth  $\text{poly}(s)$ . A corollary would be that almost  $\text{poly}(s)$ -wise distributions fool functions of low sensitivity. So, one may ask to construct comparable PRGs for functions of low sensitivity.

This question was first considered by Hatami and Tal [9]. They constructed a PRG with sub-exponential seed length  $\exp(O(\sqrt{s}))$ . Theorem 23 gives an improved construction that essentially matches the consequence of the sensitivity conjecture. Our PRG uses the recent bounds of Gopalan et al. [7] on the Fourier tail of functions of low sensitivity. Concretely, Gopalan et al. [7] show that if  $s(f) = s$  then  $f \in \mathcal{L}_1(1, t)$  for  $t = O(s)$ . It is straightforward to verify that a restriction can only decrease the sensitivity of the function, so that the class of functions of sensitivity at most  $s$  is closed under restrictions. A direct application of Theorem 23 gives a PRG with seed length  $O(s^2 \log(n/\varepsilon)(\log \log(n) + \log(1/\varepsilon)))$ .

To get a somewhat improved bound, one can apply a result of Simon [19] that shows that if  $s(f) = s$  then  $f$  depends on at most  $m = 4^s$  many inputs. In this case, the analysis of Theorem 12 can be applied with  $m$  variables instead of  $n$  variables, so that we only need  $O(\log m/\varepsilon)$  iterations. Note that the fractional PRG still requires a seed length which depends on the original  $n$ . We obtain:

► **Corollary 24.** *For any  $n, s \geq 1$  and  $\varepsilon > 0$ , there exists an explicit PRG which fools  $n$ -variate Boolean functions with sensitivity  $s$  with error  $\varepsilon$ , whose seed length is  $O(s^3 \log(1/\varepsilon)(\log \log n + \log(1/\varepsilon)))$ .*

We note that the  $\log \log n$  term cannot be removed. Indeed, even if we restrict attention to functions which are XOR of at most 2 bits (for which  $s = 2$ ) the seed length required is  $\Omega(\log \log n + \log(1/\varepsilon))$ .



### 4.1.2 Unordered branching programs

An oblivious read-once branching program (abbrv ROBP)  $B$  of width  $w$  is a non-uniform model of computation, that captures randomized algorithms with space  $\log w$ . A branching program  $B$  maintains a state in the set  $\{1, \dots, w\}$  and reads the input bits in a known fixed order. At time step  $i = 1, \dots, n$ ,  $B$  reads a bit and based on the time step, the read bit and the current state it transitions to a new state. Thus,  $B$  can be thought of as a layered directed graph, with  $w$  nodes in each layer, and two edges going out of each node to the immediately next layer, one labeled with a 1 and the other labeled with a  $-1$ .

Let  $\mathcal{B}^n(w)$  be the class of  $n$ -variate Boolean functions computed by read-once oblivious branching programs of width  $w$ , where the order of the inputs is arbitrary. A recent work of Chattopadhyay et al. [5] showed that these functions have  $L_1$  bounded Fourier tails. Concretely,  $\mathcal{B}^n(w) \subset \mathcal{L}_1^n(t)$  for  $t = (\log n)^w$ . They used this to construct a PRG with seed length  $O(\log n)^{w-1} \log^2(n/\epsilon) \log \log n$ . Using our PRG from Theorem 23 we get a comparable (although slightly worse) seed length. Note that  $\mathcal{B}^n(w)$  is closed under restrictions.

► **Corollary 25.** *Fix  $n, w \geq 1$  and  $\epsilon > 0$ . There is an explicit PRG which fools  $\mathcal{B}^n(w)$  with error  $\epsilon > 0$ , whose seed length is  $O(\log(n/\epsilon)(\log \log n + \log 1/\epsilon)(\log n)^{2w})$ .*

### 4.1.3 Permutation branching programs

A special case of read-once branching programs are permutation branching programs, where the transition function from level  $i$  to level  $i + 1$  in the graph is a permutation for every choice of the input bit. We denote it by  $\mathcal{B}_{\text{perm}}^n(w) \subset \mathcal{B}^n(w)$ . Reingold et al. [18] showed that if a Boolean function is computed by a permutation branching program of width  $w$ , then it has  $L_2$  bounded Fourier tails with parameter  $2w^2$ . Note that permutation branching programs are also closed under restrictions. Thus we obtain the following result:

► **Corollary 26.** *Fix  $n, w \geq 1$  and  $\epsilon > 0$ . There is an explicit PRG which fools  $\mathcal{B}_{\text{perm}}^n(w)$  with error  $\epsilon > 0$ , whose seed length is  $O(\log(n/\epsilon)(\log \log n + \log 1/\epsilon)w^4)$ .*

The dependence on  $n$  in our PRG is better than in the previous work of [18], as they obtained seed length  $O(w^2 \log(w) \log(n) \log(nw/\epsilon) + w^4 \log^2(w/\epsilon))$ .

The work of [18] actually shows the Fourier tail bounds for a more general class of branching programs, called regular branching programs. However, these are not closed under restriction, and hence our PRG construction fails to work (the same problem occurs also in the construction of [18]).

### 4.1.4 Bounded depth circuits

The class of bounded-depth Boolean circuits  $\text{AC}^0$  has been widely studied. In particular, Linial, Mansour and Nisan [11] showed that it has bounded  $L_2$  Fourier tails. Tal [20] obtained improved bounds. If  $f$  is an  $n$ -variate Boolean function computed by an  $\text{AC}^0$  circuit of depth  $d$  and size  $s$ , then  $f \in \mathcal{L}_2(n, t)$  for  $t = 2^{O(d)} \log^{d-1} s$ . Theorem 23 provides a new PRG for  $\text{AC}^0$  which is comparable with the existing PRGs of Nisan [16] and Braverman [4].

► **Corollary 27.** *Fix  $n, s \geq 1$  and  $\epsilon > 0$ . There is an explicit PRG which fools  $n$ -variate functions which can be computed by  $\text{AC}^0$  circuits of size  $s$  and depth  $d$ , with error  $\epsilon > 0$ , whose seed length is  $O(\log(n/\epsilon)(\log \log n + \log 1/\epsilon) \log^{2d-2} s)$ .*

## 5 PRG for functions which simplify under random restriction

Another generic application of our framework is constructing PRGs for classes that simplify under random restriction. Let  $\mathcal{F}$  be a family of functions  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  which are extended multilinearly to  $[-1, 1]^n$ . Fix a parameter  $0 < p < 1$  and define the  $p$ -averaged function of  $f$ , denoted  $f_p : \{-1, 1\}^n \rightarrow [-1, 1]$  as follows: sample  $A \subset [n]$  where  $\Pr[i \in A] = p$  independently for  $i \in [n]$ , and define

$$f_p(x) = \mathbb{E}_A \mathbb{E}_U [f(x_A, U_{A^c})]$$

where  $x_A \in \{-1, 1\}^A$  is the restriction of the input  $x$  to the coordinates in  $A$ , and  $U \in \{-1, 1\}^n$  is independently and uniformly chosen.

► **Claim 28.**  $f_p(x) = f(px)$ .

**Proof.** Let  $A, U$  be random variables as defined above. Define a random variable  $Y \in \{-1, 1\}^n$  as follows:

$$Y_i = \begin{cases} x_i & \text{if } A_i = 1 \\ U_i & \text{if } A_i = 0 \end{cases}.$$

Note that  $Y$  is a product distribution. By definition of  $f_p$ ,  $f_p(x) = \mathbb{E}[f(Y)]$ . By multilinearity of  $f$ ,  $\mathbb{E}[f(Y)] = f(\mathbb{E}[Y]) = f(px)$ . ◀

Suppose that we have a standard PRG  $X$  for the class of  $p$ -averaged functions  $\mathcal{F}_p = \{f_p : f \in \mathcal{F}\}$ . Claim 28 implies that  $X' = pX$  is a fractional PRG for the class  $\mathcal{F}$ . Theorem 12 then constructs a PRG for  $\mathcal{F}$  using  $O(\log(1/\epsilon)/p^2)$  independent copies of  $X$ .

## 6 Spectral tail bounds for low degree $\mathbb{F}_2$ -polynomials

In this section, we prove  $L_1$  Fourier tail bounds for functions computed by low degree polynomials on  $\mathbb{F}_2$ . However, our bounds fall short of implying PRGs for the class of low-degree  $\mathbb{F}_2$  polynomials in our framework.

► **Theorem 29.** Let  $p : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  be a polynomial of degree  $d$ , and let  $f(x) = (-1)^{p(x)}$ . Then

$$\sum_{\substack{S \subset [n] \\ |S|=k}} |\hat{f}(S)| \leq (k2^{3d})^k \quad \forall k = 1, \dots, n.$$

We note that  $L_2$  bounds do not hold for low-degree polynomials, as can be witnessed by taking a high-rank quadratic polynomial. We prove Theorem 29 in the remainder of this section.

We first introduce some notation to simplify the presentation. Define

$$W_k(f) := \sum_{|S|=k} |\hat{f}(S)|$$

denote the weight of the level- $k$  Fourier coefficients of a Boolean function  $f$ , and let

$$W(d, k) := \max\{W_k(f) : f = (-1)^p, \deg(p) \leq d\}$$

be the maximum of  $W_k$  over degree  $d$  polynomials. Note that we do not make any assumption on the number of variables  $n$ . We prove the following lemma from which Theorem 29 follows relatively easily.

► **Lemma 30.** For any  $d, k \geq 1$ ,

$$W(d, k)^2 \leq 2^{2k}W(d-1, 2k) + W(d, k) \cdot \sum_{\ell=1}^k \binom{k}{\ell} W(d, k-\ell).$$

We first show that Theorem 29 follows easily from Lemma 30.

**Proof of Theorem 29 given Lemma 30.** The proof of Theorem 29 is by induction, first on  $d$  and then on  $k$ . The base case of  $d = 1$  is straightforward, so assume  $d \geq 2$ . By Lemma 30 we have

$$\begin{aligned} W(d, k)^2 &\leq 2^{2k} \left(2k \cdot 2^{3(d-1)}\right)^{2k} + W(d, k) \sum_{\ell=1}^k \binom{k}{\ell} ((k-\ell)2^{3d})^{k-\ell} \\ &\leq (k \cdot 2^{3d-1})^{2k} + W(d, k) \sum_{\ell=1}^k \binom{k}{\ell} ((k-1)2^{3d})^{k-\ell} \\ &= (k \cdot 2^{3d-1})^{2k} + W(d, k) \left( ((k-1)2^{3d} + 1)^k - ((k-1)2^{3d})^k \right). \end{aligned}$$

Assume towards a contradiction that  $W(d, k) > (k2^{3d})^k$ . Dividing by  $W(d, k)$  on both sides gives

$$W(d, k) \leq (k \cdot 2^{3d-1})^k + ((k-1)2^{3d} + 1)^k - ((k-1)2^{3d})^k.$$

If  $k = 1$  then we reach a contradiction as  $2^{3d-1} + 1 \leq 2^{3d}$ . If  $k > 1$  then as  $(k-1)2^{3d} \geq k2^{3d-1}$  the first term gets canceled by the third term, and the second term is at most  $(k2^{3d})^k$ . In either case, we reached a contradiction. ◀

From now on we focus on proving Lemma 30. To that end, fix  $f$  computed by a polynomial of degree  $d$  which maximizes  $W_k(f)$ . We shorthand  $g(S) = |\hat{f}(S)|$ . The following claims are used in the proof of Lemma 30.

► **Claim 31.** For any  $0 \leq a < b \leq n$  and  $A \subset [n]$  of size  $|A| = a$ ,

$$\sum_{B: |B|=b, A \subset B} g(B) \leq W(d, b-a).$$

► **Claim 32.**

$$\sum_{S, T: |S|=|T|=k, S \cap T = \emptyset} g(S)g(T) \leq 2^{2k}W(d-1, 2k).$$

► **Claim 33.** For any  $1 \leq \ell \leq k$ ,

$$\sum_{S, T: |S|=|T|=k, |S \cap T| = \ell} g(S)g(T) \leq \binom{k}{\ell} W(d, k)W(d, k-\ell).$$

We first show how to prove Lemma 30 using the above claims.

**Proof of Lemma 30.** We have,

$$\begin{aligned}
W(d, k)^2 &= \sum_{S, T: |S|=|T|=k} g(S)g(T) \\
&= \sum_{\ell=0}^k \sum_{S, T: |S|=|T|=k, |S \cap T|=\ell} g(S)g(T) \\
&= \sum_{S, T: |S|=|T|=k, S \cap T = \emptyset} g(S)g(T) + \sum_{\ell=1}^k \sum_{S, T: |S|=|T|=k, |S \cap T|=\ell} g(S)g(T) \\
&\leq 2^{2k} W(d-1, 2k) + W(d, k) \cdot \sum_{\ell=1}^k \binom{k}{\ell} W(d, k-\ell),
\end{aligned}$$

where the last inequality follows by using the bounds from Claim 32 and Claim 33.  $\blacktriangleleft$

We now proceed to prove the missing claims.

**Proof of Claim 31.** We use induction on  $a$  and  $b$ . The claim is direct for  $a = 0$  and any  $b > a$ . Thus suppose  $b > a > 0$  and let  $i \in A$ . Let  $A' = A \setminus \{i\}$ . We have

$$\begin{aligned}
\sum_{B: |B|=k, A \subset B} g(B) &= \sum_{B' \subset [n] \setminus \{i\}: |B'|=b-1, A' \subset B'} g(B' \cup \{i\}) \\
&= \sum_{B' \subset [n] \setminus \{i\}: |B'|=b-1, A' \subset B'} |\widehat{f}(B' \cup \{i\})|.
\end{aligned}$$

Let  $f_{i \rightarrow 1}$  and  $f_{i \rightarrow -1}$  be the functions obtained from  $f$  by setting the  $i$ 'th bit to 1 and  $-1$ , respectively. It is easy to verify that  $|\widehat{f}(B \cup \{i\})| \leq \frac{1}{2}(\widehat{f_{i \rightarrow 1}}(B) + \widehat{f_{i \rightarrow -1}}(B))$ . Thus, continuing with our estimate, we have

$$\begin{aligned}
\sum_{B: |B|=k, A \subset B} g(B) &\leq \frac{1}{2} \sum_{B': |B'|=b-1, A' \subset B' \cup \{i\}} (|\widehat{f_{i \rightarrow 1}}(B')| + |\widehat{f_{i \rightarrow -1}}(B')|) \\
&\leq W(d, (b-1) - (a-1)) = W(d, b-a),
\end{aligned}$$

where the last inequality follows from induction hypothesis.  $\blacktriangleleft$

**Proof of Claim 32.** For any  $S \subset [n]$ , let  $e_S \in \{-1, 1\}$  be the sign of  $\widehat{f}(S)$ , so that  $g(S) = e_S \cdot \widehat{f}(S)$ . Let  $X, Y, Z$  be independent uniform distributions on  $\{-1, 1\}^n$ . We have

$$\begin{aligned}
\sum_{S, T: |S|=|T|=k, S \cap T = \emptyset} g(S)g(T) &= \sum_{S, T: |S|=|T|=k, S \cap T = \emptyset} e_S e_T \mathbb{E}_Y [f(Y)Y^S] \cdot \mathbb{E}_Z [f(Z)Z^T] \\
&= \sum_{S, T: |S|=|T|=k, S \cap T = \emptyset} e_S e_T \mathbb{E}_{Y, Z} [f(Y)Y^S f(Z)Z^T] \\
&= \sum_{S, T: |S|=|T|=k, S \cap T = \emptyset} e_S e_T \mathbb{E}_{X, Y, Z} [f(X \circ Y) f(X \circ Z) X^{S \cup T} Y^S Z^T].
\end{aligned}$$

This follows as  $(Y, Z)$  and  $(X \circ Y, X \circ Z)$  are identically distributed. Now consider any fixing of  $Y = y$  and  $Z = z$ . Define the function  $h_{y, z}(x) = f(x \circ y) f(x \circ z)$ . Recall that  $f = (-1)^p$  where  $p$  is a  $\mathbb{F}_2$ -polynomial of degree  $d$ . Thus  $h = (-1)^q$  where  $q$  is the derivative of  $f$  in direction  $y \circ z$ . In particular, its degree is at most  $d-1$ . Thus we have

$$\begin{aligned}
\sum_{S, T: |S|=|T|=k, S \cap T = \emptyset} e_S e_T y^S z^T \mathbb{E} [f(X \circ y) f(X \circ z) X^{S \cup T}] &\leq \binom{2k}{k} \sum_{R: |R|=2k} |\mathbb{E} [h(X) X^R]| \\
&\leq 2^{2k} W(d-1, 2k).
\end{aligned}$$

The proof follows now by noting that the above bound holds for any choice of  $y$  and  $z$ , and then averaging over  $y = Y, z = Z$ . ◀

**Proof of Claim 33.** We have,

$$\begin{aligned}
 \sum_{S,T:|S|=|T|=k,|S\cap T|=\ell} g(S)g(T) &\leq \sum_{L:|L|=\ell} \left( \sum_{S:|S|=k,L\subset S} g(S) \right)^2 \\
 &\leq \left( \max_{L:|L|=\ell} \sum_{S:|S|=k,L\subset S} g(S) \right) \left( \sum_{L,S:|L|=\ell,|S|=k,L\subset S} g(S) \right) \\
 &\leq W(d,k-\ell) \cdot \left( \sum_{S:|S|=k} \sum_{L:L\subset S,|L|=\ell} g(S) \right) \\
 &\hspace{15em} ((\text{using Claim 31})) \\
 &\leq W(d,k-\ell) \cdot \binom{k}{\ell} \cdot W(d,k). \quad \blacktriangleleft
 \end{aligned}$$

## 7 Smoothness

In this section we provide a partial answer for Open Problem 5, regarding early termination of the random walk. Let  $Y_t \in [-1, 1]^n$  be the location of the random walk at time  $t$ . We would like to guarantee that if  $Y_t$  is close enough to  $\text{sign}(Y_t)$  then we can round  $Y_t$  to  $\text{sign}(Y_t)$  without changing the value of  $f$  by too much. Therefore, given  $f : [-1, 1]^n \rightarrow [-1, 1]$ , it would be desirable to show  $f$  is “smooth” enough: there is a bound  $W$  such that

$$\forall \alpha, \beta \in [-1, 1]^n, |f(\alpha) - f(\beta)| \leq W \|\alpha - \beta\|_\infty.$$

Observe that should such  $W$  exist, then if at some step  $t$  we have  $\|Y_t - \text{sign}(Y_t)\|_\infty \leq \varepsilon/W$ , then we can terminate the random walk immediately and guarantee that  $\|f(Y_t) - f(\text{sign}(Y_t))\|_\infty \leq \varepsilon$ . We show that such smoothness property holds for functions with bounded sensitivity.

### 7.1 Bounded sensitivity functions.

We show that smoothness follows from a bound on the (maximum) sensitivity of a boolean function.

► **Lemma 34.** *Let  $f : \{-1, 1\}^n \rightarrow [-1, 1]$  be a boolean function with maximum sensitivity  $s$ . Then, for any  $\alpha, \beta \in [-1, 1]^n$  it holds that*

$$|f(\alpha) - f(\beta)| \leq 4s \|\alpha - \beta\|_\infty.$$

We first consider the case that  $\|\alpha - \beta\|_\infty$  is very small.

► **Claim 35.** *Let  $f : \{-1, 1\}^n \rightarrow [-1, 1]$  be a boolean function with maximum sensitivity  $s$ . Let  $\alpha, \beta \in [-1, 1]^n$  such that  $\|\alpha - \beta\|_\infty \leq 1/n^2$ . Then*

$$|f(\alpha) - f(\beta)| \leq 4s \|\alpha - \beta\|_\infty.$$

To prove the result for arbitrary  $\alpha, \beta \in [-1, 1]^n$  using Claim 35, consider the line segment from  $\alpha$  to  $\beta$  and integrate  $f$  along that line segment. Thus, Lemma 34 follows directly from Claim 35.

**Proof of Claim 35.** Let  $\delta = \|\alpha - \beta\|_\infty$ . We first consider the easier case of  $\alpha \in \{-1, 1\}^n$ . Pick  $b \in \{-1, 1\}^n$  randomly by flipping each coordinate of  $\alpha$  independently with probability  $|\alpha_i - \beta_i|/2$  so that  $\mathbb{E}f(b) = f(\beta)$ . Note that  $f(b) \neq f(\alpha)$  if either exactly one sensitive coordinate of  $\alpha$  is flipped, which occurs with probability at most  $s\delta$ , or if at least two coordinates get flipped, which occurs with probability at most  $(n\delta)^2$ . Therefore

$$|f(\alpha) - \mathbb{E}f(b)| \leq s\delta + n^2\delta^2 \leq 2s\delta$$

given our assumption on  $\delta$ .

Next, consider the general case of  $\alpha \in [-1, 1]^n$ . This case requires introducing an extra point  $\gamma \in [-1, 1]^n$  in a way that allows us to prove

$$|f(\alpha) - f(\gamma)| \leq 2s \cdot \|\alpha - \gamma\|_\infty \tag{1}$$

and

$$|f(\beta) - f(\gamma)| \leq 2s \cdot \|\beta - \gamma\|_\infty \tag{2}$$

separately. We choose  $\gamma$  in a way that  $\forall i \in [n], \gamma_i = \alpha_i$  or  $\gamma_i = \beta_i$ . These equations altogether give the claim. To choose  $\gamma$ , let  $S \subset [n]$  be the set of coordinates that  $|\alpha_i| < |\beta_i|$  and pick  $\gamma_i = \alpha_i$  if  $i \in S$ , and  $\gamma_i = \beta_i$  otherwise.

We next prove Equation (1). The proof of Equation (2) is analogous. Consider a joint random variable  $(a, c)$  that satisfies the following properties:

1.  $a \in \{-1, 1\}^n, c \in [-1, 1]^n, \mathbb{E}a = \alpha$ , and  $\mathbb{E}c = \gamma$ .
2. The marginal distributions of  $a$  and  $c$  are product distributions.
3.  $\|a - \mathbb{E}_c[c|a]\|_\infty \leq \|\alpha - \gamma\|_\infty$  holds with probability one.

Observe that given such  $(a, c)$ ,

$$|f(\alpha) - f(\gamma)| = |\mathbb{E}_{a,c}[f(a) - f(c)]| \leq \mathbb{E}_a |f(a) - \mathbb{E}_c[f(c)|a]| \leq 2s \cdot \|\alpha - \gamma\|_\infty,$$

where the last inequality uses the first case in the proof, as  $a \in \{-1, 1\}^n$ .

Now let us construct the joint random variable  $(a, c)$ . Fix  $i \in [n]$  and suppose without loss of generality that  $\alpha_i \geq 0$ . Note that by construction  $-\alpha_i \leq \gamma_i \leq \alpha_i$ . First sample  $a_i$  so that  $\mathbb{E}[a_i] = \alpha_i$ . If  $a_i = -1$  then set  $c_i = -1$ , otherwise set  $c_i = \frac{2\gamma_i + 1 - \alpha_i}{1 + \alpha_i}$ . It's easy to check that this choice of  $(a, c)$  satisfies the required conditions, finishing the proof.  $\blacktriangleleft$

---

## References

- 1 Miklós Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant depth circuits. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 11–19. IEEE, 1985.
- 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 Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 3–10. IEEE, 2010.
- 4 Mark Braverman. Polylogarithmic independence fools ac 0 circuits. *Journal of the ACM (JACM)*, 57(5):28, 2010.
- 5 Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *Electronic Colloquium on Computational Complexity (ECCC), pages TR17–171*, 2017.

- 6 Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 120–129. IEEE, 2012.
- 7 Parikshit Gopalan, Rocco A Servedio, and Avi Wigderson. Degree and sensitivity: tails of two distributions. In *Proceedings of the 31st Conference on Computational Complexity*, page 13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.
- 8 Johan Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 6–20. ACM, 1986.
- 9 Pooya Hatami and Avishay Tal. Pseudorandom generators for low-sensitivity functions. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 24, page 25, 2017.
- 10 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.
- 11 Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.
- 12 Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 61–67. IEEE, 2012.
- 13 Shachar Lovett, Avishay Tal, and Jiapeng Zhang. Robust sensitivity. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 23, page 161, 2016.
- 14 Yishay Mansour. An  $n^{O(\log \log n)}$  learning algorithm for dnf under the uniform distribution. *Journal of Computer and System Sciences*, 50(3):543–550, 1995.
- 15 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM journal on computing*, 22(4):838–856, 1993.
- 16 Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991.
- 17 Noam Nisan and Avi Wigderson. Hardness vs. randomness. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 2–11. IEEE, 1988.
- 18 Omer Reingold, Thomas Steinke, and Salil Vadhan. Pseudorandomness for regular branching programs via fourier analysis. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 655–670. Springer, 2013.
- 19 Hans-Ulrich Simon. A tight  $\omega(\log \log n)$ -bound on the time for parallel ram’s to compute nondegenerated boolean functions. In *International Conference on Fundamentals of Computation Theory*, pages 439–444. Springer, 1983.
- 20 Avishay Tal. Tight bounds on the fourier spectrum of ac0. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 79. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- 21 Luca Trevisan and Tongke Xue. A derandomized switching lemma and an improved derandomization of ac0. In *Computational Complexity (CCC), 2013 IEEE Conference on*, pages 242–247. IEEE, 2013.
- 22 Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. *Computational Complexity*, 18(2):209–217, 2009.





# A PRG for Boolean PTF of Degree 2 with Seed Length Subpolynomial in $\epsilon$ and Logarithmic in $n$

Daniel Kane<sup>1</sup>

UC San Diego  
dakane@ucsd.edu

Sankeerth Rao

UC San Diego  
skaringu@ucsd.edu

---

## Abstract

---

We construct and analyze a *pseudorandom generator* for degree 2 boolean *polynomial threshold functions*. Random constructions achieve the optimal seed length of  $O(\log n + \log \frac{1}{\epsilon})$ , however the best known explicit construction of [8] uses a seed length of  $O(\log n \cdot \epsilon^{-8})$ . In this work we give an *explicit* construction that uses a seed length of  $O(\log n + (\frac{1}{\epsilon})^{o(1)})$ . Note that this improves the seed length substantially and that the dependence on the error  $\epsilon$  is *additive* and only grows *subpolynomially* as opposed to the previously known multiplicative polynomial dependence.

Our generator uses *dimensionality reduction* on a *Nisan-Wigderson* based pseudorandom generator given by Lu, Kabanets [18].

**2012 ACM Subject Classification** Theory of computation → Pseudorandomness and derandomization

**Keywords and phrases** Pseudorandomness, Polynomial Threshold Functions

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.2

**Acknowledgements** The authors would like to thank Zhenjian Lu and Valentine Kabanets for all their spontaneous help and collaboration.

## 1 Introduction

### 1.1 Background and importance

We say that a function  $f : \mathbb{R}^n \rightarrow \{+1, -1\}$  is a (degree- $d$ ) *polynomial threshold function* (PTF) if it is of the form  $f(x) = \text{sgn}(p(x))$  for  $p$  some (degree- $d$ ) polynomial in  $n$  variables. Polynomial threshold functions make up a natural class of Boolean functions and have applications to a number of fields of computer science such as circuit complexity [2], communication complexity [17] and learning theory [14].

In this paper, we study the question of pseudorandom generators (PRGs) for polynomial threshold functions of Bernoulli inputs (and in particular for  $d=2$ ). In other words, we wish to find explicit functions  $F : \{\pm 1\}^s \rightarrow \{\pm 1\}^n$  so that for any degree-2 polynomial threshold function  $f$ , we have

$$\left| \mathbb{E}_{x \sim_u \{\pm 1\}^s} [f(F(x))] - \mathbb{E}_{X \sim \{\pm 1\}^n} [f(X)] \right| < \epsilon.$$

We say that such an  $F$  is a pseudorandom generator of seed length  $s$  that fools degree-2 polynomial threshold functions with respect to the Bernoulli distribution to within  $\epsilon$ . In

---

<sup>1</sup>Supported by NSF Award CCF-1553288(CAREER) and a Sloan Research Fellowship.



■ **Table 1** Pseudorandom Generators

Paper	Bernoulli/Gaussian	$d$	Seedlength $s$
Diakonikolas, Gopalan, etal [5]	Bernoulli	1	$\log n \cdot O(\epsilon^{-2} \log^2(1/\epsilon))$
Meka, Zuckerman [15]	Bernoulli	1	$O(\log n + \log^2(1/\epsilon))$
Gopalan, Kane, Meka [7]	Bernoulli	1	$O(\log(n/\epsilon) \cdot [\log \log(n/\epsilon)]^2)$
Diakonikolas, Kane, Nelson [8]	Gaussian	1	$\log n \cdot O(\epsilon^{-2})$
Kane [12]	Gaussian	1	$O(\log n + \log^{3/2}(1/\epsilon))$
Diakonikolas, Kane, Nelson [8]	Both	2	$\log n \cdot O(\epsilon^{-8})^\dagger$
Kane [12]	Gaussian	2	$\log n \cdot \exp[\tilde{O}(\log(1/\epsilon)^{2/3})]$
Kane [13]	Gaussian	2	$O(\log^6(1/\epsilon) \cdot \log n \cdot \log \log(n/\epsilon))$
<b>Kane, Sankeerth This paper</b>	<b>Bernoulli</b>	<b>2</b>	$O(\log n + e^{\sqrt{\log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}}})$
Kane [9]	Both	$d$	$\log n \cdot O_d(\epsilon^{-2^{O(d)}})$
Meka, Zuckerman [15]	Bernoulli	$d$	$\log n \cdot 2^{O(d)} \epsilon^{-8d-3}$
Kane [11]	Bernoulli	$d$	$\log n \cdot O_d(\epsilon^{-11.1})$
Kabanets, Lu [18]	Bernoulli	$d$	$e^{O(\sqrt{d \log n \log \log(n/\epsilon)})}$
Kane [10]	Gaussian	$d$	$\log n \cdot 2^{O(d)} \epsilon^{-4.1}$
Kane [11]	Gaussian	$d$	$\log n \cdot O_d(\epsilon^{-2.1})$
Kane [12]	Gaussian	$d$	$\log n \cdot O_{c,d}(\epsilon^{-c})$

this paper, we develop a generator with  $s = O(\log n + e^{\sqrt{\log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}}})$ . The main idea is to apply a Johnson-Lindenstrauss like dimensionality reduction on the Nisan-Wigderson based pseudorandom generator by Lu, Kabanets [18]. A random construction shows the existence of a PRG that uses a seed length of  $s = O(\log n + \log \frac{1}{\epsilon})$ , however there are no known constructions that achieve this. The best known constructions for Boolean degree 2 PTFs use a seed length of  $s = \log n \cdot \text{poly}(\frac{1}{\epsilon})$ , the current work improves this especially the error dependence to  $s = O(\log n + \text{subpoly}(\frac{1}{\epsilon}))$ . The Meka-Zuckerman PRG for LTFs in [15] uses a similar type of dimensionality reduction idea to reduce the seed length from  $O(\log^2(\frac{n}{\epsilon}))$  to  $O(\log n + \log^2 \frac{1}{\epsilon})$ .

## 1.2 Prior Work

An existential argument shows that there are optimal pseudo random generators of seed length  $O(d \log n + \log \frac{1}{\epsilon})$ . There has been a lot of research towards giving explicit constructions that approach this seed length. The following are the past results of pseudorandom generators constructed for PTFs of degree  $d$ .

## 1.3 Our results and merits of the paper

The main goal for degree 2 PRG constructions has been to achieve the optimal seed length of  $O(\log n + \log(\frac{1}{\epsilon}))$  via explicit constructions. Random constructions do achieve this optimal seed length, however the best known explicit construction of [8] uses a seed length of  $O(\log n \cdot \epsilon^{-8})$ . In this paper we give an *explicit* construction that uses a seed length of  $O(\log n + (\frac{1}{\epsilon})^{o(1)})$ . Note that this improves the seed length substantially and that the dependence on the error  $\epsilon$  is *additive* and only grows *subpolynomially* as opposed to the

<sup>†</sup>The original analysis only got  $\log n \cdot \tilde{O}(\epsilon^{-9})$  until [11] led to an improved analysis using the same ideas.

previously known multiplicative polynomial dependence. In particular we give a construction for a seed length of  $O(\log n + e^{\sqrt{\log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}}})$ . The major improvement of this work is in separating out the  $n$ -dependence from the  $\epsilon$ -dependence. It would be very interesting to improve this further to the optimal logarithmic dependence on  $\epsilon$ .

The main theorem of this paper is:

► **Theorem 1.** *Given  $\epsilon > 0, n \in \mathbb{N}$ , we construct a function  $F : \{\pm 1\}^s \rightarrow \{\pm 1\}^n$  such that for any degree 2 polynomial  $p : \{\pm 1\}^n \rightarrow \mathbb{R}$ , the probability that  $p(x) \geq 0$  at a uniformly random point in  $\{\pm 1\}^n$  is approximately (within  $\epsilon$ ) equal to the probability that  $p(F[z]) \geq 0$  at a uniformly random point in  $\{\pm 1\}^s$ . That is,*

$$\left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{z \sim \{\pm 1\}^s} \text{sgn}(p(F[z])) \right| \leq \epsilon.$$

Here  $s$  is called the seed length of  $F$  and it is given by  $s = O(\log n + e^{\sqrt{\log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon}}})$ .

We construct  $F$  by doing a dimensionality reduction like argument on a Nisan-Wigderson based pseudorandom generator for Boolean PTFs constructed by Kabanets, Lu in [18]. Their construction uses a seed length of  $O(e^{\sqrt{\log n \log \log \frac{n}{\epsilon}}})$ .

Our generator is best thought of as a dimension reduction gadget. It reduces the problem of finding a PRG in  $n$  dimensions to that of finding a PRG in  $\text{poly}(1/\epsilon)$  dimensions (with an additive loss of  $O(\log n)$  in seed length). This means that if you combine it with a generator that has seed length  $s(n, \epsilon)$ , we get a new generator with seed length  $O(\log n) + s(\text{poly}(1/\epsilon), \epsilon)$ . This is particularly useful if the other generator is the Kabanets-Lu generator, since that generator has a great  $\epsilon$  dependence at the expense of having a poor dependence on  $n$ . One could also use the trivial generator (i.e. the uniform distribution over the entire hypercube for which  $s(n, \epsilon) = n$ ), and get a generator with seed length  $O(\log n) + \text{poly}(1/\epsilon)$ .

In particular we don't require the Kabanets-Lu generator, but since what we do only reduces the dimension of the problem, we do need *some* other generator. When we use the trivial PRG instead after using our technique, we can get  $O(\log n) + \text{poly}(1/\epsilon)$ . We believe that even this is new.

## 1.4 Proof overview with an outline of key technical ideas used

We construct our PRG  $F$  by composing a Johnson-Lindenstrauss matrix  $L^t$  with the following PRG  $H$  constructed by Kabanets, Lu in [18], that is  $F = L \circ H$ . They construct  $H$  by constructing a hard function that can't be computed by PTFs and using the Nisan-Wigderson hardness vs randomness template.

► **Theorem 2.** *Given  $\epsilon > 0, n \in \mathbb{N}$ , one can construct a function  $H : \{\pm 1\}^t \rightarrow \{\pm 1\}^n$  such that for any degree 2 polynomial  $q : \{\pm 1\}^n \rightarrow \mathbb{R}$ , the probability that  $q(x) \geq 0$  at a uniformly random point in  $\{\pm 1\}^n$  is approximately (within  $\epsilon$ ) equal to the probability that  $q(H[z]) \geq 0$  at a uniformly random point in  $\{\pm 1\}^t$ . That is,*

$$\left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(q(x)) - \mathbb{E}_{z \sim \{\pm 1\}^t} \text{sgn}(q(H[z])) \right| \leq \epsilon.$$

where the seed length of  $H$  is  $t = O(e^{\sqrt{\log n \log \log \frac{n}{\epsilon}}})$ .

Let's first understand the seed length needed for our PRG  $F$ .

### Seed length

We use Kabanets PRG  $H$  to stretch from an initial seed of length  $t$  to dimension  $m$ . This is further stretched by  $L$  from  $m$  to  $n$  (think of  $m$  as  $(\frac{1}{\epsilon})^{\Omega(1)}$ ). Thus the seed length  $t$  needed to make Kabanets PRG  $H$  work is  $t = O(e^{\sqrt{\log m \log \log(\frac{m}{\epsilon})}})$ , since  $m = (\frac{1}{\epsilon})^{\Omega(1)}$  this would amount to a seed of  $t = O(e^{\sqrt{\log(\frac{1}{\epsilon}) \log \log(\frac{1}{\epsilon})}})$ .  $L$  would further use randomness needing an extra seed of  $O(\log n)$ . Thus  $F$  would need a total seed length  $s = O(\log n + e^{\sqrt{\log(\frac{1}{\epsilon}) \log \log(\frac{1}{\epsilon})}})$ .

### Analysis

To analyse our PRG we split the error into two steps as follows:

- *Replace the  $n$  pure random bits input by  $m$  pure random bits* We replace the  $n$  pure random bits  $x$  by  $Ly$ , where  $y$  has only  $m$  purely random bits.
- *Replace the  $m$  pure random bits by  $t$  pseudorandom bits* We further replace the  $m$  pure random bits  $y$  by even fewer  $t$  purely random bits  $z$ . This is done via  $H$ , that is  $y = H[z]$ .

We depict this in the following equation:

$$\begin{aligned} \left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{z \sim \{\pm 1\}^t} \text{sgn}(p(F[z])) \right| &= \left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{z \sim \{\pm 1\}^t} \text{sgn}(pL(H[z])) \right| \\ &\leq \left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right| \\ &\quad + \left| \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) - \mathbb{E}_{z \sim \{\pm 1\}^t} \text{sgn}(pL(H[z])) \right|. \end{aligned}$$

Let's understand these steps:

#### 1.4.1 Stretch $t$ pure bits to $m$ pure bits,

$$\left| \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) - \mathbb{E}_{z \sim \{\pm 1\}^t} \text{sgn}(pL(H[z])) \right|$$

As  $L$  is a linear operator,  $pL$  would still be a polynomial of degree 2. This error is small because  $H$  fools all degree 2 PTFs including  $pL$ . Thus we are using the PRG  $H$  to go from a space of dimension  $t = O(e^{\sqrt{\log(\frac{1}{\epsilon}) \log \log(\frac{1}{\epsilon})}})$  to a space of dimension  $m = \frac{1}{\epsilon^{\Omega(1)}}$ . The main technical idea used by [18] to achieve this is to give a hard function for PTFs and invoke the Nisan-Wigderson *hardness vs randomness* template.

#### 1.4.2 Stretch $m$ pure bits to $n$ pure bits,

$$\left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right|$$

We show that this error is small in two steps:

- *Move from Boolean to Gaussian setting* We first move from the Boolean input to the Gaussian input setting. This can be done very easily for some special polynomials (*regular*). For a non regular polynomial we use technical ideas like the *regularity lemma* [6].
- *$L$  is a good PRG for Gaussian inputs* When the input is Gaussian we have a lot of geometric structure. In particular using Central limit theorems (as done in [3]) any polynomial can be seen as a low dimensional very structured part and a lump mass that can be approximated by a single Gaussian. We show that  $L$  preserves this structure and thus we don't incur much error in changing  $x$  to  $Ly$ .

## Move from Boolean to Gaussian setting

There are two technical ideas used here.

- **Regular polynomials** A polynomial is *regular* if no single input variable has a huge influence over the value of the polynomial. When a polynomial is regular one doesn't incur much loss when switching the input from boolean to gaussian as shown by the *Invariance principle* [16]. Think of this *replacement* as a telescope of replacing the variables one at a time and the error incurred when the  $i$ th variable is replaced is captured by its *influence*. In fact in this paper we show that  $L$  keeps *regular* polynomials regular. Thus if  $p$  is regular then so is  $pL$ . Thus we switch from boolean inputs to gaussian inputs for both  $p$  and  $pL$ .
- **Regularity Lemma** If a polynomial is not *regular*, then you could incur huge loss by directly switching the inputs from boolean to Gaussian. However there could be very few variables that have such a huge influence over the polynomial. So if these few variables are fixed the rest of the polynomial will either have negligible mass or be regular both of which are amenable to replacement from boolean to gaussian inputs. Thus the technical idea used here is the *Regularity lemma* of [6] which shows that every polynomial can be seen as a decision tree corresponding to the high influence variables that are fixed wherein the leaves are either regular or almost constant polynomials. We show that our JL matrix  $L$  interacts well with the Regularity Lemma. That is under the hash function of  $L$  we don't see any collision for the high influence variables whp. Also the low influence variables that do hash collide with these high influence variables contribute very little mass to  $pL$ .

## $L$ is a good PRG for Gaussian inputs

There are two technical ideas used here.

- **Central Limit Theorem** If all the eigenvalues of a polynomial are small relative to its variance then the polynomial can be well approximated by a single Gaussian as shown in [3] via a Central Limit Theorem. Since the variance of the polynomial is a constant, there can be only few large eigenvalues. Thus any polynomial can be seen as a structured polynomial consisting of the few large eigenvalues and an eigenregular polynomial that can be replaced by a single Gaussian. Thus the only essential information is in the top eigenstructure and the lump mass of the rest of the eigenvalues.
- **Structure preservation by  $L$**  We show that our JL matrix preserves the structure of these top few eigenvalues and also maps polynomials with small eigenvalues to polynomials with small eigenvalues with high probability. Thus  $L$  keeps this top eigenstructure+lump mass structure intact. It also approximately preserves the  $L^2$  norms and covariance of polynomials and thus we see that  $L$  is a good PRG in the Gaussian setting.

The Meka-Zuckerman PRG for LTFs in [15] uses a similar type of dimensionality reduction idea to reduce the seed length from  $O(\log^2(\frac{n}{\epsilon}))$  to  $O(\log n + \log^2 \frac{1}{\epsilon})$ .

## 1.5 Overview of the paper

We present the mathematical preliminaries required in section 2 and show that the PRG construction works under the assumption that  $p$  is regular in section 3. Then we prove a reduction from the general case to the special case of regular polynomials in Section 4. We present the conclusions in section 5.

**Note:**

All through the paper we will be bounding errors whp as  $\frac{1}{m^{\Omega(1)}}$ . Note that these errors are less than  $\epsilon$  if  $m$  is chosen to be a sufficiently a large polynomial of  $\frac{1}{\epsilon}$ . Think of whp to mean with probability  $1 - \frac{1}{m^{\Omega(1)}}$ .

All through the paper we leave the errors in terms of  $m$ , think of adding up all the errors and union bounding the probabilities and fixing all the parameters in terms of  $\epsilon$  and then we choose a sufficiently large  $m = \frac{1}{\epsilon^{\Omega(1)}}$  to make the sum of all the errors  $O(\epsilon)$  and the union of all the error probabilities  $O(\epsilon)$ .

**2 Preliminaries****2.1 Basic results on polynomials, concentration, anticoncentration, invariance and regularity****Concentration**

We begin with a standard concentration bound from [4] that says that Gaussian degree-2 polynomials are concentrated around their mean. We would need this multiple times in the paper to show concentration of Gaussian polynomials.

► **Lemma 3.** *Let  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  be a degree-2 polynomial. We have*

$$\Pr_{x \sim \mathcal{N}^n(0,1)} \left[ |p(x) - \mathbb{E}[p(x)]| > t\sqrt{\text{Var}[p]} \right] \leq e^{-\Omega(t)}.$$

**Anticoncentration**

We will need the following standard Carbery-Wright anticoncentration bound from [1],[4] that proves a bound on the mass a Gaussian degree-2 polynomial could have around any point. This would be useful in many instances including when we change functions of Gaussians.

► **Lemma 4.** *Let  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  be a degree-2 polynomial that is not identically 0. Then for all  $\epsilon > 0$  and all  $\theta \in \mathbb{R}$ , we have*

$$\Pr_{x \sim \mathcal{N}^n(0,1)} \left[ |p(x) - \theta| < \epsilon\sqrt{\text{Var}[p]} \right] \leq O(\sqrt{\epsilon}).$$

The following lemma from [4] is very useful as it helps us bound the distributional distance between two Gaussian polynomials by just bounding the  $L^2$  norm. The proof follows from an application of Lemmas 3,4.

► **Lemma 5.** *Let  $a(x), b(x)$  be degree-2 polynomials over  $\mathbb{R}^n$ . For  $x \sim \mathcal{N}^n(0,1)$ , if  $\mathbb{E}[a(x) - b(x)] = 0$ ,  $\text{Var}[a] = 1$  and  $\text{Var}[a - b] \leq (\beta/2)^6$ , then*

$$\left| \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(a(x)) - \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(b(x)) \right| \leq O(\beta).$$

**Invariance Principle**

The Invariance principle bounds the change in  $\mathbb{E}[\text{sgn}(p(x))]$  when the input is changed from Boolean to Gaussian. We use the following lemma based on [16].

► **Lemma 6.** For any degree 2 multilinear polynomial  $p = \sum_{i,j \in [n]} a_{ij}x_i x_j + \sum_{l \in [n]} b_l x_l + C$ , we have the following bound:

$$\left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) \right| \leq O \left[ \frac{\sum_{i=1}^n \text{Inf}_i^2(p)}{(\text{Var}[p])^2} \right]^{\frac{1}{9}}.$$

where the  $i$ th influence of  $p$  is defined as  $\text{Inf}_i(p) = \mathbb{E} \left| \frac{\partial p}{\partial x_i} \right|^2 = 2 \sum_{j \in [n]} a_{ij}^2 + b_i^2$ .

Think of  $i$ th influence as the variance of  $p$  along the  $i$ th coordinate.

Observe that  $\text{Var}[p] \leq \sum_{i=1}^n \text{Inf}_i(p) \leq 2\text{Var}[p]$ . Now we define the notion of *regularity* for polynomials which essentially means that there is no single variable whose influence is very large as compared to the rest of the variables.

► **Definition 7.** We say that the polynomial  $p$  is  $\tau$ -regular if  $\max_{i \in [n]} \text{Inf}_i(p) \leq \tau \text{Var}[p]$ .

Thus for a  $\tau$ -regular polynomial  $p$  we can bound the replacement error above as  $O(\tau^{\frac{1}{9}})$  because

$$\frac{\sum_{i=1}^n \text{Inf}_i^2(p)}{(\text{Var}[p])^2} \leq \frac{[\max_i \text{Inf}_i(p)] \sum_{i=1}^n \text{Inf}_i(p)}{(\text{Var}[p])^2} \leq 2\tau.$$

Note that when we apply this, we pick  $\tau = \epsilon^{O(1)}$ .

**Regularity Lemma**

We will use the following Regularity Lemma from [6]:

► **Lemma 8.** Every multilinear degree 2 polynomial  $p : \{\pm 1\}^n \rightarrow \mathbb{R}$  can be written as a decision tree of depth  $D = \frac{1}{\tau} \cdot O\left(\log \frac{1}{\tau\theta}\right)^{O(1)}$  such that with probability  $(1-\theta)$  over a random leaf the resulting polynomial  $p_\alpha$  is either

- (i)  $\tau$  regular, OR
- (ii)  $\text{Var}(p_\alpha) < \theta \|p\|_2^2$ .

Note that when we apply this Regularity lemma we will choose  $\theta = \frac{1}{\sqrt{m}}$ ,  $\tau = \epsilon^{O(1)}$  so that  $D = (\log m)^{O(1)}$ . After all the parameters are fixed we finally pick  $m = \frac{1}{\epsilon^{\Omega(1)}}$  large enough so that all the errors get bounded by  $O(\epsilon)$ .

**2.2 Eigenvalues of polynomials, Central Limit Theorem.**

**Eigenvalues**

Let  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  be a multilinear polynomial of degree 2. Thus there exist a real symmetric matrix  $A$ , a vector  $B^t$  and a constant  $C$  such that

$$p(x) = x^t A x + B^t x + C.$$

The eigenvalues of  $p$  are defined to be the eigenvalues  $\lambda_1, \dots, \lambda_n$  of the real symmetric matrix  $A$ . Since  $p$  is a multilinear polynomial we have  $\sum_{i=1}^n \lambda_i = 0$ .





Note that this is a standard *Johnson-Lindenstrauss* matrix. In the following Lemma we show that they preserve  $L^2$  norms and inner products of vectors to give a feel for the kind of computations we need. In fact  $L^t$  preserves a lot more structure as we shall see in the next section.

► **Lemma 11.** *For any  $n, \epsilon > 0$ , there exists an  $m = \text{poly}(\frac{1}{\epsilon})$  and an explicit family of Linear transformations  $L^t$  (with seed length  $O(\log n)$  from  $\{\pm 1\}^m \rightarrow \{\pm 1\}^n$ ) so that for any two unit vectors  $v_1, v_2 \in \mathbb{R}^n$  we have*

$$|\langle L^t v_1, L^t v_2 \rangle - \langle v_1, v_2 \rangle| < \epsilon \text{ wp } 1 - 2\epsilon \text{ over } L.$$

**Proof.** We know that  $L^t v_1 = \sum_{i=1}^n v_1^i c_i, L^t v_2 = \sum_{j=1}^n v_2^j c_j$ .

Thus we have

$$\begin{aligned} \langle L^t v_1, L^t v_2 \rangle &= \left\langle \sum_{i=1}^n v_1^i c_i, \sum_{j=1}^n v_2^j c_j \right\rangle = \sum_{i,j \in [n]} v_1^i v_2^j \langle c_i, c_j \rangle \\ \langle L^t v_1, L^t v_2 \rangle - \langle v_1, v_2 \rangle &= \sum_{i \neq j \in [n]} v_1^i v_2^j \langle c_i, c_j \rangle \\ (\langle L^t v_1, L^t v_2 \rangle - \langle v_1, v_2 \rangle)^2 &= \sum_{\substack{i_1 \neq j_1 \\ i_2 \neq j_2}} v_1^{i_1} v_1^{i_2} v_2^{j_1} v_2^{j_2} \langle c_{i_1}, c_{j_1} \rangle \langle c_{i_2}, c_{j_2} \rangle. \end{aligned}$$

Note that when averaged wrt  $\mathbb{E}_L$ , the only terms that survive are those that are paired either as  $(i_1 = i_2, j_1 = j_2)$  or  $(i_1 = j_2, i_2 = j_1)$ .

The rest of the terms average to 0 because of the sign  $\sigma$ , that is  $\mathbb{E}_\sigma[\sigma(i_1)\sigma(i_2)\sigma(j_1)\sigma(j_2)]$  only survives if the indices are paired and we already have the constraints  $i_1 \neq j_1, i_2 \neq j_2$ .

Thus we have

$$\begin{aligned} \mathbb{E}_L(\langle L^t v_1, L^t v_2 \rangle - \langle v_1, v_2 \rangle)^2 &= \sum_{i \neq j} (v_1^i)^2 (v_2^j)^2 \mathbb{E}_L \langle c_i, c_j \rangle^2 + \sum_{i \neq j} v_1^i v_2^i v_1^j v_2^j \mathbb{E}_L \langle c_i, c_j \rangle^2 \\ &= \frac{1}{m} \sum_{i \neq j} [(v_1^i)^2 (v_2^j)^2 + v_1^i v_2^i v_1^j v_2^j] \\ &\leq \frac{1}{m} (|v_1|_2^2 |v_2|_2^2 + \langle v_1, v_2 \rangle^2) \leq \frac{2}{m} \end{aligned}$$

Thus using Chebyshev's inequality we have

$$|\langle L^t v_1, L^t v_2 \rangle - \langle v_1, v_2 \rangle| \leq \frac{1}{m^{1/3}} \text{ wp } \left(1 - \frac{2}{m^{1/3}}\right) \text{ over } L.$$

Now we choose  $m = \frac{1}{\epsilon^3}$  to have

$$|\langle L^t v_1, L^t v_2 \rangle - \langle v_1, v_2 \rangle| \leq \epsilon \text{ wp } 1 - 2\epsilon \text{ over } L.$$

This completes the proof. ◀

To see that norms are preserved too just choose  $v_1 = v_2$  above.

## Note

All through the paper we will be computing such expected moments and bounding them by  $\frac{1}{m^{\Omega(1)}}$  and then use Markov|Chebyshev's inequality (We can't use big moments because  $L$  has limited independence). Think of these errors as small because after all the parameters are fixed we pick  $m = \frac{1}{\epsilon^{\Omega(1)}}$ , to be a sufficiently large polynomial of  $\frac{1}{\epsilon}$  to bound all the terms by  $O(\epsilon)$ . We showed the constants explicitly in the above Lemma but we would not be computing them exactly later on and just denote them with  $O(1)$ .

## 2.4 Technical Lemmas involving L

We show that the transformation  $p \rightarrow pL$  doesn't change the variance by a lot. If  $p(x) = x^t Ax + B^t x + C$  then  $pL(y) = y^t(L^t AL)y + (B^t L)y + C$ . Note that this is just a basic moment computation and doesn't involve anything non trivial.

► **Lemma 12.** *If  $p(x) = x^t Ax + B^t x + C$  is a multilinear polynomial,  $pL(y) = y^t(L^t AL)y + (B^t L)y + C$ . Then,*

$$\mathbb{E}_L \text{Var}[pL] = \sum_{i=1}^n b_i^2 + \left(1 + \frac{3}{m}\right) |A|_F^2 = \text{Var}[p] + \frac{3}{m} |A|_F^2$$

**Proof.** We know that

$$\text{Var}[p] = \sum_{i=1}^n (b_i^2 + a_{ii}^2) + \|A\|_F^2 = \sum_{i=1}^n b_i^2 + \|A\|_F^2.$$

Let's compute the same for  $pL$ . Note that  $L^t AL = \sum_{i,j \in [n]} a_{ij} c_i \otimes c_j$ .

Thus,

$$\begin{aligned} |L^t AL|_F^2 &= \sum_{i_1, j_1, i_2, j_2 \in [n]} a_{i_1, j_1} a_{i_2, j_2} \langle c_{i_1}, c_{i_2} \rangle \langle c_{j_1}, c_{j_2} \rangle \\ &= \sum_{\substack{i_1 \neq j_1 \\ i_2 \neq j_2}} \sigma(i_1) \sigma(i_2) \sigma(j_1) \sigma(j_2) a_{i_1, j_1} a_{i_2, j_2} I\{h(i_1)=h(i_2), h(j_1)=h(j_2)\}. \end{aligned}$$

Let's take expectation over  $\sigma$ . We know that  $\mathbb{E}_\sigma[\sigma(i_1)\sigma(i_2)\sigma(j_1)\sigma(j_2)] \neq 0$  iff  $(i_1, j_1) = (i_2, j_2)$  or  $(i_1, j_1) = (j_2, i_2)$ .

Let  $T_1$  denote the terms of the first kind, then we have  $T_1 = \sum_{i_1, j_1} a_{i_1, j_1}^2 = |A|_F^2$ . Let  $T_2$  denote the terms of the second kind, then we have  $T_2 = \sum_{i_1, j_1} a_{i_1, j_1}^2 I\{h(i_1)=h(j_1)\}$  and thus

$$\mathbb{E}_L[T_2] = \sum_{i_1, j_1} a_{i_1, j_1}^2 \frac{1}{m} = \frac{1}{m} |A|_F^2.$$

Also

$$\begin{aligned} \langle B^t L, B^t L \rangle &= \sum_{i_1, i_2 \in [n]} b_{i_1} b_{i_2} \langle c_{i_1}, c_{i_2} \rangle = \sum_{i_1, i_2} \sigma(i_1) \sigma(i_2) b_{i_1} b_{i_2} I\{h(i_1) = h(i_2)\} \\ \mathbb{E}_\sigma \langle B^t L, B^t L \rangle &= \sum_i b_i^2. \end{aligned}$$

We now compute  $\sum_{l \in [m]} (L^t AL)_{ll}^2$ .

$$\begin{aligned} \sum_{l \in [m]} (L^t AL)_{ll}^2 &= \sum_{l=1}^m \left( \sum_{i, j \in [n]} a_{ij} c_i^l c_j^l \right)^2 = \sum_{i_1, i_2, j_1, j_2} a_{i_1 j_1} a_{i_2 j_2} \sum_{l=1}^m c_{i_1}^l c_{i_2}^l c_{j_1}^l c_{j_2}^l \\ &= \sum_{\substack{i_1 \neq j_1 \\ i_2 \neq j_2}} \sigma(i_1) \sigma(i_2) \sigma(j_1) \sigma(j_2) a_{i_1, j_1} a_{i_2, j_2} I\{h(i_1)=h(i_2)=h(j_1)=h(j_2)\} \end{aligned}$$

Let's take expectation over  $\sigma$ . We know that  $\mathbb{E}_\sigma[\sigma(i_1)\sigma(i_2)\sigma(j_1)\sigma(j_2)] \neq 0$  iff  $(i_1, j_1) = (i_2, j_2)$  or  $(i_1, j_1) = (j_2, i_2)$ .

$$\mathbb{E}_\sigma \sum_{l \in [m]} (L^t AL)_{ll}^2 = 2 \sum_{i_1, j_1} a_{i_1 j_1}^2 I\{h(i_1) = h(j_1)\}$$

Thus,

$$\mathbb{E}_L \sum_{l \in [m]} (L^t AL)_{ll}^2 = \frac{2}{m} |A|_F^2. \quad \blacktriangleleft$$

In the following Lemma we prove bounds on  $\text{Var}_L[\text{Var}_y[pL]]$ . This would help us show that  $\text{Var}_y[pL] = \Theta(\text{Var}[p])$  whp.

► **Lemma 13.**

$$\text{Var}_L[\text{Var}_y[pL]] = \frac{O(1)}{m}.$$

**Proof.** From Lemma 12 we have

$$\begin{aligned} \text{Var}_y[pL] &= |L^t AL|_F^2 + |B^t L|_2^2 + \sum_{l=1}^m (L^t AL)_{ll}^2 \\ &= \sum_{i_1, i_2, j_1, j_2} a_{i_1 j_1} a_{i_2 j_2} \langle c_{i_1}, c_{i_2} \rangle \langle c_{j_1}, c_{j_2} \rangle + \sum_{r_1, r_2} b_{r_1} b_{r_2} \langle c_{r_1}, c_{r_2} \rangle + \sum_{l=1}^m (L^t AL)_{ll}^2 \end{aligned}$$

where

$$\sum_{l \in [m]} (L^t AL)_{ll}^2 = \sum_{\substack{i_1 \neq j_1 \\ i_2 \neq j_2}} \sigma(i_1) \sigma(i_2) \sigma(j_1) \sigma(j_2) a_{i_1 j_1} a_{i_2 j_2} I\{h(i_1)=h(i_2)=h(j_1)=h(j_2)\}$$

Thus we have

$$\begin{aligned} \text{Var}_y[pL] - \mathbb{E}_L[\text{Var}_y[pL]] &= \sum_{(i_1, j_1) \neq (i_2, j_2)} a_{i_1 j_1} a_{i_2 j_2} \langle c_{i_1}, c_{i_2} \rangle \langle c_{j_1}, c_{j_2} \rangle + \sum_{r_1 \neq r_2} b_{r_1} b_{r_2} \langle c_{r_1}, c_{r_2} \rangle \\ &\quad + \sum_{l=1}^m (L^t AL)_{ll}^2 - \frac{3}{m} |A|_F^2. \end{aligned}$$

We skip showing the elaborate yet simple moment calculations but observe that when squared and averaged over  $L$  each term above will have atleast a  $\frac{1}{m}$  term in it. Also the corresponding coefficients can be bounded using Cauchy Schwarz and noting that  $|B|_2^2 \leq 1$  and  $|A|_F^2 \leq 1$ .

Thus

$$\mathbb{E}_L \left( \text{Var}_y[pL] - \mathbb{E}_L[\text{Var}_y[pL]] \right)^2 = O\left(\frac{\text{Var}^2[p]}{m}\right). \quad \blacktriangleleft$$

Now we put together these two Lemmas to show that  $\text{Var}_y[pL] = \Theta(\text{Var}[p])$  whp. We exclude the proof as it is a direct consequence of Chebyshev inequality using Lemma 12 and Lemma 13.

► **Lemma 14.**

$$|\text{Var}_y[pL] - \text{Var}[p]| \leq O\left(\frac{\text{Var}[p]}{m^{1/3}}\right) \text{ wp } \left(1 - \frac{1}{m^{1/3}}\right) \text{ over } L.$$

The following lemma would also be useful. Intuitively it means that  $L$  would not perturb an eigenvalue of  $A$  by a huge amount whp. In fact this would imply that all the eigenvalues of  $A$  would be in the *pseudospectrum* of  $L^t AL$ .

► **Lemma 15.** *Let  $\lambda$  be an eigenvalue of  $A$  and let the unit vector  $v$  be the corresponding eigenvector. Then we have*

$$\mathbb{E}_L |(L^t AL - \lambda I_{m \times m}) L^t v|_2^2 = O\left(\frac{1}{m}\right).$$

**Proof.** Substituting  $Av = \lambda v$ , we have

$$(L^t AL - \lambda I_{m \times m}) L^t v = L^t ALL^t v - L^t Av.$$

Expanding the product  $L^t ALL^t v$  we have,

$$\begin{aligned} L^t ALL^t v &= \sum_{i,j,k \in [n]} a_{i,j} c_i \langle c_j, c_k \rangle v_k \\ L^t ALL^t v - L^t Av &= \sum_{\substack{i \\ j \neq k}} a_{i,j} c_i \langle c_j, c_k \rangle v_k \end{aligned}$$

Thus

$$|L^t ALL^t v - L^t Av|_2^2 = \sum_{\substack{i_1, i_2 \\ j_1 \neq k_1 \\ j_2 \neq k_2}} a_{i_1, j_1} a_{i_2, j_2} v_{k_1} v_{k_2} \langle c_{i_1}, c_{i_2} \rangle \langle c_{j_1}, c_{k_1} \rangle \langle c_{j_2}, c_{k_2} \rangle.$$

A term survives  $\mathbb{E}_\sigma$  only if all the indices  $\{i_1, i_2, j_1, j_2, k_1, k_2\}$  are paired appropriately. However when we take  $\mathbb{E}_h$  since we have  $j_1 \neq k_1, j_2 \neq k_2$  we would see at least a  $1/m$  in every term. Now the corresponding coefficient can be bounded using Cauchy Schwarz and noting that  $|v|_2^2 = 1$  and  $|A|_F^2 \leq 1$ . Thus we have

$$\mathbb{E}_L |(L^t AL - \lambda I_{m \times m}) L^t v|_2^2 = \frac{O(1)}{m}. \quad \blacktriangleleft$$

### 3 The regular case

A polynomial is *regular* if a single variable can't influence its value by a lot. This comes into play when we try to employ the *Invariance principle*. Invariance principle shows that when the underlying variables are changed from Boolean to Gaussian the probability that the polynomial is positive will change by an amount proportional to the maximum influence of a variable over the polynomial. Thus let's assume *regularity* in this section so that we don't incur much error when we switch between Boolean and Gaussian inputs.

#### Proof under the assumption that polynomial is regular

In this section we assume that the degree 2 polynomial  $p(x)$  is  $\tau$ -regular and show that  $\left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right|$  is small whp over  $L$ .

We do this in three steps:

- *Replacement from Boolean to Gaussian for  $p(x)$*  We change the underlying input variables from Boolean to Gaussian. Since we assume the polynomial is regular, we do not incur much error when we do this via Invariance principle.
- *PRG error for Gaussian setting* Once we are in Gaussian setting we show that  $L$  is a *pseudorandom generator* for degree 2 polynomials for Gaussian inputs. The basic idea is that for PTFs in the Gaussian context one only needs to keep track of the top few eigenvalues and the total  $L^2$  norm of rest of rest of the eigenvalues. We show that a Johnson-Lindenstrauss matrix preserves this top eigenvalue structure and the mass in the rest of the eigenvalues.

- *Replacement from Gaussian to Boolean for  $pL$*  Since  $p$  is regular we show that  $pL$  is regular whp too. This let's us go back from Gaussian to Boolean setting via the Invariance principle.

This is depicted in the following equation:

$$\begin{aligned} \left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right| &\leq \left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) \right| \\ &+ \left| \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) \right| \\ &+ \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right|. \end{aligned}$$

The first term  $\left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) \right| \leq O(\epsilon)$  using invariance principle from [16] since we assumed that  $p(x)$  is  $\tau$ -regular where  $\tau = \epsilon^{O(1)}$ .

Now we bound the other two terms in the following sections.

### 3.1 Gaussian PRG $\left| \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) \right|$

In this section we will show that in the Gaussian setting  $p$  cannot distinguish between  $x$  and  $Ly$ . The main idea is that to understand the average sign of a degree 2 polynomial you just need to keep track of the top few eigenvalues and the total mass in the rest of the eigenvalues. This is because either the latter eigenvalues are too small and thus the truncated part overall contributes very little mass to the total polynomial or these eigenvalues are small but do contribute a significant fraction of the total mass (we call this part the eigenregular part), then you could replace all of them by a single Gaussian with the same total mass via the CLT tools used in [3].

Thus let's think of the polynomial  $p$  as the top few eigenvalues and a lump mass of the rest of the eigenvalues. The Johnson-Lindenstrauss like matrix  $L$  we use preserves the top eigenvalue structure of the polynomial and also keeps the eigenregular part still *eigenregular*. It introduces some negligible dependence between the top eigenvalue part and the *eigenregular* part which we remove to begin with to keep them independent.

To begin with assume  $p(x) = x^t A x + B^t x + C$  be a degree 2 multilinear polynomial with  $|A|_F = 1$ . Since  $A$  is a real symmetric matrix, let it be diagonalised as  $A = V \Lambda V^t$ , where  $V$  is an orthonormal matrix whose columns are the eigenvectors of  $A$ . Let the eigenvalues of  $A$  be  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$ . Now let  $k + 1$  be the first index with  $|\lambda_{k+1}| < \delta$  where we will choose  $\delta = \epsilon^{O(1)}$  later on. Since  $\sum_{i \in [n]} \lambda_i^2 = 1$ , we know that  $k \leq \frac{1}{\delta^2} = \left(\frac{1}{\epsilon}\right)^{O(1)} \ll m$ . Let  $V_{\leq k}$

denote the first  $k$  eigenvectors of  $V$  and  $\Lambda_k$  denote the top  $k \times k$  diagonal submatrix of  $\Lambda$  containing the top  $k$  eigenvalues of  $A$ .

► **Definition 16.** Define  $A_1 = V_{\leq k} \Lambda_k V_{\leq k}^t$  to be the top eigenpart of  $A$  and  $A_2 = V_{>k} \Lambda_{k+1}^n V_{>k}^t$  to be the lower eigenpart of  $A$ , we have  $A = A_1 + A_2$ .

Accordingly decompose  $p(x) = q_1(x) + r_1(x)$  where

$$\begin{aligned} q_1(x) &= x^t A_1 x + B^t V_{\leq k} V_{\leq k}^t x + C, \\ r_1(x) &= x^t A_2 x + B^t V_{>k} V_{>k}^t x. \end{aligned}$$

Note that  $q_1(x)$  and  $r_1(x)$  are independent of each other because the columns of  $V_{\leq k}$  are orthogonal to the columns of  $V_{>k}$ . In the following lemma we replace  $r_1(x)$  by just a single Gaussian that has the same mass and thus ignoring the total structure of  $r_1(x)$ . Let  $z$  be an one dimensional Gaussian independent of  $x$ .

► **Lemma 17.** *Given  $\epsilon > 0$  let  $\delta$  be a sufficiently large power of  $\epsilon, \delta = \epsilon^{O(1)}$ . If  $p(x)$  can be written as a sum of two independent polynomials, that is  $p(x) = q_1(x) + r_1(x)$  where  $|\lambda_{max}(r_1)| < \delta$ , then*

$$\left| \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) - \mathbb{E}_{\substack{x \sim \mathcal{N}^n(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}\left(q_1(x) + \sqrt{\text{Var}[r_1]}z\right) \right| < O(\epsilon).$$

**Proof.** We consider two cases:

- Case I - Say  $r_1$  has very small variance, that is  $\sqrt{\text{Var}[r_1(x)]} < \frac{\delta}{\epsilon}$ . Then we can use Lemma 5 to see that the replacement of  $r_1(x)$  by  $\sqrt{\text{Var}[r_1]}z$  will only incur an error of at most  $O(\frac{\delta}{\epsilon})^{\frac{1}{3}}$ . By an appropriate choice of  $\delta = \epsilon^{O(1)}$  that we make later on this error will be  $O(\epsilon)$ .
- Case II - Say  $\sqrt{\text{Var}[r_1(x)]} > \frac{\delta}{\epsilon}$ , then note that every eigenvalue  $\lambda$  of  $r_1(x)$  satisfies  $|\lambda| < \epsilon\sqrt{\text{Var}[r_1]}$ . Such a polynomial all of whose eigenvalues are small compared to its variance are called eigenregular polynomials and we could use Lemma 10 to replace  $r_1(x)$  by  $\sqrt{\text{Var}[r_1]}z$  and incur an error of at most  $O(\epsilon)$ . Note that we are using the independence of  $q_1(x)$  and  $r_1(x)$  in a *convolution* argument used to insert  $q_1$  after applying the CLT.

Thus in either case the lemma holds after an appropriate choice of  $\delta = \epsilon^{O(1)}$ . ◀

To keep the presentation simple henceforth we assume that  $L^t V_{\leq k}$  still has *orthonormal* columns, that is  $V_{\leq k}^t L L^t V_{\leq k} = I_{k \times k}$ . The exact computation proceeds by first using the *Gram Schmidt process* to orthonormalize  $\{L^t v_1, \dots, L^t v_k\}$ . However this would not be very different from the exact analysis because  $L$  approximately preserves inner products and norms whp and we can union bound because  $k$  is a small constant depending on  $\epsilon$ . In particular we have the following lemma.

► **Lemma 18.**

$$\mathbb{E}_L \left| V_{\leq k}^t L L^t V_{\leq k} - I_{k \times k} \right|_F^2 = O\left(\frac{k^2}{m}\right).$$

**Proof.** This is a straightforward computation. Replacing  $I_{k \times k} = V_{\leq k}^t V_{\leq k}$ , we have  $V_{\leq k}^t L L^t V_{\leq k} - I_{k \times k} = V_{\leq k}^t (L L^t - I_{n \times n}) V_{\leq k}$ . This gives,

$$\begin{aligned} \left| V_{\leq k}^t (L L^t - I_{n \times n}) V_{\leq k} \right|_F^2 &= \sum_{a,b \in [k]} \left( \sum_{i_1 \neq i_2} v_a^{i_1} v_b^{i_2} \langle c_{i_1}, c_{i_2} \rangle \right)^2 \\ &= \sum_{a,b \in [k]} \sum_{\substack{i_1 \neq i_2 \\ i_3 \neq i_4}} v_a^{i_1} v_b^{i_2} v_a^{i_3} v_b^{i_4} \langle c_{i_1}, c_{i_2} \rangle \langle c_{i_3}, c_{i_4} \rangle. \end{aligned}$$

This gives

$$\mathbb{E}_L \left| V_{\leq k}^t (L L^t - I_{n \times n}) V_{\leq k} \right|_F^2 \leq O\left(\frac{k^2}{m}\right) \quad \blacktriangleleft$$

Let  $y \sim \mathcal{N}^m(0,1)$  be a Gaussian independent of  $x, z$ . Since Gaussian distribution is invariant to rotations  $V_{\leq k}^t x \sim \mathcal{N}^k(0,1)$  and  $[V_{\leq k}^t L]y \sim \mathcal{N}^k(0,1)$  are identically distributed. Thus  $q_1(x) = [x^t V_{\leq k}] \Lambda_k [V_{\leq k}^t x] + B^t V_{\leq k} [V_{\leq k}^t x] + C$  is *identically* distributed as  $[y^t L^t V_{\leq k}] \Lambda_k [V_{\leq k}^t L y] + B^t V_{\leq k} [V_{\leq k}^t L y] + C$  which is exactly  $q_1(Ly)$ .

Thus we have,

$$\left| \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) - \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}\left(q_1(Ly) + \sqrt{\text{Var}[r_1]}z\right) \right| < O(\epsilon).$$

Let's look at  $p(Ly)$ . We have

$$p(Ly) = y^t L^t A Ly + B^t Ly + C = y^t [L^t V] \Lambda [V^t L] y + B^t Ly + C.$$

Let  $P$  denote the *projection matrix* onto the vector space spanned by  $L^t v_1, \dots, L^t v_k$ . The *projection matrix* can be expressed by the  $m \times m$  matrix  $P \stackrel{\text{def}}{=} L^t V_{\leq k} (V_{\leq k}^t L L^t V_{\leq k})^{-1} V_{\leq k}^t L$ . Note that  $P^2 = P, P^t = P$ . Since  $V_{\leq k}^t L L^t V_{\leq k} = I_{k \times k}$ , this simplifies to  $P = L^t V_{\leq k} V_{\leq k}^t L$ . Now as before we break  $p(Ly)$  into two pieces  $p(Ly) = q_2(y) + r_2(y)$ , wherein

$$\begin{aligned} q_2(y) &= y^t L^t A_1 Ly + B^t LPy + C \\ r_2(y) &= y^t L^t A_2 Ly + B^t L[I-P]y \end{aligned}$$

The goal is to do similar *CLT like analysis* but the problem is that  $q_2(y)$  and  $r_2(y)$  are not independent. We refine  $r_2(y)$  to  $r_3(y)$  to make it independent of  $q_2(y)$  by separating the part of it that correlates with  $q_2(y)$ . That is, define

$$\begin{aligned} r_3(y) &= y^t [I-P] L^t A_2 L [I-P] y + B^t L [I-P] y \\ s(y) &= y^t P L^t A_2 L [I-P] y + y^t L^t A_2 L P y. \end{aligned}$$

Observe that  $r_3(y)$  is independent of  $q_2(y)$ . We have  $p(Ly) = q_2(y) + r_3(y) + s(y)$ . First let's get rid of  $s(y)$  by showing that  $\text{Var}[s]$  is small whp over  $L$  and invoking Lemma 5.

► **Lemma 19.**  $\text{Var}[s] = O\left(\frac{1}{\sqrt{m}}\right)$  wp  $\left(1 - \frac{O(1)}{\sqrt{m}}\right)$  over  $L$ .

**Proof.** It suffices to show that  $|L^t A_2 L P|_F$  is small. Since  $P$  is a projection matrix we have,

$$|L^t A_2 L P|_F^2 = \text{Tr}[L^t A_2 L P L^t A_2 L] = |L^t A_2 L L^t V_{\leq k}|_F^2.$$

Since  $A = A_1 + A_2$ , we have

$$L^t A_2 L = L^t A L - L^t A_1 L = L^t A L - L^t V_{\leq k} \Lambda_k V_{\leq k}^t L.$$

Thus

$$\begin{aligned} L^t A_2 L L^t V_{\leq k} &= (L^t A L) L^t V_{\leq k} - L^t V_{\leq k} \Lambda_k \overbrace{V_{\leq k}^t L L^t V_{\leq k}}^{I_{k \times k}} \\ &= (L^t A L) L^t V_{\leq k} - L^t V_{\leq k} \Lambda_k, \end{aligned}$$

Thus we have

$$|L^t A_2 L L^t V_{\leq k}|_F^2 = \sum_{l=1}^k |(L^t A L) L^t v_l - \lambda_l L^t v_l|_2^2.$$

Now we could use Lemma 15 to bound this. So we have,

$$\mathbb{E}_L |L^t A_2 L L^t V_{\leq k}|_F^2 = O\left(\frac{k}{m}\right).$$

Now the Lemma follows by Markov's inequality and noting that  $\text{Var}[s] = O(|L^t A_2 L P|_F^2)$ . ◀

Now we could apply Lemma 5 to remove  $s$ . That is,

$$\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(q_2(y) + r_3(y)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) \right| \leq O\left(\frac{1}{m^{O(1)}}\right) \text{ wp } \left(1 - \frac{O(1)}{\sqrt{m}}\right) \text{ over } L$$

Now that  $q_2(y)$  and  $r_3(y)$  are independent, to go ahead with the CLT like analysis we first show that the largest eigenvalue of  $r_3(y)$  is at most  $\sqrt{\delta}$ .

► **Lemma 20.**  $\lambda_{\max}[r_3(y)] \leq \sqrt{\delta}$  whp

**Proof.** We want to show that the eigenvalues of  $[I-P](L^t A_2 L)[I-P]$  are small. Its eigenvalues are interlaced into the eigenvalues of  $L^t A_2 L$  because  $[I-P]$  is a projection matrix. Thus it suffices to bound the eigenvalues of  $L^t A_2 L$ , where  $A_2 = V_{>k} \Lambda_{k+1}^n V_{>k}^t$ . Note that  $A_2$  is a symmetric matrix with spectrum  $0^k, \lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_n$ . To bound the eigenvalues of  $L^t A_2 L$  we bound  $\text{Tr}(L^t A_2 L)^4 = |(L^t A_2 L)^2|_F^2$ . We have

$$\begin{aligned} |(L^t A_2 L)^2|_F^2 &= \sum_{j_1 \dots j_8 \in [n]} A_{2j_1 j_2} A_{2j_3 j_4} A_{2j_5 j_6} A_{2j_7 j_8} \langle c_{j_1}, c_{j_5} \rangle \langle c_{j_2}, c_{j_3} \rangle \langle c_{j_4}, c_{j_8} \rangle \langle c_{j_6}, c_{j_7} \rangle \\ \mathbb{E}_L |(L^t A_2 L)^2|_F^2 &= \sum_{j_1, j_2, j_4, j_6 \in [n]} A_{2j_1 j_2} A_{2j_2 j_4} A_{2j_1 j_6} A_{2j_6 j_4} + \frac{O(1)}{m} \\ &= \text{Tr}(A_2^4) + \frac{O(1)}{m} \leq \delta^2 + \frac{O(1)}{m}. \end{aligned}$$

This shows that the maximum absolute eigenvalue of  $r_3(y)$  is at most  $O(\sqrt{\delta})$  whp. This let's us either remove it as a low variance term or apply the CLT machinery on  $r_3(y)$ . ◀

Now that  $q_2$  and  $r_3$  are independent polynomials and since Lemma 20 gives  $\lambda_{\max}[r_3(y)] \leq \sqrt{\delta}$  we could use a slight variant of Lemma 17 to bound the following error:

$$\left| \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}(q_2(y) + \sqrt{\text{Var}[r_3]}z) - \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}(q_2(y) + r_3(y)) \right| < O(\epsilon).$$

We now bound the remaining term that finishes the telescoping for the Gaussian PRG part.

► **Lemma 21.**

$$\left| \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}(q_1(Ly) + \sqrt{\text{Var}[r_1]}z) - \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}(q_2(y) + \sqrt{\text{Var}[r_3]}z) \right| \leq O(\epsilon) \text{ whp}$$

**Proof.** Since  $y$  and  $z$  are independent it suffices to show that  $\text{Var}_y[q_1(Ly) - q_2(y)]$  and  $|\text{Var}[r_1] - \text{Var}[r_3]|$  are both small and invoke Lemma 5 to prove this Lemma.

We have

$$\begin{aligned} q_2(y) - q_1(Ly) &= B^t [LL^t - I] V_{\leq k} V_{\leq k}^t Ly \\ \text{Var} [q_2(y) - q_1(Ly)] &= \left| B [LL^t - I] V_{\leq k} V_{\leq k}^t L \right|_2^2 \end{aligned}$$

Since  $L^t V_{\leq k}$  has orthonormal columns, this simplifies further to

$$\text{Var} [q_2(y) - q_1(Ly)] = \left| B^t [LL^t - I] V_{\leq k} \right|_2^2 = \sum_{l=1}^k \left[ \sum_{j_1 \neq j_2 \in [n]} \langle c_{j_1}, c_{j_2} \rangle b_{j_1} v_l^{j_2} \right]^2$$

$$\text{Thus } \mathbb{E}_L \text{Var} [q_2(y) - q_1(Ly)] = O\left(\frac{k|B|_2}{m}\right).$$

To see that  $\text{Var}[r_1] \approx \text{Var}[r_3]$ , note that  $\text{Var}[r_3] \approx \text{Var}[r_2]$  because  $\text{Var}[s(y)]$  is small as shown above. Now to show that  $\text{Var}[r_1] \approx \text{Var}[r_2]$  we need to show the following:

■  $|A_2|_F \approx |L^t A_2 L|_F$ . This follows from Lemma 12.



- $|B^t V_{>k} V_{>k}^t|_2 \approx |B^t L[I-P]|_2$ . To show this note that  $|B^t V_{>k} V_{>k}^t|_2^2 = \sum_{t=k+1}^n \langle B^t, v_t \rangle^2$ . Since  $L^t V_{\leq k}$  has orthonormal columns, we have

$$|B^t L[I-P]|_2^2 = |B^t L|_2^2 - \sum_{l=1}^k \langle B^t L, L^t v_l \rangle^2.$$

Now this follows by noting that  $L^t$  approximately preserves the norms and inner products of vectors and that since  $k$  is a constant we can union bound. ◀

To summarize we telescoped the Gaussian PRG error as:

$$\begin{aligned} & \left| \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) \right| \leq \\ & \left| \mathbb{E}_{x \sim \mathcal{N}^n(0,1)} \text{sgn}(p(x)) - \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}\left(q_1(Ly) + \sqrt{\text{Var}[r_1]}z\right) \right| \\ & + \left| \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}\left(q_1(Ly) + \sqrt{\text{Var}[r_1]}z\right) - \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}\left(q_2(y) + \sqrt{\text{Var}[r_3]}z\right) \right| \\ & + \left| \mathbb{E}_{\substack{y \sim \mathcal{N}^m(0,1) \\ z \sim \mathcal{N}(0,1)}} \text{sgn}\left(q_2(y) + \sqrt{\text{Var}[r_3]}z\right) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(q_2(y) + r_3(y)) \right| \\ & + \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(q_2(y) + r_3(y)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) \right| \end{aligned}$$

and showed that each of the terms is small whp over  $L$ . This completes the analysis of the Gaussian PRG error term.

Now we move back from Gaussian to Boolean setting to finish the analysis for regular polynomials.

### 3.2 Replacement for $pL$ $\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right|$

We do this change in two parts:

- *Linearize  $pL$  to  $pL_{lin}$*  The application of invariance principle needs the polynomial to be multilinear but  $pL$  need not be multilinear even though  $p$  is. Thus we pre-process  $pL$  to convert to the multilinear polynomial  $pL_{lin}$ .
- *Replacement for  $pL_{lin}$*  Since  $p$  is multilinear, we show that  $pL_{lin}$  is regular whp and then apply the invariance principle.

Thus we split the replacement term for  $pL$  as an error between  $pL, pL_{lin}$  in Gaussian setting and a replacement error for  $pL_{lin}$ . This is depicted in the following equation.

$$\begin{aligned} & \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right| \leq \\ & \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL_{lin}(y)) \right| \\ & + \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL_{lin}(y)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL_{lin}(y)) \right|. \end{aligned}$$

#### 3.2.1 Linearize $pL$ to $pL_{lin}$

$$\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL_{lin}(y)) \right|$$

Note that  $p$  is a multilinear polynomial but  $pL$  need not be multilinear. For example  $L$  could map both  $x_i, x_j$  to  $y_l$  and thus the monomial  $x_i x_j$  to  $y_l^2$ .  $y_l^2$  would be the constant 1 in the boolean case but would be a non linear term in the Gaussian case. However the invariance principle works only for multilinear polynomials. Thus we linearize  $pL$  as follows:

► **Definition 22.**  $pL_{lin}$  is the linearized version of  $pL$  - Every occurrence of a term like  $y_t^2$  is replaced by the constant 1.

Note that  $pL_{lin}$  satisfies the following properties:

- (i)  $pL_{lin}(y) = pL(y)$  when  $y$  is Boolean.
- (ii)  $\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [pL(y) - pL_{lin}(y)] = 0$ .

We bound this linearization error using lemma 23 and lemma 5. Lemma 5 shows that the distributional distance between  $pL, pL_{lin}$  is small if  $|pL - pL_{lin}|_2$  is small and Lemma 23 shows that this is the case.

► **Lemma 23.** *The non-linear part of  $pL$  has small variance with high probability over  $L$ ,*

$$\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [pL(y) - pL_{lin}(y)]^2 < \frac{\text{Var}[p]}{\sqrt{m}} \text{ wp } \left(1 - \frac{O(1)}{\sqrt{m}}\right) \text{ over } L.$$

**Proof.** From the basic definitions of  $p, L, pL_{lin}$  we have,

$$\begin{aligned} p &= \sum_{i,j} a_{ij} x_i x_j + \sum_k b_k x_k + C. \\ pL &= \sum_{i,j} a_{ij} \sigma(i) \sigma(j) y_{h(i)} y_{h(j)} + \sum_k b_k \sigma(k) y_{h(k)} + C. \\ pL - pL_{lin} &= \sum_{t \in [m]} [y_t^2 - 1] \sum_{\substack{i,j: \\ h(i)=h(j)=t}} a_{ij} \sigma(i) \sigma(j). \end{aligned}$$

We calculate the variance of  $pL - pL_{lin}$  by noting that  $\mathbb{E}_{y_t \sim \mathcal{N}^1(0,1)} [y_t^2 - 1]^2 = 2$ ,

$$\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [pL(y) - pL_{lin}(y)]^2 = 2 \sum_{t \in [m]} \sum_{\substack{i,j,k,l: \\ h(i)=h(j)=h(k)=h(l)=t}} a_{ij} a_{kl} \sigma(i) \sigma(j) \sigma(k) \sigma(l).$$

We then calculate the expected variance over the sign  $\sigma$ . This makes a term 0 unless it is paired as  $\{i, j\} = \{k, l\}$ .

$$\mathbb{E}_\sigma \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [pL(y) - pL_{lin}(y)]^2 = 2 \sum_{t \in [m]} \sum_{\substack{i,j: \\ h(i)=h(j)=t}} a_{ij}^2.$$

Now we calculate the expected value of this over the hash function  $h$ .

$$\mathbb{E}_h \mathbb{E}_\sigma \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [pL(y) - pL_{lin}(y)]^2 = 2 \sum_t \sum_{i,j} \frac{1}{m^2} a_{ij}^2 = \frac{2}{m} \|A\|_F^2 \leq \frac{2}{m} \text{Var}[p].$$

The lemma now follows by the Markov inequality applied to  $\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [pL(y) - pL_{lin}(y)]^2$ . ◀

Now we invoke Lemma 5 to finish the bound on the *linearization error*.

$$\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL(y)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL_{lin}(y)) \right| \leq \frac{O(1)}{m^{1/12}} \text{ wp } \left(1 - \frac{O(1)}{\sqrt{m}}\right) \text{ over } L.$$

### 3.2.2 Replacement for $pL_{lin}$

$$\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL_{lin}(y)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL_{lin}(y)) \right|$$

Using Invariance principles from [16], Lemma 6 we have,

$$\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL_{lin}(y)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL_{lin}(y)) \right| \leq O \left[ \frac{\sum_{r=1}^m \text{Inf}_r^2(pL_{lin})}{(\text{Var}[pL_{lin}])^2} \right]^{\frac{1}{9}}.$$

where

$$\text{Inf}_r(pL_{lin}) = 2 \sum_{s \in [m] \setminus r} \left[ \sum_{\substack{h(i)=r \\ h(j)=s}} a_{ij} \sigma(i) \sigma(j) \right]^2 + \left[ \sum_{h(l)=r} b_l \sigma(l) \right]^2$$

A simple but elaborate computation would show that

$$\mathbb{E}_L \left( \sum_{r=1}^m \text{Inf}_r^2(pL_{lin}) - \sum_{i=1}^n \text{Inf}_i^2(p) \right)^2 = O \left( \frac{\text{Var}^4[p]}{m} \right) = \frac{O(1)}{m}.$$

Thus using Chebyshev's inequality we have,

$$\sum_{r=1}^m \text{Inf}_r^2(pL_{lin}) \leq \sum_{i=1}^n \text{Inf}_i^2(p) + \frac{\text{Var}^2[p]}{m^{\frac{1}{3}}} \text{wp} \left( 1 - \frac{O(1)}{m^{\frac{1}{3}}} \right) \text{ over } L.$$

Since  $p$  is  $\tau$ -regular, we have

$$\sum_{j=1}^n \text{Inf}_j^2(p) \leq \max_i \text{Inf}_i(p) \cdot \sum_{i=1}^n \text{Inf}_i(p) \leq 2\tau \text{Var}^2[p].$$

Note that  $\text{Var}[pL_{lin}] = \Theta(\text{Var}[p])$  wp  $(1 - \frac{1}{m^{O(1)}})$  over  $L$  by Lemma 12.

Thus we can bound the *replacement error* for  $pL_{lin}$  as follows:

$$\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(pL_{lin}(y)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL_{lin}(y)) \right| \leq O \left( 2\tau + \frac{1}{\sqrt{m}} \right)^{\frac{1}{9}} \text{wp} \left( 1 - \frac{1}{m^{O(1)}} \right) \text{ over } L.$$

Note that we will be choosing  $\tau = \epsilon^{O(1)}$  and  $m = \frac{1}{\epsilon^{O(1)}}$  which would also ensure that this error is  $\leq O(\epsilon)$ .

## 4 Reduction to the regular case

If the polynomial  $p$  is *not regular* we fix few variables that have large influence to get to a polynomial that is either regular or constant. We note that under  $L$  the high influence variables would most likely have landed in separate bins and thus remain independent. The other variables that land in the same bin as one of these high influence variables do not contribute much to the size of the polynomial.

### Proof that theorem holds for regular polynomials implies it holds for all polynomials

The idea of this reduction is a careful analysis of Lemma 8. This is the standard *Regularity Lemma* from [6].

We look at  $p$  as a decision tree using Lemma 8. If  $\alpha$  denotes a path to the leaf in the decision tree then let  $p_\alpha$  denote the restriction polynomial along the path. Let  $S(\alpha)$  denote the set of variables that are set along the path  $\alpha$ .

**Note**

$p_\alpha$  is only a function  $n - |S(\alpha)|$  coordinates that are not set along the decision tree path  $\alpha$ . For the ease of notation we look at it still as a function of  $n$  coordinates, wherein it just ignores the coordinates that are already fixed along  $\alpha$ .

Averaging over all the decision tree paths we have,

$$\mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) = \mathbb{E}_\alpha \left[ \mathbb{E}_{x \sim \{\pm 1\}^{[n] \setminus S(\alpha)}} \text{sgn}(p_\alpha(x)) \right].$$

Thus let's split the change from  $x$  to  $Ly$ ,  $|\mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y))|$  into two parts:

- *Change  $x$  to  $Ly$  only at decision tree leaves* Here we average the values along the decision paths based on the values set by  $\alpha$  and only make the change  $x$  to  $Ly$  at the leaves. Note that analyzing this would be easier since the disagreement is only at the leaves and we know that the leaves are either regular or constant.
- *Changing  $Ly$  at leaves to  $Ly$  overall* Here we bound the error we incur by going from  $\alpha$  setting the values along the decision path followed by  $Ly$  setting the values at the leaves to  $Ly$  setting the values overall. To analyse this we will be introducing a new distribution  $Ly'$  that only disagrees with  $Ly$  on very few variables.

This split is depicted in the following equation,

$$\begin{aligned} & \left| \mathbb{E}_{x \sim \{\pm 1\}^n} \text{sgn}(p(x)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right| \leq \\ & \quad \left| \mathbb{E}_\alpha \left[ \mathbb{E}_{x \sim \{\pm 1\}^{[n] \setminus S(\alpha)}} \text{sgn}(p_\alpha(x)) \right] - \mathbb{E}_\alpha \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly)) \right] \right| \\ & \quad + \left| \mathbb{E}_\alpha \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly)) \right] - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y)) \right| \end{aligned}$$

**4.1 Leaf change**

$$\left| \mathbb{E}_\alpha \left[ \mathbb{E}_{x \sim \{\pm 1\}^{[n] \setminus S(\alpha)}} \text{sgn}(p_\alpha(x)) \right] - \mathbb{E}_\alpha \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly)) \right] \right|$$

Using Jensen's inequality, also known here as the triangle inequality we have

$$\begin{aligned} & \left| \mathbb{E}_\alpha \left[ \mathbb{E}_{x \sim \{\pm 1\}^{S^c}} \text{sgn}(p_\alpha(x)) \right] - \mathbb{E}_\alpha \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly)) \right] \right| \leq \\ & \quad \mathbb{E}_\alpha \left| \mathbb{E}_{x \sim \{\pm 1\}^{S^c}} \text{sgn}(p_\alpha(x)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly)) \right|. \end{aligned}$$

By the regularity lemma we know that with probability atleast  $(1-\theta)$  (where  $\theta = \frac{1}{\sqrt{m}}$ ), the leaf  $p_\alpha$  is either  $\tau$ -regular (where  $\tau = \epsilon^{O(1)}$ ) or  $p_\alpha$  is almost constant (that is  $\text{Var}[p_\alpha] \leq \frac{1}{\sqrt{m}}$ ). Now

- *Regular* If  $p_\alpha$  is  $\tau$ -regular, we could just bound the error by  $\epsilon$  using our results from the previous section.
- *Constant* If  $p_\alpha$  is almost constant, then wlog it is of the form  $1+q$  where  $\text{Var}[q] = \frac{1}{\sqrt{m}}$ , thus  $\text{sgn}(p_\alpha(x)) = \text{sgn}(p_\alpha(Ly))$  with probability  $1 - \frac{1}{\sqrt{m}}$ .

In either case we have

$$\begin{aligned} & \left| \mathbb{E}_\alpha \left[ \mathbb{E}_{x \sim \{\pm 1\}^{[n] \setminus S(\alpha)}} \text{sgn}(p_\alpha(x)) \right] - \mathbb{E}_\alpha \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly)) \right] \right| \\ & \leq (1-\theta)\epsilon + 2\theta \text{ wp } \left( 1 - \frac{1}{m^{O(1)}} \right) \text{ over } L \\ & = \epsilon + \frac{1}{m^{O(1)}} \text{ wp } \left( 1 - \frac{1}{m^{O(1)}} \right) \text{ over } L \text{ since } \theta = \frac{1}{\sqrt{m}}. \end{aligned}$$

## 4.2 Changing $Ly$ overall

$$|\mathbb{E}_\alpha[\mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly))] - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y))|$$

To bound this term we introduce a new intermediate distribution  $Ly'$ . As long as  $L$  doesn't hash collide two high influence decision path  $\alpha$  variables,  $Ly$  and  $Ly'$  are *identically* distributed. In fact they agree on all variables except on those variables that hash collide with the decision path variables. On these variables  $Ly'$  just assigns the decision path variable's value to every other variable that lands in its bin.

► **Definition 24.** We define  $Ly'$  as follows:

$$(Ly')_i = \begin{cases} x_i & i \in S(\alpha) \\ (Ly)_i & i \in LS(\alpha)^c \\ x_{h^{-1}(h(i))} & i \in LS(\alpha) \setminus S(\alpha). \end{cases}$$

where  $LS = \{i \in [n] : h(i) \in h(S)\}$  and for  $j \in [m]$ ,  $h^{-1}(j)$  is the smallest  $i \in [n]$  such that  $h(i) = j$ . This essentially fixes all the bits that hash collide with the decision path variables.

Now we split the error in changing  $Ly$  on leaves to  $Ly$  overall further into two steps.

- *Changing  $Ly$  on leaves to  $Ly'$*  Here we observe that  $Ly$  and  $Ly'$  agree everywhere except the variables that hash collide with the decision path variables. The depth of the tree is small  $D = \log m$ , thus each variable collides with a decision path variable with very small probability  $\frac{\log m}{m}$ . Thus this difference amounts to a negligible fraction of the variance of the polynomial.
- *Changing from  $Ly'$  to  $Ly$  overall* Here we only need to bound the probability that two decision path variables along  $\alpha$  don't hash collide. As long as thing doesn't happen  $Ly$  and  $Ly'$  are identically distributed.

This is depicted in the following equation:

$$\begin{aligned} |\mathbb{E}_\alpha[\mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly))] - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y))| \leq \\ |\mathbb{E}_\alpha[\mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly))] - \mathbb{E}_\alpha[\mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly'))]| \\ + |\mathbb{E}_\alpha[\mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly'))] - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y))| \end{aligned}$$

### 4.2.1 Changing from $Ly'$ to $Ly$ overall

$$|\mathbb{E}_\alpha[\mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_\alpha(Ly'))] - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(pL(y))|$$

Here we only need to bound the probability that there is no hash collision on  $S(\alpha)$ . This is because if  $h$  does not have a hash collision on  $S(\alpha)$ , then  $Ly$  and  $Ly'$  are identically distributed. That is,

$$\begin{aligned} \left| \mathbb{E}_\alpha[\mathbb{E}_{y \sim \{\pm 1\}^{[m] \setminus h(S(\alpha))}} \text{sgn}(p_\alpha(Ly'))] - \mathbb{E}_{y_{h[S(\alpha)]}[\mathbb{E}_{y \sim \{\pm 1\}^{[m] \setminus h(S(\alpha))}} \text{sgn}(pL(y))]} \right| \\ \leq 2 \Pr_\alpha(|h[S(\alpha)]| \neq |S(\alpha)|). \end{aligned}$$

Note that the above equation is for a *fixed hash map*  $h$ . The probability is over the choice of random paths  $\alpha$  of the decision tree but for this fixed  $h$ . It is the probability that a *random decision tree path* sees a collision wrt this fixed hash map  $h$ .

We show that for most hash maps (whp over  $L$ ) this term is very small.

► **Lemma 25.**

$$\mathbb{E}_h \left[ \Pr_{\alpha} \left( |h[S(\alpha)]| \neq |S(\alpha)| \right) \right] \leq \frac{D^2}{m}.$$

**Proof.** Interchange the expectations and fix a depth  $D$  decision tree path and observe that the probability that a random  $h$  has a collision along this path is at most  $\frac{\binom{D}{2}}{m}$ . ◀

Hence by Markov's inequality we have,

$$\Pr_{\alpha} \left( |h[S(\alpha)]| \neq |S(\alpha)| \right) \leq \frac{D^2}{\sqrt{m}} \text{ wp } \left( 1 - \frac{O(1)}{\sqrt{m}} \right) \text{ over } L.$$

Note that  $D$  here is poly(log  $m$ ).

#### 4.2.2 Changing $Ly$ on leaves to $Ly'$

$$\left| \mathbb{E}_{\alpha} \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly)) \right] - \mathbb{E}_{\alpha} \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly')) \right] \right|$$

Note that  $Ly$  and  $Ly'$  agree everywhere except the variables that hash collide with the decision path variables. Thus the part of the polynomial that disagrees wrt  $Ly$  and  $Ly'$  only contributes little mass to the total polynomial. However in order to use this fact to bound this term we would need to move back to Gaussian setting so that we could use Anticoncentration like ideas.

Jensen's inequality, also known here as the triangle inequality gives

$$\begin{aligned} & \left| \mathbb{E}_{\alpha} \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly)) \right] - \mathbb{E}_{\alpha} \left[ \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly')) \right] \right| \\ & \leq \mathbb{E}_{\alpha} \left[ \left| \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly')) \right| \right]. \end{aligned}$$

Now we move back to Gaussian setting by doing a replacement on both  $p_{\alpha}(Ly), p_{\alpha}(Ly')$ .

$$\begin{aligned} & \mathbb{E}_{\alpha} \left[ \left| \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly)) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly')) \right| \right] \leq \\ & \mathbb{E}_{\alpha} \left[ \left| \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_{\alpha}(Ly)) \right| \right] \\ & + \mathbb{E}_{\alpha} \left[ \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_{\alpha}(Ly)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_{\alpha}(Ly')) \right| \right] \\ & + \mathbb{E}_{\alpha} \left[ \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_{\alpha}(Ly')) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly')) \right| \right]. \end{aligned}$$

#### Replacement Errors:

Note that with probability  $(1-\theta)$  (where  $\theta = \frac{1}{\sqrt{m}}$ ) the decision tree path  $\alpha$  is such that  $p_{\alpha}(\cdot)$  is either regular or almost constant. We use similar analysis as done in the previous section to bound both the replacement error terms as follows:

$$\mathbb{E}_{\alpha} \left[ \left| \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_{\alpha}(Ly)) \right| \right] \leq O\left(\epsilon + \frac{1}{m^{O(1)}}\right) \text{ wp } \left( 1 - \frac{O(1)}{\sqrt{m}} \right) \text{ over } L$$

and

$$\mathbb{E}_{\alpha} \left[ \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_{\alpha}(Ly')) - \mathbb{E}_{y \sim \{\pm 1\}^m} \text{sgn}(p_{\alpha}(Ly')) \right| \right] \leq O\left(\epsilon + \frac{1}{m^{O(1)}}\right) \text{ wp } \left( 1 - \frac{O(1)}{\sqrt{m}} \right) \text{ over } L$$

**Change  $Ly$  on leaves to  $Ly'$  in Gaussian setting**

We just need to bound  $\mathbb{E}_\alpha \left[ \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_\alpha(Ly)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_\alpha(Ly')) \right| \right]$ .

We show that  $\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [p_\alpha(Ly) - p_\alpha(Ly')]^2$  is small in Lemma 26 and then invoke Lemma 5 to bound  $\left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_\alpha(Ly)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_\alpha(Ly')) \right|$ .

We expect  $\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [p_\alpha(Ly) - p_\alpha(Ly')]^2$  to be small because  $p_\alpha(Ly), p_\alpha(Ly')$  agree on all terms except those that contain a variable that hash collides with the decision tree path. Since this is a low probability event ( $\frac{D}{m} = \frac{(\log m)^{O(1)}}{m}$ ), we expect the overall mass in this difference  $p_\alpha(Ly) - p_\alpha(Ly')$  to be very small.

► **Lemma 26.**

$$\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [p_\alpha(Ly) - p_\alpha(Ly')]^2 \leq \frac{1}{\sqrt{m}} \text{wp} \left( 1 - \frac{O(1)}{\sqrt{m}} \right) \text{ over } L.$$

**Proof.** Let  $W = LS(\alpha) \setminus S(\alpha)$ . Expanding out the terms we have

$$\begin{aligned} p_\alpha(Ly) - p_\alpha(Ly') &= 2 \sum_{j_1 \in S, j_2 \in W} a_{j_1 j_2} x_{j_1} [\sigma(j_2) y_{h(j_2)} - x_{h^{-1}[h(j_2)]}] \\ &\quad + \sum_{l \in W} b_l [\sigma(l) y_{h(l)} - x_{h^{-1}(h(l))}] \\ &\quad + 2 \sum_{j_1 \in LS^c, j_2 \in W} \sigma(j_1) a_{j_1 j_2} y_{h(j_1)} [\sigma(j_2) y_{h(j_2)} - x_{h^{-1}[h(j_2)]}] \\ &\quad + \sum_{j_1, j_2 \in W} a_{j_1 j_2} [\sigma(j_1) \sigma(j_2) y_{h(j_1)} y_{h(j_2)} - x_{h^{-1}[h(j_1)]} x_{h^{-1}[h(j_2)]}]. \end{aligned}$$

Note that  $j_2$  needs to be in  $W$  in all of the terms above. This is a very low probability event. In fact  $Pr_h(j_2 \in W) = \frac{D}{m}$  where  $D = |S(\alpha)|$  is the depth of the decision tree and is chosen to be  $(\log m)^{O(1)}$ .

$$\mathbb{E}_L \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [p_\alpha(Ly) - p_\alpha(Ly')]^2 = O\left(\frac{DVar[p_\alpha]}{m}\right) = O\left(\frac{(\log m)^{O(1)}}{m}\right).$$

Thus

$$\mathbb{E}_{y \sim \mathcal{N}^m(0,1)} [p_\alpha(Ly) - p_\alpha(Ly')]^2 < \frac{(\log m)^{O(1)}}{\sqrt{m}} \text{wp} \left( 1 - \frac{O(1)}{\sqrt{m}} \right) \text{ over } L. \quad \blacktriangleleft$$

Now we invoke Lemma 5 to finish bounding this term,

$$\mathbb{E}_\alpha \left[ \left| \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_\alpha(Ly)) - \mathbb{E}_{y \sim \mathcal{N}^m(0,1)} \text{sgn}(p_\alpha(Ly')) \right| \right] < \left( \frac{(\log m)^{O(1)}}{\sqrt{m}} \right)^{1/6} \text{wp} \left( 1 - \frac{O(1)}{\sqrt{m}} \right) \text{ over } L$$

---

**References**

- 1 J. Wright A. Carbery. Distributional and  $l^q$  norm inequalities for polynomials over convex bodies in  $\mathbb{R}^n$ . *Mathematical Research Letters*, 2001.
- 2 Richard Beigel. The polynomial method in circuit complexity. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference, San Diego, CA, USA, May 18-21, 1993*, pages 82–95. IEEE Computer Society, 1993. doi:10.1109/SCT.1993.336538.

- 3 Anindya De, Ilias Diakonikolas, and Rocco A. Servedio. Deterministic approximate counting for degree-2 polynomial threshold functions. *CoRR*, abs/1311.7105, 2013. [arXiv:1311.7105](https://arxiv.org/abs/1311.7105).
- 4 Anindya De and Rocco A. Servedio. Efficient deterministic approximate counting for low-degree polynomial threshold functions. *CoRR*, abs/1311.7178, 2013. [arXiv:1311.7178](https://arxiv.org/abs/1311.7178).
- 5 I. Diakonikolas, P. Gopalan, R. Jaiswal, R.A. Servedio, and E. Viola. Bounded independence fools halfspaces. *SIAM Journal on Computing*, 2010.
- 6 Ilias Diakonikolas, Rocco A. Servedio, Li-Yang Tan, and Andrew Wan. A regularity lemma, and low-weight approximators, for low-degree polynomial threshold functions. *CoRR*, abs/0909.4727, 2009. [arXiv:0909.4727](https://arxiv.org/abs/0909.4727).
- 7 Parikshit Gopalan, Daniel M. Kane, and Raghu Meka. Pseudorandomness via the discrete fourier transform. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 903–922. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.60.
- 8 Jelani Nelson Ilias Diakonikolas, Daniel M. Kane. Bounded independence fools degree-2 threshold functions. *Foundations of Computer Science (FOCS)*, 2010.
- 9 Daniel M. Kane.  $k$ -independent gaussians fool polynomial threshold functions. *Conference on Computational Complexity (CCC)*, 2011.
- 10 Daniel M. Kane. A small prg for polynomial threshold functions of gaussians. *Symposium on the Foundations Of Computer Science (FOCS)*, 2011.
- 11 Daniel M. Kane. A structure theorem for poorly anticoncentrated gaussian chaoses and applications to the study of polynomial threshold functions. *CORR*, 2012. URL: <http://arxiv.org/abs/1204.0543>.
- 12 Daniel M. Kane. A pseudorandom generator for polynomial threshold functions of gaussians with subpolynomial seed length. *Conference on Computational Complexity (CCC)*, 2014.
- 13 Daniel M. Kane. A polylogarithmic PRG for degree 2 threshold functions in the gaussian setting. In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPICs*, pages 567–581. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPICs.CCC.2015.567.
- 14 Adam R. Klivans and Rocco A. Servedio. Learning DNF in time  $2^{\tilde{O}(n^{1/3})}$ . In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 258–265. ACM, 2001. doi:10.1145/380752.380809.
- 15 Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. *CoRR*, abs/0910.4122, 2009. [arXiv:0910.4122](https://arxiv.org/abs/0910.4122).
- 16 E. Mossel, R. O’Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. *ArXiv Mathematics e-prints*, 2005. [arXiv:math/0503503](https://arxiv.org/abs/math/0503503).
- 17 Alexander A. Sherstov. Separating  $ac^0$  from depth-2 majority circuits. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 294–301. ACM, 2007. doi:10.1145/1250790.1250834.
- 18 Valentine Kabanets and Zhenjian Lu. Nisan-wigderson pseudorandom generators for circuits with polynomial threshold gates. *ECCC*, <https://ecc.weizmann.ac.il/report/2018/012/>, 2018. URL: <https://ecc.weizmann.ac.il/report/2018/012/>.



# A New Approach for Constructing Low-Error, Two-Source Extractors

**Avraham Ben-Aroya**<sup>1</sup>

The Blavatnik School of Computer Science, Tel-Aviv University  
Tel Aviv 69978, Israel

**Eshan Chattopadhyay**<sup>2</sup>

Department of Computer Science, Cornell University and School of Mathematics, IAS  
Ithaca, NY 14850, USA; Princeton, NJ 08540, USA  
eshanc@ias.edu

**Dean Doron**<sup>3</sup>

The Blavatnik School of Computer Science, Tel-Aviv University  
Tel Aviv 69978, Israel  
deandoron@mail.tau.ac.il

**Xin Li**<sup>4</sup>

Department of Computer Science, Johns Hopkins University  
Baltimore, MD 21218, USA  
lixints@cs.jhu.edu

**Amnon Ta-Shma**<sup>5</sup>

The Blavatnik School of Computer Science, Tel-Aviv University  
Tel Aviv 69978, Israel  
amnon@tau.ac.il

---

## Abstract

Our main contribution in this paper is a new reduction from explicit two-source extractors for polynomially-small entropy rate and negligible error to explicit  $t$ -non-malleable extractors with seed-length that has a good dependence on  $t$ . Our reduction is based on the Chattopadhyay and Zuckerman framework (STOC 2016), and surprisingly we dispense with the use of resilient functions which appeared to be a major ingredient there and in follow-up works. The use of resilient functions posed a fundamental barrier towards achieving negligible error, and our new reduction circumvents this bottleneck.

The parameters we require from  $t$ -non-malleable extractors for our reduction to work hold in a non-explicit construction, but currently it is not known how to explicitly construct such extractors. As a result we do not give an unconditional construction of an explicit low-error two-source extractor. Nonetheless, we believe our work gives a viable approach for solving the important problem of low-error two-source extractors. Furthermore, our work highlights an existing barrier in constructing low-error two-source extractors, and draws attention to the dependence of the parameter  $t$  in the seed-length of the non-malleable extractor. We hope this work would lead to further developments in explicit constructions of both non-malleable and two-source extractors.

---

<sup>1</sup> Supported by the Israel Science Foundation grant no. 994/14.

<sup>2</sup> Supported by NSF grants CCF-1526952, CCF-1412958 and the Simons Foundation. Part of this work was done when the author was a graduate student in UT Austin and while visiting the Simons Institute for the Theory of Computing at UC Berkeley.

<sup>3</sup> Supported by the Israel Science Foundation grant no. 994/14. This work was done in part while visiting the Simons Institute for the Theory of Computing at UC Berkeley.

<sup>4</sup> Supported by NSF Grant CCF-1617713.

<sup>5</sup> Supported by the Israel Science Foundation grant no. 994/14. This work was done in part while visiting the Simons Institute for the Theory of Computing at UC Berkeley.



© Avraham Ben-Aroya, Eshan Chattopadhyay,  
Dean Doron, Xin Li, and Amnon Ta-Shma;  
licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 3; pp. 3:1–3:19



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**2012 ACM Subject Classification** Theory of computation → Pseudorandomness and derandomization

**Keywords and phrases** Two-Source Extractors, Non-Malleable Extractors, Pseudorandomness, Explicit Constructions

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.3

**Acknowledgements** We are grateful to Gil Cohen for discussions regarding [20] in the early stages of our work.

## 1 Introduction

A two-source extractor hashes samples from two *independent* weak sources into one output whose distribution is close to uniform. Formally, we say a distribution  $X$  is an  $(n, k)$  source if  $X$  is distributed over  $\{0, 1\}^n$  and its min-entropy is at least  $k$  (i.e., all strings in its support have probability mass at most  $2^{-k}$ ). An  $((n_1, k_1), (n_2, k_2), \varepsilon)$  two-source extractor is a function  $E: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$  that maps any pair of independent  $(n_1, k_1)$  and  $(n_2, k_2)$  sources  $X_1, X_2$  to a distribution  $E(X_1, X_2)$  which is  $\varepsilon$ -close to  $U_m$ , the uniform distribution over  $\{0, 1\}^m$ .

Non-explicitly there are  $((n, k), (n, k), \varepsilon)$  two-source extractors as long as  $k \geq \log n + 2 \log(\frac{1}{\varepsilon}) + O(1)$ . More generally,

► **Fact 1.** *Assume  $k_1 + k_2 \geq \log(2^{k_1} n_1 + 2^{k_2} n_2) + 2 \log(\frac{1}{\varepsilon}) + O(1)$ . Then, there exists a (non-explicit)  $((n_1, k_1), (n_2, k_2), \varepsilon)$  two-source extractor  $E: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$ .*

Finding such *explicit* constructions is a long-standing, important and challenging problem. A key parameter is the error  $\varepsilon$  obtained by the two-source extractor. Research in the area can be divided into three regimes:

**Very large error:** Finding explicit two-source extractors with any error smaller than 1 (i.e., any non-trivial error) is already very challenging and is essentially equivalent to finding an explicit *bipartite* Ramsey graph. A  $K$  *Ramsey graph* is a graph that contains no monochromatic set (i.e., a clique or an independent set) of size  $K$ ; a  $K$  *bipartite Ramsey graph* is a bipartite graph with no bipartite monochromatic sets of size  $K$ . A  $K = 2^k$  bipartite Ramsey graph over  $2N = 2 \cdot 2^n$  vertices, is essentially equivalent to an  $((n, k), (n, k), \varepsilon)$  two-source extractor, with  $\varepsilon = \varepsilon(n) < 1$ .

A long line of research was devoted to explicitly constructing Ramsey graphs [1, 30, 22, 12, 23, 31, 2, 24, 3], bipartite Ramsey graph [4, 5, 17], and two-source extractors [11, 34, 7]. Two years ago, Cohen [17] constructed a  $K$  bipartite Ramsey graph over  $2N$  vertices with  $\log K = \text{polylog}(\log N)$ . This corresponds to an  $((n, k), (n, k), \varepsilon)$  two-source extractor, with  $k = \text{polylog } n$  and some non-trivial error  $\varepsilon$ . Independently, Chattopadhyay and Zuckerman [10] gave another construction that gives about the same bipartite Ramsey graphs, but with smaller error. We discuss this next.

**Medium size error:** Chattopadhyay and Zuckerman constructed an efficient  $((n, k), (n, k), \varepsilon)$  two-source extractor, with  $k = \text{polylog } n$  and running time polynomial in  $1/\varepsilon$ . Several improvements followed, including [29, 27]. Currently, following [6, 18, 28], the best explicit construction achieves  $k = O(\log n \log \log n)$  which is pretty close to the optimal  $\Omega(\log n)$  bound.

All these constructions have running time which is at best polynomial in  $1/\varepsilon$ , and as we explain below this seems to be inherent to the approach that is taken. In contrast, non-explicit constructions may have exponentially small error in the entropy  $k$  of the two sources. Similarly, these constructions usually output few close-to-uniform bits, while non-explicitly, almost all of the entropy can be extracted.

**Exponentially small error:** There are several explicit two-source extractors constructions with exponentially small error:

1. The inner-product function gives a simple construction when  $k > n/2$  [11].
2. Bourgain [7] gave a two-source extractor construction for  $k = (\frac{1}{2} - \alpha)n$ , for some small constant  $\alpha > 0$ .
3. Raz [34] constructed an  $((n_1, k_1), (n_2, k_2), \varepsilon)$  two-source extractor that has an unbalanced entropy requirement; the first source is long (of length  $n_1$ ) and very weak ( $k_1$  can be as small as  $\log \log n_1 + O(1)$ ), the second source is short (of length  $O(\log n_1)$ ) and somewhat dense with  $k_2 \geq \alpha n_2$ , for some constant  $\alpha > \frac{1}{2}$ .

On the positive side, all of these constructions have exponentially small error (in Raz's extractor, the error is exponentially small in the smaller entropy). On the negative side, however, in all of these constructions one of the sources is required to have entropy rate close to half, i.e., the entropy of the source has to be at least  $(\frac{1}{2} - \alpha)n > 0.49n$ .

To summarize:

- Current explicit constructions of low-error, two-source extractors require one source to have entropy rate close to half, and,
- There are explicit two-source extractors that work with astonishingly small min-entropy, but currently they only handle large error, or, more precisely, their running time is polynomial in  $1/\varepsilon$ .

As we shall see shortly, there is a good reason for the two barriers that are represented in the above two items. The goal of this paper is to present a new approach for bypassing these barriers.

## 1.1 Extractors and Entropy-Rate Half

Let us start with the rate-half barrier for low-error constructions. For that we compare two-source extractors with *strong seeded extractors*.

► **Definition 2.**  $E: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  is a strong  $(k, \varepsilon)$  extractor if for every  $(n, k)$  source  $X$ ,  $(Y, E(X, Y))$  is  $\varepsilon$ -close to  $Y \times U_m$ , where  $Y$  is uniformly distributed over  $\{0, 1\}^d$  and is independent of  $X$ .

A seeded extractor  $E$  must have seed length  $d \geq \log n + 2 \log(\frac{1}{\varepsilon}) - O(1)$  [33]. In essence, the error of a *seeded* extractor has two origins:

- The fraction  $\varepsilon_1$  of bad seeds for which  $E(X, y)$  is  $\varepsilon_2$ -far from uniform, and,
- The distance  $\varepsilon_2$  between  $E(X, y)$  and  $U_m$  for good seeds.

These two errors can be very different, for example, it might be the case that for half the seeds the error is extremely small, and then  $\varepsilon_1$  is constant and  $\varepsilon_2$  is tiny, or vice versa. In the terminology of a seeded extractor, these two errors are unified to one parameter  $\varepsilon$ . In the two-source extractor notation these two errors are essentially *separated*, where  $2^{k_2}$  is, roughly, the number of bad seeds making  $\varepsilon_1 \approx 2^{k_2 - n_2}$ , where  $\varepsilon$  of the two-source extractor represents the  $\varepsilon_2$  above. More formally:

► **Fact 3.** Suppose  $E: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}$  is an  $((n, k), (d, d'), \varepsilon_2)$  two-source extractor. Then,  $E$  is a strong  $(k, \varepsilon = \varepsilon_1 + \varepsilon_2)$  extractor, for  $\varepsilon_1 = 2^{d'+1-d}$ , and furthermore, for every  $(n, k)$  source  $X$ ,

$$\Pr_{y \in \{0, 1\}^d} [E(X, y) \not\approx_{\varepsilon_2} U_1] \leq \varepsilon_1.$$

**Proof.** Let  $X$  be an  $(n, k)$  source and let  $B \subseteq \{0, 1\}^d$  so that for every  $y \in B$ ,  $E(X, y) \not\approx_{\varepsilon_2} U_1$ . Partition  $B = B_0 \cup B_1$  where  $y \in B_z$  if the  $\varepsilon_2$  bias is towards  $z$ . Assume towards contradiction that  $|B_z| \geq 2^{d'}$  for some  $z$  and consider the flat distribution  $Y$  over the set  $B_z$ . Thus,  $H_\infty(Y) \geq d'$  so  $E(X, Y) \approx_{\varepsilon_2} U_1$  but by our definition,  $E(X, Y)$  is biased towards  $z$  – a contradiction. Altogether,  $|B| \leq 2^{d'+1}$  so  $\varepsilon_1 \leq |B|/2^d = 2^{d'+1-d}$ . ◀

The lower bound  $d \geq \log n + 2 \log(\frac{1}{\varepsilon}) - O(1)$  imposed on extractors, does not reveal which of the two errors forces  $d$  to be large. Stating it more precisely, define a  $(k, \varepsilon_1, \varepsilon_2)$  function  $E: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  so that for every  $(n, k)$  source  $X$ ,  $\Pr_{y \in \{0, 1\}^d} [E(X, y) \not\approx_{\varepsilon_2} U_1] \leq \varepsilon_1$ . What is the dependence of  $d$  on  $\varepsilon_1$  and  $\varepsilon_2$ ?

The existence of  $((n, k), (d = n, d' = O(\log n)), \varepsilon)$  two-source extractors, implies that the dependence of  $d$  on  $\varepsilon_1$  might be very close  $1 \cdot \log \frac{1}{\varepsilon_1}$ . On the other hand, the dependence of  $d$  on  $\varepsilon_2$  is larger,  $d \geq d' \geq 2 \log \frac{1}{\varepsilon_2}$ , since we can view  $E$  as a strong  $(d', \varepsilon_2)$  extractor  $\{0, 1\}^d \times \{0, 1\}^k \rightarrow \{0, 1\}$  and  $d' \geq 2 \log \frac{1}{\varepsilon_2}$  is again a lower bound [33]. Thus, the two-source extractor terminology allows a finer characterization of the quality of an extractor, separating the two errors  $\varepsilon_1$  and  $\varepsilon_2$  above.

Looking at it that way we see why rate-half is a natural barrier: An extractor with seed length dependence  $2 \log(\frac{1}{\varepsilon})$  guarantees that out of the  $D = 2^d$  possible seeds, at most  $D^{\frac{1}{2}+\beta}$  are  $D^{-\beta}$  bad. Thus, one can get an explicit two-source extractor, where the seed has some constant density  $\frac{1}{2} + \beta$ , and exponentially small error, by constructing an explicit strong seeded extractor with seed length dependence  $(2 + \gamma) \log(\frac{1}{\varepsilon})$  for some small constant  $\gamma$ . Constructing a two-source extractor with  $d'/d$  below half necessarily means using techniques that do not apply to strong seeded extractors. Bourgain achieves that in an ingenious way, by using additive combinatorics together with the inner product function, but, at least so far, this approach can only handle min-entropies slightly below half.

## 1.2 The CZ Approach

We now explain the main ideas in the construction of the two-source extractor of [10] and the bottleneck for achieving smaller error. The CZ construction builds upon two main ingredients: the existence of explicit non-malleable extractors and resilient functions, and we recall both now.

► **Definition 4.**  $E: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  is a strong  $(k, \varepsilon)$   $t$ -non-malleable ( $n.m.$ ) extractor, if for every  $(n, k)$  source  $X$  and every  $t$  functions  $f_1, \dots, f_t: \{0, 1\}^d \rightarrow \{0, 1\}^d$  with no fixed-points<sup>6</sup> it holds that

$$|(Y, E(X, Y), E(X, f_1(Y)), \dots, E(X, f_t(Y))) - (Y, U_m, E(X, f_1(Y)), \dots, E(X, f_t(Y)))| \leq \varepsilon,$$

where  $Y$  is uniformly distributed over  $\{0, 1\}^d$  and is independent of  $X$  and  $U_m$  is the uniform distribution over  $\{0, 1\}^m$ .

<sup>6</sup> That is, for every  $i$  and every  $x$ , we have  $f_i(x) \neq x$ .

In words and roughly speaking, this means that there are many good seeds, and for a good seed  $y$ ,  $E(X, y)$  is close to uniform even given the value of  $E$  on  $t$  other seeds  $f_1(y), \dots, f_t(y)$  maliciously chosen by an adversary. Said differently, if we build a table with  $D = 2^d$  rows, and put  $E(X, i)$  in the  $i$ -th row, then rows of good seeds are close to uniform, and, furthermore, those good rows are close to being  $t$ -wise independent, in the sense that every  $t$  good rows are  $\approx t\varepsilon$  close to uniform (see Lemma 10).

A *resilient function* is a nearly-balanced function  $f: \{0, 1\}^D \rightarrow \{0, 1\}$  whose output cannot be heavily influenced by any small set of  $q$  “bad” bits. We think of the bad bits as a coalition of malicious players trying to bias the output *after seeing* the  $D - q$  coin tosses of the honest players (the honest players toss independent random coin). The function  $f$  is  $(q, t)$  resilient if it is resilient even when there are  $q$  bad players and even when the honest players are only  $t$ -wise independent.

Now, let  $X_1$  and  $X_2$  be two independent  $(n, k)$  sources. The starting point of [10] is to use a  $t$ -non-malleable extractor  $E$  with error  $\varepsilon_1$  and seed length  $d_1$  to produce a table  $T_1$  with  $D_1 = 2^{d_1}$  entries, where the  $i$ -th entry is  $E(X_1, i)$ . Using the property of the non-malleable extractor, one can show that  $(1 - \sqrt{\varepsilon_1})$ -fraction of the rows are uniform and almost  $t$ -wise independent (in the sense that any  $t$  good rows are close to uniform). The remaining rows are, however, arbitrarily correlated with those rows. Then, they

- Use the second source  $X_2$  to sample a sub-table  $T_2$  with some  $D_2$  rows of the table  $T_1$ , such that a fraction of at most  $\varepsilon_2$  of its rows are bad, and every  $t$  good rows are  $\sqrt{\varepsilon_1}$ -close to uniform, and,
- Apply a resilient function  $f: \{0, 1\}^{D_2} \rightarrow \{0, 1\}$  on the sub-table  $T_2$ .  $f$  has to be resilient against  $\sqrt{\varepsilon_2}D_2$  bad players, and should perform correctly even when the good players are  $t$ -wise independent.

It turns out that the sub-table  $T_2$  is  $D_2^t t \sqrt{\varepsilon_1}$ -close to a table where the good players are *truly*  $t$ -wise independent (as required by  $f$ ) and so it is enough to choose  $\varepsilon_1$  small enough so that  $D_2^t t \sqrt{\varepsilon_1}$  is small, and this proves the correctness of the construction.

While this beautiful approach does give an unbiased output bit, it seems that it is inherently bound to have running time polynomial in  $1/\varepsilon$ . This is because no matter which resilient function we use, even if there is just a single bad player among the  $D_2$  players, then that player alone may have  $1/D_2$  influence over the result (in fact, [25] showed there is a player with  $\Omega(\frac{\log D_2}{D_2})$  influence) and therefore that player can bias the result by  $1/D_2$ . Thus, the running time, which is at least  $D_2$ , is at least  $\Omega(\frac{1}{\varepsilon})$ , and this is indeed a common feature of all the constructions so far that use the CZ approach.

One could have hoped to sample a sub-table  $T_2$  that w.h.p. avoids *all* bad players, thus dispensing with the use of the resilient function. This approach is futile: If  $T_2$  avoids all bad players then every row  $y$  of it will do, so indeed  $E(X, y)$  is close to uniform and we can compute it fast, allowing for a small error. However, this brings us back to the seeded extractors case, and we already saw this cannot handle densities above half.

### 1.3 Our Main Result

The main result in the paper is a reduction showing how to explicitly construct low-error two-source extractors given explicit  $t$ -non-malleable extractors with small seed length dependence on  $t$ . Formally,

► **Theorem 5.** *Suppose for some constant  $\alpha > 0$  for every  $n_1, k_1, \varepsilon_1$  and  $t$  there exists an explicit function*

$$E: \{0, 1\}^{n_1} \times \{0, 1\}^d \rightarrow \{0, 1\}^m$$

that is a strong  $(k_1, \varepsilon_1)$   $t$ -non-malleable extractor with  $d \leq \alpha t \cdot \log(\frac{1}{\varepsilon_1})$ .  
Then, there exists an explicit function

$$F: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$$

that is a  $((n_1, k_1), (n_2 = O(\frac{d}{\alpha}), k_2 = O(\alpha n_2)), 2\sqrt{\varepsilon_1})$  two-source extractor, where the constants hidden in the big- $O$  notation are independent of  $\alpha$ .

We first remark that such non-malleable extractors non-explicitly exist. In fact, much better parameters are possible:

► **Theorem 6.** *Let  $n, k, t$  and  $\varepsilon$  be such that  $k \geq (t + 1)m + 2 \log \frac{1}{\varepsilon} + \log d + 4 \log t + 3$ . There exist a strong  $(k, \varepsilon)$   $t$ -n.m. extractor  $E: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  with  $d \leq 2 \log \frac{1}{\varepsilon} + \log(n - k) + 2 \log(t + 1) + 3$ .*

The proof of the Theorem is based on [21], where they only handle the  $t = 1$  case. The Theorem was also independently proved by Cohen and Shinkar [20]. For completeness we give the proof in Appendix A.

The currently best explicit construction of  $t$ -n.m. extractors is due to Li:

► **Theorem 7 ([28]).** *For any integer  $n, t$  and  $\varepsilon > 0$ , there exists an efficiently-computable function*

$$\text{nmEXT}: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}$$

that is a strong  $(k = d, t\varepsilon)$   $t$ -non-malleable extractor with seed length  $d = O(t^2(\log n + \log \frac{1}{\varepsilon} \cdot \log \log \frac{1}{\varepsilon}))$ .

So far the main focus in explicit constructions of  $t$ -non-malleable extractors has been getting an optimal seed length dependence on  $n$  and  $\varepsilon$ . Thus, Chattopadhyay et al. has  $d = \log^2(\frac{n}{\varepsilon})$  [8] and this has been improved in [15, 16, 9, 14] with the current best construction being Theorem 7 of [28] with  $d = O(\log n + \log \frac{1}{\varepsilon} \log \log \frac{1}{\varepsilon})$ . However, in all these constructions  $t$  is treated as a constant. In fact, Cohen [15, Lemma 2.5] proved that if one constructs a n.m. extractor for  $t = 1$  then an explicit construction for  $t$  follows at the cost of multiplying the seed by a  $t^2$  multiplicative factor.

There is a huge gap between the dependence of the seed length on  $t$  in the non-explicit construction of Theorem 6, where  $t$  contributes an *additive*  $2 \log t$  factor to the seed length, and the explicit Theorem 7 where  $t$  contributes a *multiplicative*  $t^2$  factor to the seed length.<sup>7</sup> Correspondingly, the quality of the two source construction we give significantly improves with a better dependence of the seed on the parameter  $t$ . In Table 1 we list the two-source extractors constructions we get for:

- The current best explicit constructions (we get nothing),
- A quadratic improvement over currently best explicit (we improve upon Raz's extractor), and,
- A further polynomial improvement.

The parameters in the second row (and Theorem 5) resemble those of Raz's extractor: one source is long with very low entropy, the other is short with constant entropy rate. The

<sup>7</sup> It is worth mentioning that an early construction of Cohen, Raz and Segev [19], although not explicitly stating it, does get a very good dependence of  $d$  on  $t$  with  $d = O(\log \frac{n}{\varepsilon} + t)$ . However, their construction only works for high min entropy and so does not imply a two-source extractor for densities below half.

■ **Table 1** Bounds for  $((n, k), (n_2, k_2), \varepsilon)$  two-source extractors assuming an explicit  $t$  n.m. extractor with various seed length  $d$  dependence on  $t$ . In all cases, the error  $\varepsilon$  is low.

Dependence on $t$	$k$	$n_2$	$k_2$	
$\omega\left(t \log \frac{1}{\varepsilon}\right)$				The approach fails
$\alpha t \log\left(\frac{1}{\varepsilon}\right)$	arbitrary	$O\left(\frac{d}{\alpha}\right)$	$O(\alpha)n_2$	$\alpha$ is any constant
$t^\alpha \log\left(\frac{1}{\varepsilon}\right)$ or better	arbitrary	$\text{poly}_{\alpha, \beta}(d)$	$n_2^\beta$	For some constants $\alpha, \beta < 1$
$t^\alpha \log\left(\frac{1}{\varepsilon}\right)$ or better	small enough	$n$	$n^\beta$	For some constant $\beta < 1$

main difference is that in Raz's extractor the entropy rate has to be above half, whereas here, assuming the existence of the appropriate explicit non-malleable extractors, the entropy rate can be an arbitrarily small constant.

By allowing the seed-length of the n.m. extractor to have an even better dependence on  $t$  (and non-explicitly it does), we succeed in supporting polynomially-small min-entropies. More specifically, if the seed length dependence on  $t$  is  $t^\alpha \log\left(\frac{1}{\varepsilon}\right)$  for a small enough constant  $\alpha$ , then we can support min-entropy of  $k_2 = n_2^\beta$  where  $\beta = \beta(\alpha)$  is another constant.

Also, in that regime of dependence, we can set the error  $\varepsilon$  to be small enough so that  $n_2 = n$ , in which case we get a *balanced* two-source extractor supporting some polynomially-small min-entropy (see Corollary 19).

We believe this clearly demonstrates that the dependence of the seed length on  $t$  in non-malleable extractors is directly related to the required density of the seed (i.e., second source) in low-error, two-source constructions. We believe this understanding is an important, qualitative understanding. We believe our work is the first to draw attention to this important question and we hope it will facilitate further research on achieving the correct dependence of the seed on the non-malleability parameter  $t$ .

## 1.4 Our Technique

In the CZ construction we have the following ingredients:

1. The use of the first source to construct a table with many good rows (every row in the table corresponds to applying an extractor on the first source, with some fixed seed).
2. The use of  $t$ -non-malleable extractors to get *local*  $t$ -wise independence, where every  $t$  good rows are close to uniform.
3. The use of the second source to sample a sub-table of the table constructed from the first source.
4. The realization that with the right choice of parameters the sub-table is *globally* close to a table where the good rows are perfectly  $t$ -wise.
5. The use of resilient functions.

In our solution we keep (1)-(3) and completely dispense with (4) and (5), i.e., we do not use resilient functions and we do not try to achieve a sub-table that is *globally* close to a *truly*  $t$ -wise independent distribution. Instead, we work with the much weaker *local* guarantee that every  $t$  good rows are close to uniform.

Thus, our construction is as follows. We are given two samples from independent sources  $x_1 \sim X_1$  and  $x_2 \sim X_2$ . Then:

1. We use a  $t$ -non-malleable extractor  $E$  with error  $\varepsilon_1$  and seed length  $d_1$  to construct a table with  $D_1 = 2^{d_1}$  entries, where the  $i$ -th entry is  $E(X_1, i)$ . Using the property of



non-malleable extractors one can show that  $(1 - \sqrt{\varepsilon_1})$ -fraction of the rows are good in the sense that a good row is close to uniform even conditioned on  $t - 1$  other rows. The remaining rows are arbitrarily correlated with the good ones. So far, everything is identical to the [10] construction.

2. We use the second sample  $x_2$  to sample  $t$  rows from that table, with the property that with high probability (over the choice of  $x_2 \sim X_2$ ) **at least one of the  $t$  samples is a good row** (in the table with  $D_1$  rows).

We note that this is very different from the [10] construction, where the requirement is that with high probability (over the choice of  $x_2 \sim X_2$ ) the fraction of bad rows in the sub-table is about the same as the fraction of bad rows in the original table.

3. We then take the *parity* of the  $t$  strings written in the  $t$  rows we sampled.

This is again very different from the [10] construction, where a resilient function is applied on the sub-table (and notice that the parity function is not resilient at all).

Conceptually, what happened is that we take a *dramatically smaller* sample set than before. Specifically, in [10, 6] the sample set is much larger than  $t$ , whereas in our algorithm the sample size is  $t$ . Accordingly, we replace the requirement that the fraction of bad players in the sample set is small, with the weaker requirement that *not all* of the players in the sample set are bad. If the sample size is  $t$  and not all the players in the sample are bad, then every good player (and even if there is just a single good player) is almost independent of the other  $t - 1$  players, and therefore we can just apply the parity function on the  $t$  bits in the sample. Thus, we can also dispense with the resilient function  $f$  and just use the parity function instead.

Notice that by doing so we also get rid of the annoying (and expensive) requirement that  $D_2^t \varepsilon_1 < 1$ , because we no longer need to convert a table where every  $t$  rows are locally close to uniform, to a table that is globally close to being perfectly  $t$ -wise independent.

There is still a fundamental question we need to answer. Inspecting the argument, we see that there is a circular dependency in the construction: The sample size of the sampler determines the required  $t$ -non-malleability of the extractor, which then affects the parameters of the extractor, and in particular the number of bad rows, which, in turn, affects the required degree of the sampler. It is therefore, offhand, not clear whether such a construction is possible at all even assuming the best possible non-malleable extractors.

The above inquiry raises the question of what is the dependence of the seed length of non-malleable extractors on the non-malleability parameter  $t$ . This question was considered before by several people. In particular, Cohen and Shinkar [20] independently investigated this. As we explained before, it turns out that in non-explicit constructions the dependence is very mild, and such an approach can be easily supported.

In the paper we analyze what is the threshold beyond which such an approach cannot work. Roughly speaking, non-malleable extractors with seed length below  $t \log(\frac{n}{\varepsilon})$  work well, while non-malleable extractors with seed length above it do not. In Section 3 we demonstrate how the dependence of the seed length  $d$  on  $t$  affects the parameters of the two-source extractor construction.

Finally, we are left with two questions regarding *explicitness*:

- We ask whether the sampler can be made explicit, i.e., whether we can find a sampler with such a small sample size that except for very few  $x_2$ -s always sees at least one good row. This question readily translates to the existence (or the explicit existence) of *dispersers* that are good against small tests. Remarkably, Zuckerman [35] gave a beautiful explicit construction with nearly optimal bounds, and we show the dispersers he constructed work well for us.



- Current explicit constructions of non-malleable extractors [13, 8, 15, 14, 9, 18, 28] for small entropies are above that threshold. This is mainly due to the use of alternating extraction techniques which treat the seed and the source symmetrically. Thus, this paper raises the challenge of explicitly constructing non-malleable extractors with better seed length dependence on  $t$ .

We believe identifying the connection between the seed length dependence on  $t$  and low error, two-source extractors is important on its own, and is a major contribution of the paper. We hope this work would lead to further developments in explicit constructions of both non-malleable and two-source extractors.

## 1.5 Related work

Li [26] showed how to build a  $((n, 0.499n), (n, k), 2^{-\Omega(n)})$  two-source extractor assuming a 1-non-malleable extractor with seed-length  $d = 2 \log(1/\varepsilon) + o(n)$ . Li's work is orthogonal to ours. First, it asks for small seed dependence on the error: the seed-length of the non-malleable extractor has to be at most 2.001, while we look on the dependence on  $t$ . Also, it achieves limited parameters (even assuming non-explicit constructions) that are close to those in Bourgain's construction, and it is also close in spirit to Bourgain's construction.

As we said before, we believe our work reveals an intrinsic connection between the dependence of the seed length of a non-malleable extractor on the non-malleability parameter  $t$  and the quality of low-error two-source extractors, and is the first work to draw attention to the important problem of the dependence of the seed length on  $t$  in explicit construction. We hope, and believe, this approach may lead to getting better explicit, low-error, two source extractors, which is a fundamental problem and a long standing barrier in TCS.

## 2 Preliminaries

Throughout the paper we have the convention that lowercase variables are the logarithm (in base-2) of their corresponding uppercase variables, e.g.,  $n = \log N$ ,  $d = \log D$ , etc. The density of a set  $B \subseteq [D]$  is  $\rho(B) = \frac{|B|}{D}$ .

### 2.1 Random Variables, Min-Entropy

The *statistical distance* between two distributions  $X$  and  $Y$  on the same domain  $D$  is defined as  $|X - Y| = \max_{A \subseteq D} (\Pr[X \in A] - \Pr[Y \in A])$ . If  $|X - Y| \leq \varepsilon$  we say that  $X$  is  $\varepsilon$ -close to  $Y$  and denote it by  $X \approx_\varepsilon Y$ . We will denote by  $U_n$  a random variable distributed uniformly over  $\{0, 1\}^n$  and which is independent of all other variables. We also say that a random variable is *flat* if it is uniform over its support.

For a function  $f: D_1 \rightarrow D_2$  and a random variable  $X$  distributed over  $D_1$ ,  $f(X)$  is the random variable, distributed over  $D_2$ , which is obtained by choosing  $x$  according to  $X$  and computing  $f(x)$ . For a set  $A \subseteq D_1$ , we simply denote  $f(A) = \{f(x) \mid x \in A\}$ . It is well-known that for every  $f: D_1 \rightarrow D_2$  and two random variables  $X$  and  $Y$ , distributed over  $D_1$ , it holds that  $|f(X) - f(Y)| \leq |X - Y|$ .

The *min-entropy* of a random variable  $X$  is defined by

$$H_\infty(X) = \min_{x \in \text{Supp}(X)} \log \frac{1}{\Pr[X = x]}.$$

A random variable  $X$  distributed over  $\{0,1\}^n$  with min-entropy at least  $k$  is called an  $(n,k)$ -source. Every distribution  $X$  with  $H_\infty(X) \geq k$  can be expressed as a convex combination of flat distributions, each with min-entropy at least  $k$ .

## 2.2 Extractors

► **Definition 8.** A function  $2\text{Ext}: \{0,1\}^{n_1} \times \{0,1\}^{n_2} \rightarrow \{0,1\}^m$  is an  $((n_1, k_1), (n_2, k_2), \varepsilon)$  two-source extractor if for every two independent sources  $X_1$  and  $X_2$  where  $X_1$  is an  $(n_1, k_1)$  source and  $X_2$  is an  $(n_2, k_2)$  source, it holds that  $2\text{Ext}(X_1, X_2) \approx_\varepsilon U_m$ .

► **Definition 9.**  $E: \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$  is a strong  $(k, \varepsilon)$   $t$ -non-malleable ( $n.m.$ ) extractor, if for every  $(n, k)$  source  $X$  and every functions  $f_1, \dots, f_t: [D] \rightarrow [D]$  with no fixed-points it holds that,

$$\left| (Y, E(X, Y), \{E(X, f_i(Y))\}_{i=1}^t) - (Y, U_m, \{E(X, f_i(Y))\}_{i=1}^t) \right| \leq \varepsilon,$$

where  $Y$  is uniformly distributed over  $\{0,1\}^d$  and is independent of  $X$ .

A simple consequence, proved in [10], is:

► **Lemma 10** ([10], Lemma 3.4). *Let  $E: \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$  be a strong  $(k, \varepsilon)$   $t$ -non-malleable extractor. Let  $X$  be any  $(n, k)$  source. Then there exists a set  $BAD \subseteq [N]$  with  $\rho(BAD) \leq \sqrt{\varepsilon}$  such that for every  $y \notin BAD$ , and every  $y'_1, \dots, y'_t \in [D] \setminus y$ ,*

$$\left| (E(X, y), \{E(X, y'_i)\}_{i \in [t]}) - (U_m, \{E(X, y'_i)\}_{i \in [t]}) \right| \leq \sqrt{\varepsilon}.$$

## 2.3 Dispersers

► **Definition 11.** A function  $\Gamma: [N] \times [D] \rightarrow [M]$  is a  $(K, K')$  disperser if for every  $A \subseteq [N]$  with  $|A| \geq K$  it holds that  $\left| \bigcup_{i \in [D]} \Gamma(A, i) \right| \geq K'$ .

Zuckerman showed the following remarkable explicit construction:

► **Theorem 12** ([35], Theorem 1.9). *There exists a constant  $c_{disp}$  such that the following holds. For every constants  $0 < a, b < 1$ , every  $N, K = N^a, M \leq K^{1-b}$  and  $K' < M$  there exists an efficient family of  $(K, K')$  dispersers*

$$\Gamma: [N] \times [D] \rightarrow [M]$$

with degree  $D = c_{disp} \cdot \frac{\log \frac{N}{K}}{\log \frac{M}{K'}}$ .

The parameters in Theorem 12 are tight up to a constant factor:

► **Theorem 13** ([33], Theorem 1.5). *There exists a constant  $c_0$  such that the following holds. Let  $\Gamma: [N] \times [D] \rightarrow [M]$  be a  $(K, K')$  disperser where  $K < N$  and  $K' < M/2$ . Then,*

$$D \geq c_0 \cdot \frac{\log \frac{N}{K}}{\log \frac{M}{K'}}.$$

# 3 The Construction

## 3.1 The Overall Structure

Given:

$$E: \{0,1\}^{n_1} \times [D] \rightarrow \{0,1\}^m$$

$$\Gamma: \{0,1\}^{n_2} \times [t+1] \rightarrow [D]$$

We define  $2\text{Ext}: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$  by

$$2\text{Ext}(x_1, x_2) = \bigoplus_{y: \exists i \text{ s.t. } \Gamma(x_2, i) = y} E(x_1, y).$$

► **Theorem 14.** *Assume  $E$  is a strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor and  $\Gamma$  is a  $(B_2, \sqrt{\varepsilon_1}D)$  disperser. Then, for every  $k_2$ ,  $2\text{Ext}$  is a  $\left((n_1, k_1), (n_2, k_2), \frac{B_2}{K_2} + \sqrt{\varepsilon_1}\right)$  two-source extractor.*

**Proof.** Let  $X_1$  be an  $(n_1, k_1)$  source and  $X_2$  an  $(n_2, k_2)$  source. W.l.o.g.  $X_1$  and  $X_2$  are flat. As  $E$  is  $t$ -n.m., by Lemma 10 there exists a set  $BAD_1 \subseteq [D]$  with  $\rho(BAD_1) \leq \sqrt{\varepsilon_1}$  such that for every  $y \notin BAD_1$  and every  $y'_1, \dots, y'_t \in [D] \setminus \{y\}$ ,

$$\left| \left( E(X, y), \{E(X, y'_i)\}_{i \in [t]} \right) - \left( U_m, \{E(X, y'_i)\}_{i \in [t]} \right) \right| \leq \sqrt{\varepsilon_1}.$$

Let  $BAD_2 \subseteq [N_2]$  be

$$BAD_2 = \{x_2 \in \{0, 1\}^{n_2} : \Gamma(x_2) \subseteq BAD_1\}.$$

Thus,  $\Gamma(BAD_2) \subseteq BAD_1$ . Since  $|BAD_1| \leq \sqrt{\varepsilon_1}D$  and  $\Gamma_2$  is a  $(B_2, \sqrt{\varepsilon_1}D)$  disperser, it follows that  $|BAD_2| \leq B_2$ . However, for any  $x_2 \in \{0, 1\}^{n_2} \setminus BAD_2$ , there exists an  $i \in [t+1]$  such that  $y = \Gamma(x_2, i) \notin BAD_1$ . Hence,

$$\left| \left( E(X, y), \{E(X, y_j)\}_{y_j \in \Gamma(x_2) \setminus \{y\}} \right) - \left( U_m, \{E(X, y_j)\}_{y_j \in \Gamma(x_2) \setminus \{y\}} \right) \right| \leq \sqrt{\varepsilon_1}.$$

Thus,

$$\left| \bigoplus_{y: \exists i \text{ s.t. } \Gamma(x_2, i) = y} E(x_1, y) - U_m \right| \leq \sqrt{\varepsilon_1}.$$

Altogether, the error is at most  $\frac{|BAD_2|}{K_2} + \sqrt{\varepsilon_1}$  and the proof is complete. ◀

## 3.2 The Activation Threshold

In the previous subsection we assumed the existence of a  $(B_2, \sqrt{\varepsilon_1}D)$  disperser  $\Gamma$  and a  $t$ -n.m. extractor  $E$ . However,

- The degree  $D_2$  of the disperser  $\Gamma$  affects the non-malleability parameter  $t$  of the extractor, because the argument requires  $t \geq D_2 - 1$ ,
- The non-malleability parameter  $t$  affects the degree  $2^d = D$  of the extractor, because intuitively, the greater  $t$  is the greater the degree has to be,
- The degree  $D$  determines  $|BAD_1| = \sqrt{\varepsilon_1}D$ , and,
- The size  $B_1$  of the set  $BAD_1$  determines the degree of the disperser  $\Gamma$  as  $D_2 = O\left(\frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}}\right)$ , and up to a multiplicative factor this is also a lower bound on  $D_2$ .

Thus we have a circular dependence and it is not clear at all that such a construction is even possible. Indeed, as we shall see, if the seed length of  $E$  is larger than  $t \log(\frac{1}{\varepsilon_1})$  such a construction is impossible. However, at least non-explicitly, better non-malleable extractors exist that comfortably suffice for the construction. Our goal in this section is to determine which dependence of the seed length on  $t$  and  $\varepsilon_1$  suffices for the construction.

### 3.3 The analysis fails when $d \geq ct \log(\frac{1}{\varepsilon})$ for some constant $c$

► **Lemma 15.** *Suppose*

$$E: \{0, 1\}^{n_1} \times [D] \rightarrow \{0, 1\}^m$$

$$\Gamma: \{0, 1\}^{n_2} \times [t+1] \rightarrow [D]$$

are such that  $E$  is a strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor and  $\Gamma$  is any  $(B_2, B_1 = \sqrt{\varepsilon_1}D)$  disperser, as required by Theorem 14. Suppose Theorem 14 gives that

$$2\text{Ext}: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$$

is an  $((n_1, k_1), (n_2, k_2), 2\sqrt{\varepsilon_1})$  two-source extractor with  $K_2 < \sqrt{N_2}$ . Then,  $\log_{1/\varepsilon_1} D \leq \frac{t+1}{c_0}$ , where  $c_0$  is the constant guaranteed by Theorem 13.

**Proof.** We first give some easy bounds on the parameters:

- $B_2 \leq K_2$ , for otherwise Theorem 12 constructs 2Ext with the trivial error 1.
- Also,  $tB_2 \geq B_1$ , for otherwise we can take a set  $A \subseteq \{0, 1\}^{n_2}$  of cardinality  $B_2$  and the size of its neighbor set is at most  $B_2 t < B_1$  violating the disperser property.
- Finally,  $\frac{B_1}{t} \geq \sqrt{B_1}$  because otherwise  $\sqrt{B_1} < t$  and then

$$D_1 = \frac{B_1}{\sqrt{\varepsilon_1}} < \frac{t^2}{\sqrt{\varepsilon_1}} \leq \frac{n_1^2}{\sqrt{\varepsilon_1}} \leq \frac{1}{\varepsilon_1^2},$$

where the last inequality follows from the assumption on  $\varepsilon_1$ . This contradicts the lower-bound for extractors [33].

Together,  $\frac{N_2}{B_2} \geq \frac{N_2}{K_2} \geq K_2 \geq B_2 \geq \frac{B_1}{t} \geq \sqrt{B_1} = \sqrt{\varepsilon_1}D$  and  $\frac{D}{B_1} = \frac{1}{\sqrt{\varepsilon_1}}$ . Now,  $\Gamma: \{0, 1\}^{n_2} \times [t+1] \rightarrow [D]$  is a  $(B_2, B_1 = \sqrt{\varepsilon_1}D)$  disperser and therefore by Theorem 13 it has degree at least  $c_0 \cdot \frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}}$  for some constant  $c_0$ . Therefore,

$$\begin{aligned} t+1 &\geq c_0 \cdot \frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}} \geq c_0 \cdot \frac{\log \sqrt{\varepsilon_1}D}{\log \frac{1}{\sqrt{\varepsilon_1}}} \\ &= 2c_0 \cdot \log_{1/\varepsilon_1}(\sqrt{\varepsilon_1}D) = 2c_0 \cdot (\log_{1/\varepsilon_1} D - 1/2) \geq c_0 \log_{1/\varepsilon_1} D. \quad \blacktriangleleft \end{aligned}$$

The analysis in the above proof is quite tight and in the next subsection we prove the converse (which also entails Theorem 5).

### 3.4 When $d = O(t \log(\frac{1}{\varepsilon}))$

► **Lemma 16.** *Let  $\varepsilon_1 \leq \frac{1}{n}$ . Suppose there exists an explicit*

$$E: \{0, 1\}^{n_1} \times [D_1] \rightarrow \{0, 1\}^m$$

that is a strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor with  $\log_{1/\varepsilon_1} D_1 \leq \frac{\alpha}{8c_{disp}} t$  for some constant  $\alpha > 0$ , some constant  $t$  and some  $k_1$ . Then there exists an explicit

$$2\text{Ext}: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$$

that is a  $((n_1, k_1), (n_2 = \frac{4}{\alpha}d_1, k_2 = \alpha n_2), 2\sqrt{\varepsilon_1})$  two-source extractor.

**Proof.** Fix  $t$  as in the hypothesis of the lemma. Set  $D$  such that  $\log_{1/\varepsilon_1} D = \frac{\alpha t}{8c_{disp}}$ . Let

$$\Gamma: [N_2 = D^{4/\alpha}] \times [D_2] \rightarrow [D]$$

be the  $(B_2 = D^2, B_1 = \sqrt{\varepsilon_1}D)$  disperser promised to us by Theorem 12 for  $a = \frac{\alpha}{2}$  (because  $B_2 = N_2^a$ ) and  $b = \frac{1}{2}$  (because  $D = B_2^b$ ). By Theorem 12 the degree  $D_2$  of  $\Gamma$  is

$$\begin{aligned} D_2 &= c_{disp} \cdot \frac{\log \frac{N_2}{B_2}}{\log \frac{D}{B_1}} = c_{disp} \cdot \frac{\frac{4(1-a)}{\alpha} \log D}{\log \frac{1}{\sqrt{\varepsilon_1}}} \\ &= c_{disp} \cdot \left( \frac{1}{\alpha} - \frac{1}{2} \right) \frac{8 \log D}{\log 1/\varepsilon_1} = c_{disp} \cdot \left( \frac{8}{\alpha} - 4 \right) \log_{1/\varepsilon_1} D \\ &= c_{disp} \cdot \left( \frac{8}{\alpha} - 4 \right) \frac{\alpha t}{8c_{disp}} = \left( 1 - \frac{\alpha}{2} \right) t < t. \end{aligned}$$

Let

$$E: \{0, 1\}^{n_1} \times [D_1] \rightarrow \{0, 1\}^m$$

be the explicit, strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor with  $\log_{1/\varepsilon_1} D_1 \leq \frac{\alpha}{8c_{disp}} t = \log_{1/\varepsilon_1} D$  promised by the hypothesis of the lemma. As  $\frac{1}{\varepsilon} > 1$ , we see that  $D_1 \leq D$  and we may take  $D_1$  larger so that it equals  $D$ .

Now let

$$2\text{Ext}: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$$

be constructed from  $E$  and  $\Gamma$  as above. As  $E$  is a strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor and  $\Gamma$  is a  $(B_2, \sqrt{\varepsilon_1}D_1)$  disperser, Theorem 14 tells us that for every  $k_2$ ,  $2\text{Ext}$  is a  $((n_1, k_1), (n_2, k_2), \frac{B_2}{K_2} + \sqrt{\varepsilon_1})$  two-source extractor. Taking  $k_2 = \alpha n_2$ ,

$$\frac{B_2}{K_2} + \sqrt{\varepsilon_1} = \frac{D_1^2}{D_1^4} + \sqrt{\varepsilon_1} = \frac{1}{D_1^2} + \sqrt{\varepsilon_1}.$$

But  $D_1 \geq \frac{1}{\varepsilon_1}$  (this is true for any seeded extractor [33]). Altogether the error is at most  $\sqrt{\varepsilon_1} + \frac{1}{\varepsilon_1^2} \leq 2\sqrt{\varepsilon_1}$ .  $\blacktriangleleft$

### 3.5 When $d = O(t^\alpha \log(\frac{1}{\varepsilon}))$

A careful examination of the parameters shows that if the dependence of  $d_1$  on  $t$  is better, our scheme yields a two-source extractor that supports even smaller min-entropies. Roughly speaking, if  $\log_{1/\varepsilon_1} D_1 = t^\alpha$  for some  $\alpha < 1$  we can support some polynomially-small min-entropy  $k_2 = n_2^\beta$ , instead of only supporting min-entropies of constant rate. Specifically:

► **Lemma 17.** *Let  $\varepsilon_1 \leq \frac{1}{n}$ . There exists a constant  $\beta_0 < 1$  such that for every  $\beta_0 < \beta < 1$  there exist constants  $\alpha < 1$  and  $\gamma > 1$  so that the following holds. Suppose there exists an explicit*

$$E: \{0, 1\}^{n_1} \times [D_1] \rightarrow \{0, 1\}^m$$

*that is a strong  $(k, \varepsilon_1)$   $t$ -n.m. extractor with  $\log_{1/\varepsilon_1} D_1 \leq t^\alpha$  for some  $k_1$ , and  $t$  which is a large enough polynomial in  $\log \frac{1}{\varepsilon_1}$ . Then there exists an explicit*

$$2\text{Ext}: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$$

*that is a  $((n_1, k_1), (n_2 = d_1^\gamma, k_2 = n_2^\beta), 2\sqrt{\varepsilon_1})$  two-source extractor.*

The proof is similar to the proof of Lemma 16. However, it is no longer true that  $K_2$  is a *constant* power of  $N_2$ , so we should be more careful with the parameters of Zuckerman's disperser. Particularly, in this regime of parameters, the degree  $D_2$  (and consequently  $t$ ) is no longer constant but will be poly-logarithmic in  $\frac{1}{\varepsilon}$ . The following Theorem extends Theorem 12 for the more general case.

► **Theorem 18** ([35], Theorem 1.9). *There exist constants  $c_1, c_2 > 1$  such that the following holds. For every  $0 < \delta < 1$ ,  $N, K = N^\delta$ ,  $M \leq N^{\delta^{c_2}}$  and  $K' < M$  there exists an efficient family of  $(K, K')$  dispersers*

$$\Gamma: [N] \times [D] \rightarrow [M]$$

$$\text{with degree } D = \left(\frac{1}{\delta}\right)^{c_1} \cdot \frac{n}{\log \frac{M}{K'}}.$$

We are now ready to prove Lemma 17.

**Proof of Lemma 17.** Let  $c_1$  and  $c_2$  be as in Theorem 18. Set  $\beta_0 = 1 - \frac{1}{c_2}$  and fix some  $\beta_0 < \beta < 1$ . Fix  $t$  as in the hypothesis of the lemma. Set  $D$  such that  $\log_{1/\varepsilon_1} D = t^\alpha$  for  $\alpha = \alpha(\beta)$  we will soon explicitly determine. Let

$$\Gamma: [N_2 = D^{1/\delta^{c_2}}] \times [D_2] \rightarrow [D]$$

be the  $(B_2 = N_2^\delta, B_1 = \sqrt{\varepsilon_1} D)$  disperser promised to us by Theorem 18, for  $\delta = \frac{1}{2} n_2^{-(1-\beta)}$ . Notice that  $b_2 = \delta n_2 = \frac{1}{2} n_2^\beta$  and set  $k_2 = 2b_2 = n_2^\beta$ . Also, observe that  $n_2 = \frac{1}{\delta^{c_2}} d = (2^{c_2} d)^{\gamma'}$  for

$$\gamma' = \frac{1}{1 - c_2(1 - \beta)}.$$

As  $\beta > \beta_0$  we see that  $\gamma' > 1$ . It follows that  $n_2 = d^{\gamma'}$  for some  $\gamma' < \gamma < 2\gamma'$ .

By Theorem 18, the degree  $D_2$  of  $\Gamma$  is

$$\begin{aligned} D_2 &= \left(\frac{1}{\delta}\right)^{c_1} \cdot \frac{n_2}{\log \frac{D}{B_1}} = \left(\frac{1}{\delta}\right)^{c_1} \cdot \frac{2n_2}{\log(1/\varepsilon_1)} \\ &= \left(2n_2^{1-\beta}\right)^{c_1} \cdot \frac{2 \cdot n_2}{\log(1/\varepsilon_1)} = 2^{c_1+1} \cdot \frac{n_2^{1+c_1(1-\beta)}}{\log(1/\varepsilon_1)} = 2^{c_1+1} \cdot \frac{(\log D)^{\gamma(1+c_1(1-\beta))}}{\log(1/\varepsilon_1)}. \end{aligned}$$

Set  $\xi = \gamma(1 + c_1(1 - \beta)) > 1$  and  $\alpha = \frac{1}{2\xi}$  (note that  $\alpha$  is in fact a function of  $\beta$ ). We get that:

$$D_2 = 2^{c_1+1} \frac{\log^\xi D}{\log(1/\varepsilon_1)} = 2^{c_1+1} \left(\log^{\xi-1} \frac{1}{\varepsilon_1}\right) \left(\log_{1/\varepsilon_1} D\right)^\xi = 2^{c_1+1} \left(\log^{\xi-1} \frac{1}{\varepsilon_1}\right) t^{\alpha\xi}.$$

Now, note that  $t^{\alpha\xi} = \sqrt{t}$ , so  $D_2 < t$  as long as  $t > 4^{c_1+1} \log^{2(\xi-1)} \frac{1}{\varepsilon_1}$ .

Let

$$D: \{0, 1\}^{n_1} \times [D_1] \rightarrow \{0, 1\}^m$$

be the explicit, strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor with  $\log_{1/\varepsilon_1} D_1 \leq t^\alpha = \log_{1/\varepsilon_1} D$  promised by the hypothesis of the lemma. Again, we can take  $D_1 = D$ .

Now let

$$2\text{Ext}: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$$

be constructed from  $E$  and  $\Gamma$  as in Section 3.1. We have that  $E$  is a strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor and  $\Gamma$  is a  $(B_2, \sqrt{\varepsilon_1}D_1)$  disperser, so by Theorem 14 2Ext is a  $((n_1, k_1), (n_2, k_2), \frac{B_2}{K_2} + \sqrt{\varepsilon_1})$  two-source extractor.

In our case,  $\frac{B_2}{K_2} = 2^{b_2 - k_2} = 2^{-b_2}$ . We stress that  $b_2 \geq \frac{1}{2} \log \frac{1}{\varepsilon_1}$ . To see this, note that  $2b_2 = n_2^\beta = d_1^{\beta\gamma}$ . As  $\beta\gamma \geq \beta\gamma' = \frac{\beta}{1 - c_2(1 - \beta)} \geq 1$ , and  $d_1 \geq 2 \log \frac{1}{\varepsilon_1}$  (again, this is true for any seeded extractor), we finally have that  $2b_2 \geq d_1 > \log \frac{1}{\varepsilon_1}$ . Overall,

$$\frac{B_2}{K_2} + \sqrt{\varepsilon_1} \leq 2\sqrt{\varepsilon_1}$$

and we are done. ◀

Next, we show that we can *balance* the above two-source extractor (i.e.,  $n_1 = n_2$ ) by choosing the error  $\varepsilon_1$  appropriately and assuming  $k_1$  is small enough. The resulting two-source extractor supports polynomially-small min-entropies from both sources. Formally:

► **Corollary 19.** *Let  $\varepsilon_1 \leq \frac{1}{n}$ . There exists a constant  $\beta_0 < 1$  such that for every  $\beta_0 < \beta < 1$  there exists a constant  $\alpha < 1$  so that the following holds. Suppose there exists an explicit*

$$E: \{0, 1\}^{n_1} \times [D_1] \rightarrow \{0, 1\}^m$$

*that is a strong  $(k_1, \varepsilon_1)$   $t$ -n.m. extractor with  $\log_{1/\varepsilon_1} D_1 \leq t^\alpha$  for some  $k_1 \leq d_1$ , and  $t$  which is a large enough polynomial in  $\log \frac{1}{\varepsilon_1}$ . Then there exists an explicit*

$$2\text{Ext}: \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^m$$

*that is an  $((n, k = k_1), (n, k), \varepsilon)$  two-source extractor for  $k = n^\beta$  and  $\varepsilon = 2^{-n^{\Omega(1)}}$ .*

**Proof.** Following the notations of Lemma 17, let  $\beta_0, \alpha, \gamma$  be the constants set according to  $\beta$ . Let  $2\text{Ext}: \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$  be the explicit  $((n_1, k_1), (n_2 = d_1^\gamma, k_2 = n_2^\beta), 2\sqrt{\varepsilon_1})$  that is guaranteed to us.

We require  $n = n_1 = n_2 = d_1^\gamma$ , so as  $d_1 = t^\alpha \log \frac{1}{\varepsilon_1} \leq t \log \frac{1}{\varepsilon_1}$  and  $t$  is polynomial in  $\log \frac{1}{\varepsilon_1}$ , denote  $t \log \frac{1}{\varepsilon_1} = \log^{\eta'} \frac{1}{\varepsilon_1}$  and  $n = \log^\eta \frac{1}{\varepsilon_1}$  for some large enough constants  $\eta', \eta = \gamma\eta'$ . This guarantees that  $\varepsilon = 2\sqrt{\varepsilon_1} = 2^{-n^{\Omega(1)}}$ .

Next, note that  $k_1 \leq d_1$  and  $d_1 = n^{\frac{1}{\gamma}}$ . Indeed,  $n^{\frac{1}{\gamma}} \leq n^\beta$  since we already observed in the proof of Lemma 17 that  $\gamma\beta \geq 1$ . Overall  $k_1 \leq n^\beta$  for every  $\beta > \beta_0$ . As by construction  $k_2 = n_2^\beta$  for every  $\beta > \beta_0$  as well, the proof is concluded. ◀

---

## References

- 1 H.L. Abbott. Lower bounds for some Ramsey numbers. *Discrete Mathematics*, 2(4):289–293, 1972.
- 2 Noga Alon. The Shannon capacity of a union. *Combinatorica*, 18(3):301–310, 1998.
- 3 Boaz Barak. A simple explicit construction of an  $n^{\tilde{O}(\log n)}$ -Ramsey graph. *arXiv preprint math/0601651*, 2006.
- 4 Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. *Journal of the ACM (JACM)*, 57(4):20, 2010.
- 5 Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for  $n^{o(1)}$  entropy, and Ramsey graphs beating the frankl-wilson construction. *Annals of Mathematics*, 176(3):1483–1544, 2012.

- 6 Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. An efficient reduction from two-source to non-malleable extractors: achieving near-logarithmic min-entropy. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1185–1194. ACM, 2017.
- 7 Jean Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1(01):1–32, 2005.
- 8 Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 285–298. ACM, 2016.
- 9 Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 158–167. IEEE, 2016.
- 10 Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 670–683. ACM, 2016.
- 11 Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.
- 12 Fan R.K. Chung. A note on constructive methods for Ramsey numbers. *Journal of Graph Theory*, 5(1):109–113, 1981.
- 13 Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. *SIAM Journal on Computing*, 45(4):1297–1338, 2016.
- 14 Gil Cohen. Making the most of advice: New correlation breakers and their applications. In *Proceedings of 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 188–196. IEEE, 2016.
- 15 Gil Cohen. Non-malleable extractors – new tools and improved constructions. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- 16 Gil Cohen. Non-malleable extractors with logarithmic seeds. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 23, page 30, 2016.
- 17 Gil Cohen. Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, pages 278–284. ACM, 2016.
- 18 Gil Cohen. Two-source extractors for quasi-logarithmic min-entropy and improved privacy amplification protocols. In *ECCC*, 2016.
- 19 Gil Cohen, Ran Raz, and Gil Segev. Nonmalleable extractors with short seeds and applications to privacy amplification. *SIAM Journal on Computing*, 43(2):450–476, 2014.
- 20 Gil Cohen and Igor Shinkar. Personal communication, 2017.
- 21 Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 601–610. ACM, 2009.
- 22 Peter Frankl. A constructive lower bound for ramsey numbers. *Ars Combinatoria*, 3(297-302):28, 1977.
- 23 Peter Frankl and Richard M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, 1981.
- 24 Vince Grolmusz. Low rank co-diagonal matrices and Ramsey graphs. *Journal of combinatorics*, 7(1):R15–R15, 2001.
- 25 Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 68–80. IEEE, 1988.



- 26 Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 688–697. IEEE, 2012.
- 27 Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 168–177. IEEE, 2016.
- 28 Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. 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 1144–1156. ACM, 2017. doi: 10.1145/3055399.3055486.
- 29 Raghu Meka. Explicit resilient functions matching ajtai-linial. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1132–1148. SIAM, 2017.
- 30 Zs Nagy. A constructive estimation of the ramsey numbers. *Mat. Lapok*, 23:301–302, 1975.
- 31 Moni Naor. Constructing Ramsey graphs from small probability spaces. *IBM Research Report RJ*, 8810, 1992.
- 32 Victor Neumann-Lara. The dichromatic number of a digraph. *Journal of Combinatorial Theory, Series B*, 33(3):265–270, 1982.
- 33 Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.
- 34 Ran Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 11–20. ACM, 2005.
- 35 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 681–690. ACM, 2006.

## A The dependence of the seed on the non-malleability degree

In this section we extend the [21] result, where non-malleability was considered only in the case of  $t = 1$ . We repeat Theorem 6 and prove:

► **Theorem 20.** *Let  $n, k, t$  and  $\varepsilon$  be such that  $k \geq (t + 1)m + 2 \log \frac{1}{\varepsilon} + \log d + 4 \log t + 3$ . There exist a strong  $(k, \varepsilon)$   $t$ -n.m. extractor  $E: \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  with  $d \leq 2 \log \frac{1}{\varepsilon} + \log(n - k) + 2 \log(t + 1) + 3$ .*

This was also independently proved by Cohen and Shinkar [20].

**Proof.** Choose a function  $E: [N] \times [D] \rightarrow [M]$  uniformly at random. Fix a flat source  $X$  (which we identify with a subset  $X \subseteq [N]$  of size  $K$ ),  $t$  functions  $f_1, \dots, f_t: [D] \rightarrow [D]$  with no fixed-points and a distinguisher function  $\mathcal{D}: \{0, 1\}^{(t+1)m+d} \rightarrow \{0, 1\}$ . We want to bound the probability (over  $E$ ) that

$$\Pr[\mathcal{D}(E(X, Y), E(X, f_1(Y)), \dots, E(X, f_t(Y)), Y) = 1] - \Pr[\mathcal{D}(U_m, E(X, f_1(Y)), \dots, E(X, f_t(Y)), Y) = 1] > \varepsilon.$$

For every  $y \in [D]$  and  $z_1, \dots, z_t \in [M]$ , define

$$\text{Count}(y, z_1, \dots, z_t) = |\{z \in [M] : \mathcal{D}(z, z_1, \dots, z_t, y) = 1\}|.$$

For every  $x \in X$  and  $y \in [D]$ , define the following random variables (where the randomness comes from  $E$ ):

$$\begin{aligned}\mathbf{L}(x, y) &= \mathcal{D}(E(x, y), E(x, f_1(y)), \dots, E(x, f_t(y)), y), \\ \mathbf{R}(x, y) &= \frac{1}{M} \cdot \text{Count}(y, E(x, f_1(y)), \dots, E(x, f_t(y))), \\ \mathbf{Q}(x, y) &= \mathbf{L}(x, y) - \mathbf{R}(x, y), \\ \overline{\mathbf{Q}} &= \frac{1}{KD} \sum_{x \in X, y \in [D]} \mathbf{Q}(x, y).\end{aligned}$$

As we mentioned above, we want to bound  $\Pr[\overline{\mathbf{Q}} > \varepsilon]$ . Notice that for every  $x \in X$  and  $y \in [D]$ , due to the fact that  $f_1, \dots, f_t$  have no fixed points, we have that  $\mathbb{E}[\mathbf{L}(x, y)] = \mathbb{E}[\mathbf{R}(x, y)]$  and thus  $\mathbb{E}[\overline{\mathbf{Q}}] = 0$ . However, the values of  $\mathbf{Q}$  on different inputs are not independent.

To see why the  $\mathbf{Q}$ -s are not independent, think for example about the case where  $t = 2$  and  $y$  is such that  $f_2(f_1(y)) = y$ . In such a scenario,

$$\begin{aligned}\mathbf{L}(x, y) &= \mathcal{D}(E(x, y), E(x, f_1(y)), E(x, f_2(y)), y), \\ \mathbf{L}(x, f_1(y)) &= \mathcal{D}(E(x, f_1(y)), E(x, f_1(f_1(y))), E(x, y), f_1(y)),\end{aligned}$$

so, depending on  $\mathcal{D}$ ,  $\mathbf{Q}(x, y)$  and  $\mathbf{Q}(x, f_1(y))$  may not be independent. Luckily, it is sufficient to disregard such cycles in order to obtain sufficient “independence”.

Let  $G = (V = [D], E)$  be a directed graph (multiple edges allowed) such that

$$E = \{(y, f_k(y)) : y \in [D], k \in [t]\},$$

so the out-degree of every vertex is exactly  $t$ .

► **Lemma 21.** *Assume that there exists a subset  $V' \subseteq V$  such that the induced subgraph  $G' \subseteq G$  is acyclic. Then, the set  $\{\mathbf{Q}(x, y)\}_{x \in X, y \in V'}$  can be enumerated by  $\mathbf{Q}_1, \dots, \mathbf{Q}_{m=K|V'|}$  such that*

$$\mathbb{E}[\mathbf{Q}_i \mid \mathbf{Q}_1, \dots, \mathbf{Q}_{i-1}] = 0$$

for every  $i \in [m]$ .

**Proof.**  $G'$  is acyclic so it induces a partial order on  $V'$ . Use this partial order to induce a total order on  $\{1, \dots, m\}$  such that if  $(y, y') \in E$  and  $\mathbf{Q}_j = \mathbf{Q}(x, y')$ ,  $\mathbf{Q}_i = \mathbf{Q}(x, y)$  then  $j \leq i$ .

Fix some  $i \in [m]$  and assume  $\mathbf{Q}_i = \mathbf{Q}(x, y)$ . The key point is that the variables  $\mathbf{Q}_1, \dots, \mathbf{Q}_{i-1}$  never query  $E$  on the input  $(x, y)$ . Conditioned on any choice of the value of  $E$  for all points other than  $(x, y)$ , denote them by  $e_1, \dots, e_t$ , we have that

$$\mathbb{E}[\mathbf{Q}_i] = \mathbb{E}\left[\mathcal{D}(E(x, y), e_1, \dots, e_t, y) - \frac{1}{M} \cdot \text{Count}(y, e_1, \dots, e_t)\right] = 0,$$

and as we noted,  $\mathbf{Q}_1, \dots, \mathbf{Q}_{i-1}$  are deterministic functions of  $E$  and independent of  $E(x, y)$ . ◀

We now need a partition of the vertices of  $G$  into acyclic induced subgraphs. The following lemma shows that such a partition exists with a small number of sets.

► **Lemma 22** ([32, Corollary 4]). *For any directed graph  $G = (V, E)$  with maximum out-degree  $t$  (multiple edges allowed), there exists a partition  $V = V_1 \cup \dots \cup V_{t+1}$  such that for every  $i \in [t+1]$ , the subgraph of  $G$  induced by  $V_i$  is acyclic.*

In light of the above two lemmas, there exists a partition of  $\{\mathbf{Q}(x, y)\}_{x \in X, y \in [D]}$  to  $t + 1$  sets  $\{\mathbf{Q}_1^1, \dots, \mathbf{Q}_{s_1}^1\}, \dots, \{\mathbf{Q}_1^t, \dots, \mathbf{Q}_{s_t}^t\}$  such that for every  $k \in [t + 1]$  and  $i \in [s_k]$ ,  $\mathbb{E}[\mathbf{Q}_i^k \mid \mathbf{Q}_1^k, \dots, \mathbf{Q}_{i-1}^k] = 0$ . Now, define  $S_i^k = \sum_{j=1}^i \mathbf{Q}_j^k$  and note that every sequence  $S_1^k, \dots, S_{s_k}^k$  is a martingale. Also,  $|S_i^k - S_{i-1}^k| = |\mathbf{Q}_i^k| \leq 1$  with probability 1. Thus, using Azuma's inequality,

$$\begin{aligned} \Pr[\overline{\mathbf{Q}} > \varepsilon] &= \Pr\left[\sum_{k=1}^{t+1} S_{s_k}^k > \varepsilon KD\right] \leq \sum_{k=1}^{t+1} \Pr\left[S_{s_k}^k > \frac{\varepsilon KD}{t+1}\right] \\ &\leq \sum_{k=1}^{t+1} \exp\left(-\frac{\left(\frac{\varepsilon KD}{t+1}\right)^2}{2 \cdot s_k}\right) \leq (t+1)e^{-\frac{\varepsilon^2 KD}{2(t+1)^2}}, \end{aligned}$$

where the last inequality follows from the fact that  $s_k \leq KD$ .

To complete our analysis, we require  $E$  to work for *any*  $X, f_1, \dots, f_t$  and  $\mathcal{D}$ . By the union bound, the probability for a random  $E$  to fail, denote it by  $p_E$ , is given by

$$\begin{aligned} p_E &\leq \binom{N}{K} D^{tD} 2^{D \cdot M^{t+1}} (t+1) e^{-\frac{\varepsilon^2 KD}{2(t+1)^2}} \\ &\leq 2^{K \log\left(\frac{N\varepsilon}{K}\right) + tDd + DM^{t+1} + \log(t+1) - \frac{\varepsilon^2 KD \log e}{2(t+1)^2}} \\ &\leq 2^{K(n-k+2) + tDd + DM^{t+1} + \log(t+1) - \frac{\varepsilon^2 KD}{2(t+1)^2}}. \end{aligned}$$

To prove that  $p_E < 1$  (in fact this will show  $p_E \ll 1$ ) it is sufficient to prove that:

1.  $K(n - k + 2) \leq \frac{\varepsilon^2 KD}{8(t+1)^2}$ .
2.  $D(td + M^{t+1}) + \log(t + 1) \leq \frac{\varepsilon^2 KD}{8(t+1)^2}$ , or alternatively  $D(2td + M^{t+1}) \leq \frac{\varepsilon^2 KD}{8(t+1)^2}$ .

Item (1) is true whenever

$$D \geq \frac{8(t+1)^2(n-k+2)}{\varepsilon^2}.$$

Item (2) is true whenever

$$K \geq \frac{8(t+1)^2(2td + M^{t+1})}{\varepsilon^2}.$$


The bounds on  $d$  and  $k$  follow from the above two inequalities. ◀



# Lossless Dimension Expanders via Linearized Polynomials and Subspace Designs

Venkatesan Guruswami<sup>1</sup>

Department of Computer Science, Carnegie Mellon University  
5000 Forbes Ave, Pittsburgh, PA, USA, 15213  
venkatg@cs.cmu.edu


 <https://orcid.org/0000-0001-7926-3396>

Nicolas Resch<sup>2</sup>

Department of Carnegie Mellon University  
5000 Forbes Ave, Pittsburgh, PA, USA, 15213  
nresch@cs.cmu.edu

Chaoping Xing

School of Physical and Mathematical Sciences, Nanyang Technological University  
21 Nanyang Link, Singapore 637371  
xingcp@ntu.edu.sg

 <https://orcid.org/0000-0002-1257-1033>

---

## Abstract

---

For a vector space  $\mathbb{F}^n$  over a field  $\mathbb{F}$ , an  $(\eta, \beta)$ -dimension expander of degree  $d$  is a collection of  $d$  linear maps  $\Gamma_j : \mathbb{F}^n \rightarrow \mathbb{F}^n$  such that for every subspace  $U$  of  $\mathbb{F}^n$  of dimension at most  $\eta n$ , the image of  $U$  under all the maps,  $\sum_{j=1}^d \Gamma_j(U)$ , has dimension at least  $\beta \dim(U)$ . Over a finite field, a random collection of  $d = O(1)$  maps  $\Gamma_j$  offers excellent “lossless” expansion whp:  $\beta \approx d$  for  $\eta \geq \Omega(1/d)$ . When it comes to a family of *explicit constructions* (for growing  $n$ ), however, achieving even modest expansion factor  $\beta = 1 + \varepsilon$  with constant degree is a non-trivial goal.

We present an explicit construction of dimension expanders over finite fields based on linearized polynomials and subspace designs, drawing inspiration from recent progress on list-decoding in the rank-metric. Our approach yields the following:

- *Lossless* expansion over large fields; more precisely  $\beta \geq (1 - \varepsilon)d$  and  $\eta \geq \frac{1-\varepsilon}{d}$  with  $d = O_\varepsilon(1)$ , when  $|\mathbb{F}| \geq \Omega(n)$ .
- Optimal up to constant factors expansion over fields of arbitrarily small polynomial size; more precisely  $\beta \geq \Omega(\delta d)$  and  $\eta \geq \Omega(1/(\delta d))$  with  $d = O_\delta(1)$ , when  $|\mathbb{F}| \geq n^\delta$ .

Previously, an approach reducing to monotone expanders (a form of vertex expansion that is highly non-trivial to establish) gave  $(\Omega(1), 1 + \Omega(1))$ -dimension expanders of constant degree over all fields. An approach based on “rank condensing via subspace designs” led to dimension expanders with  $\beta \gtrsim \sqrt{d}$  over large fields. Ours is the first construction to achieve lossless dimension expansion, or even expansion proportional to the degree.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Randomness, geometry and discrete structures, Theory of computation  $\rightarrow$  Pseudorandomness and derandomization, Theory of computation  $\rightarrow$  Computational complexity and cryptography, Theory of computation  $\rightarrow$  Algebraic complexity theory

**Keywords and phrases** Algebraic constructions, coding theory, linear algebra, list-decoding, polynomial method, pseudorandomness

---

<sup>1</sup> Research supported in part by NSF grants CCF-1422045 and CCF-1563742.

<sup>2</sup> Research supported in part by NSF grants CCF-1618280, CCF-1422045, NSF CAREER award CCF-1750808 and NSERC grant CGSD2-502898.



Digital Object Identifier 10.4230/LIPIcs.CCC.2018.4

Related Version Full version available at <https://ecc.weizmann.ac.il/report/2018/017/>.

**Acknowledgements** Some of this work was done when the first author was visiting the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore.

## 1 Introduction

The field of *pseudorandomness* is concerned with efficiently constructing objects that share desirable properties with random objects while using no or little randomness. The ideas developed in pseudorandomness have found broad applications in areas such as complexity theory, derandomization, coding theory, cryptography, high-dimensional geometry, graph theory, and additive combinatorics. Due to much effort on the part of many researchers, nontrivial constructions of expander graphs, randomness extractors and condensers, Ramsey graphs, list-decodable codes, compressed sensing matrices, Euclidean sections, and pseudorandom generators and functions have been presented. Interestingly, while these problems may appear superficially to be unrelated, many of the techniques developed in one context have been useful in others, and the deep connections uncovered between these pseudorandom objects have led to a unified theory of “Boolean pseudorandomness”. (See for instance this survey by Vadhan [28] for more discussion of this phenomenon.)

More recently, there is a developing theory of “algebraic pseudorandomness,” wherein the pseudorandom objects of interest now have “algebraic structure” rather than a purely combinatorial structure. In these scenarios, instead of studying the size of subsets or min-entropy, we consider the dimension of subspaces. Many analogs of classical pseudorandom objects have been defined, such as dimension expanders, subspace-evasive sets, subspace designs, rank-preserving condensers, and list-decodable rank-metric codes. Beyond being interesting in their own rights, these algebraic pseudorandom objects have found many applications: for example, subspace-evasive sets have been used in the construction of Ramsey graphs [26] and list-decodable codes [19, 17]; subspace designs have been used to list-decode codes over the Hamming metric and the rank-metric [20, 17]; and rank-preserving condensers have been used in affine extractors [11] and polynomial identity testing [23, 9].

In this work, we focus upon providing explicit constructions of *dimension expanders* over finite fields. A dimension expander is a collection of  $d$  linear maps  $\Gamma_j : \mathbb{F}^n \rightarrow \mathbb{F}^n$  such that, for any subspace  $U \subseteq \mathbb{F}^n$  of sufficiently small dimension, the sum of the images of  $U$  under all the maps  $\Gamma_1(U) + \dots + \Gamma_d(U)$  has dimension which is a constant factor larger than  $\dim U$ . As suggested by their name, dimension expanders may be viewed as a linear-algebraic analog of expander graphs. Indeed, one can imagine creating a graph with vertex set  $\mathbb{F}^n$ , and then we add an edge from a vertex  $u \in \mathbb{F}^n$  to the vertices  $\Gamma_j(u)$ .<sup>3</sup> Alternatively, one may consider the bipartite graph with left and right partition given by  $\mathbb{F}^n$ , and we attach a vertex  $u \in \mathbb{F}^n$  in the left partition to  $\Gamma_j(u)$  in the right partition for each  $j$ . For this reason,  $d$  is referred to as the *degree* of the dimension expander. The property of being a dimension expander then says that, given any (sufficiently small) *subspace*, the span of the neighborhood will have appreciably larger dimension. Indeed, we use the notation  $\Gamma_j$  for the linear maps in analogy with the “neighborhood function” of a graph. Just as with expander graphs, we seek

<sup>3</sup> In general, this yields a directed graph. However, we may assume the maps  $\Gamma_j$  are invertible and then add the maps  $\Gamma_j^{-1}$  to the collection, which makes the graph undirected.

dimension expanders with constant degree, and moreover we would like to be able expand subspaces of dimension at most  $\eta n$  by a multiplicative factor of  $\beta$ , where  $\eta = \Omega(1)$  and  $\beta = 1 + \Omega(1)$ . We refer to such an object as an  $(\eta, \beta)$ -dimension expander. If  $\beta = \Omega(d)$ , we deem the dimension expander *degree-proportional*. If moreover  $\beta = (1 - \varepsilon)d$ , we deem the dimension expander *lossless*. Via a probabilistic argument, it is a simple exercise to show that constant-degree lossless dimension expanders exist over every field (see )

Finally, we indicate that *unbalanced* bipartite expander graphs play a key role in constructions of extractors and other Boolean pseudorandom objects. In this scenario, the left partition is significantly larger than the right partition, but we still have that sufficiently small subsets  $U$  of the left partition expand significantly, with  $(1 - \varepsilon)d|U|$  neighbors in the right partition in the lossless case. Such unbalanced expanders are closely related to *randomness condensers*, which preserve all or most of the min-entropy of a source while compressing its length. The improved min-entropy *rate* at the output makes subsequent *extraction* of near-uniformly random bits easier. Indeed, the extractors in [15] were obtained via this paradigm, once lossless expanders based on list-decodable codes were constructed. Inspired by this, we consider the challenge of constructing *unbalanced* dimension expanders: for  $N$  and  $n$  not necessarily equal, we would like a collection of maps  $\Gamma_1, \dots, \Gamma_d : \mathbb{F}^N \rightarrow \mathbb{F}^n$  that expand sufficiently small subspaces by a factor of  $\approx d$ . We quantify the “unbalancedness” of the dimension expander by  $b = \frac{N}{n}$ , and we refer to it as a *b-unbalanced dimension expander* in  $\mathbb{F}^n$ . Again, if the expansion factor is  $\Omega(d)$  we deem the unbalanced dimension expander *degree-proportional*, while if the expansion factor is  $(1 - \varepsilon)d$  we deem it *lossless*.

## 1.1 Our results

We provide various explicit constructions of dimension expanders. More precisely, we have a family of sets of matrices  $\{\{\Gamma_1^{(n_k)}, \dots, \Gamma_d^{(n_k)}\}\}_{k \in \mathbb{N}}$  for an infinite sequence of integers  $n_1 < n_2 < \dots$ , where each  $\Gamma_j^{(n_k)}$  is an  $n_k \times n_k$  matrix (or  $n_k \times bn_k$  matrix in the case of *b-unbalanced* expanders). The family is *explicit* if there is an algorithm outputting the list of matrices  $\Gamma_1^{(n_k)}, \dots, \Gamma_d^{(n_k)}$  in  $\text{poly}(n_k)$  field operations.

First of all, we provide the first explicit construction of a lossless dimension expander. Moreover we emphasize that the  $\eta$  parameter is optimal as well, as one cannot hope to expand subspaces of dimension more than  $\frac{n}{d}$  by a factor of  $\approx d$ .

► **Theorem 1.1** (Informal Statement; cf. Theorem 5.2). *For all  $\varepsilon > 0$  constant, there exists an integer  $d = d(\varepsilon)$  sufficiently large such that there is an explicit family of  $(\frac{1-\varepsilon}{d}, (1-\varepsilon)d)$ -dimension expanders of degree  $d$  over  $\mathbb{F}^n$  when  $|\mathbb{F}| \geq \Omega(n)$ .*

The main drawback of the above result is the constraint on the field size. Our next result allows for smaller field sizes, but we are only able to guarantee degree-proportional expansion. We remark that prior to this work, no explicit constructions of degree-proportional dimension expanders were known.

► **Theorem 1.2** (Informal Statement; cf. Theorem 5.1). *For all  $\delta > 0$  constant, there exists an integer  $d = d(\delta)$  sufficiently large such that there is an explicit family of  $(\Omega(\frac{1}{\delta d}), \Omega(\delta d))$ -dimension expanders of degree  $d$  over  $\mathbb{F}^n$  when  $|\mathbb{F}| \geq n^\delta$ .*

Moreover, our paradigm is flexible enough to allow for the construction of unbalanced dimension expanders. We remark that while the results of Forbes and Guruswami [8] could be adapted to obtain nontrivial constructions of unbalanced expanders, our work is the first to explicitly state this. Furthermore, our work is the first to achieve lossless expansion, or even degree-proportionality. Recall that we view unbalanced dimension expanders as

mapping  $\mathbb{F}^N \rightarrow \mathbb{F}^n$  and we call it  $b$ -unbalanced dimension expander over  $\mathbb{F}^n$  where  $b = \frac{N}{n}$ . Below we provide informal statements of our results; we refer to the full version for precise statements.

First, we provide a construction of a lossless unbalanced dimension expander, again over fields of linear size.

► **Theorem 1.3 (Informal Statement).** *For all  $\varepsilon > 0$  and integer  $b \geq 1$ , there exists an integer  $d = d(\varepsilon, b)$  sufficiently large such that there is an explicit family of  $b$ -unbalanced  $(\frac{1-\varepsilon}{db}, (1-\varepsilon)d)$ -dimension expanders of degree  $d$  over  $\mathbb{F}^n$  when  $|\mathbb{F}| \geq \Omega(n)$ .*

This result is again complemented by a construction of degree-proportional unbalanced dimension expanders over fields of arbitrarily small polynomial size.

► **Theorem 1.4 (Informal Statement).** *For all  $\delta > 0$  and integer  $b \geq 1$ , there exists an integer  $d = d(\delta, b)$  sufficiently large such that there is an explicit family of  $b$ -unbalanced  $(\Omega(\frac{1}{\delta b d}), \Omega(\delta d))$ -dimension expanders of degree  $d$  over  $\mathbb{F}^n$  when  $|\mathbb{F}| \geq n^\delta$ .*

## 1.2 Our approach

Our approach for constructing dimension expanders uses ideas recently developed in the context of list-decoding rank-metric codes. A *rank-metric code* is a set of matrices  $\mathcal{C} \subseteq \mathbb{F}^{m \times n}$  with  $m \geq n$ , and we define the *rank-distance* between matrices  $A, B$  to be  $d_R(A, B) = \text{rank}(A - B)$ . A code  $\mathcal{C}$  is said to be  $(\rho, L)$ -*list-decodable* if, for any  $Y \in \mathbb{F}^{m \times n}$ , the number of matrices in  $\mathcal{C}$  at rank-distance at most  $\rho n$  from  $Y$  is at most  $L$ . A line of work [18] succeeded in constructing high-rate rank-metric codes which are list-decodable up to the Singleton bound.<sup>4</sup> The code may also readily be seen to be *list-recoverable* in the following sense: given vector spaces  $V_1, \dots, V_n \subseteq \mathbb{F}^m$  of bounded dimension, the number of matrices in  $A \in \mathcal{C}$  with  $A_i \in V_i$  for all  $i \in [n]$  is bounded, where  $A_i$  denotes the  $i$ th column of  $A$ . The code constructed in [18] is a carefully selected subcode of the Gabidulin code [10], which is based on the evaluation of low degree *linearized* polynomials and is the analog of Reed-Solomon codes for the rank metric. Briefly, the Gabidulin code  $G[n, m, k, q]$  is obtained by evaluating linearized polynomials  $f(X) = \sum_{i=0}^{k-1} f_i X^{q^i} \in \mathbb{F}_{q^m}[X]$  at the  $\mathbb{F}_q$ -linearly independent points  $\alpha_1, \dots, \alpha_n \in \mathbb{F}_{q^m}$ , and then identifying the vector  $(f(\alpha_1), \dots, f(\alpha_n))$  with the matrix in  $\mathbb{F}_q^{m \times n}$  obtained by expressing  $f(\alpha_j) \in \mathbb{F}_{q^m}$  as an element of  $\mathbb{F}_q^m$  by fixing a basis for  $\mathbb{F}_{q^m}$  over  $\mathbb{F}_q$ . The  $q$ -degree of  $f = \sum_{i=0}^{k-1} f_i X^{q^i}$  is the maximal  $i$  such that  $f_i \neq 0$ .

In the case of Boolean pseudorandomness, not long after the construction of Parvaresh-Vardy codes and folded Reed-Solomon codes [25, 14], the techniques used to prove list-decodability of these codes were adapted to show lossless expansion properties of unbalanced expanders built from these codes [15]. Our approach is strongly inspired by the connection between list recovery and expansion that drives [15] and its instantiation with algebraic codes shown to achieve optimal redundancy for list decoding. Indeed, our methodology can be viewed as an adaption of the GUV approach to the “linearized world”. Various challenges arise in attempting to adapt the approach of the GUV framework to the setting of Gabidulin-like codes. For instance, we are no longer able to “append the seed” (in our context, the field element  $\alpha_j$ ) to the output of the neighborhood functions as is done in [15], as that will prevent the maps from being linear.<sup>5</sup> More significantly, we also need to perform

<sup>4</sup> The Singleton bound from coding theory over the Hamming metric possesses a natural analog in the rank-metric case.

<sup>5</sup> One could instead try tensoring the output with the seed, but it is unclear to us how to make this approach work without suffering a significant hit in the expansion factor.



a careful “pruning” of subspaces which arise in the analysis by exploiting the extra structure possessed by these subspaces. In turn this calls for better “subspace designs” which we construct. Broadly speaking, our approach necessitates the use of more sophisticated ideas from linear-algebraic list-decoding than were present in [15].

We now describe our approach in more detail. Let  $\mathbb{F}_{q^n}[X; (\cdot)^q]_{<k}$  denote the space of all linearized polynomials of  $q$ -degree less than  $k$ . We fix a subspace  $\mathcal{F} \subseteq \mathbb{F}_{q^n}[X; (\cdot)^q]_{<k}$  of dimension  $n$  over  $\mathbb{F}_q$ , and then each  $\Gamma_j$  is simply the evaluation of  $f \in \mathcal{F}$  at a point  $\alpha_j \in \mathbb{F}_{q^n}$ , i.e.,  $\Gamma_j(f) = f(\alpha_j)$ . We will in fact choose  $\alpha_1, \dots, \alpha_d$  to span a degree  $d$  field extension  $\mathbb{F}_h$  over  $\mathbb{F}_q$ .

The analysis of this construction mirrors the proof of the list-decodability of the codes from [18] and we sketch it here. In contrapositive, the dimension expander property amounts to showing that for every subspace  $V \subseteq \mathbb{F}_{q^n}$  of bounded dimension, the space of  $f \in \mathcal{F}$  such that  $f(\alpha_j) \in V \forall j \in [d]$  has dimension about a factor  $d$  smaller. So we study the structure of the space of polynomials  $f \in \mathbb{F}_{q^n}[X, (\cdot)^q]_{<k}$  which, for some fixed subspace  $V$ , have  $f(\alpha_j) \in V$  for all  $j \in [d]$ , and show that it forms a *periodic subspace* (cf. Definition 2.6). Thus, the challenge at this point is to find an appropriate subspace  $\mathcal{F} \subseteq \mathbb{F}_{q^n}[X; (\cdot)^q]_{<k}$  that has small intersection with *every* periodic subspace.

We accomplish this by using an appropriate construction of a *subspace design* (cf. Definition 2.5). Subspace designs were originally formulated for applications to algebraic list-decoding, where they led to optimal redundancy list-decodable codes over small alphabets [20] and over the rank-metric [18]. Briefly, subspace designs are collections of subspaces  $\{H_i\}_{i=1}^k$  such that, for any subspace  $W$  of bounded dimension, the total intersection dimension  $\sum_{i=1}^k \dim(H_i \cap W)$  is small. In fact, we will be interested in a slightly more general object: we are only required to have small intersection with  $\mathbb{F}_h$ -subspaces  $W$ , where we recall that  $\mathbb{F}_h$  is an extension field of  $\mathbb{F}_q$ . Once we have a good subspace design, it will suffice to define  $\mathcal{F} = \left\{ f(X) = \sum_{i=0}^{k-1} f_i X^{q^i} : f_i \in H_{i+1} \right\}$ .

Thus, we have reduced the task of constructing dimension expanders to the task of constructing subspace designs. We provide two constructions, yielding our two claimed constructions of dimension expanders. Both use an explicit subspace design given in [13] as a black box (cf. Lemma 4.1). We remark that in this work the authors only considered the  $d = 1$  case, i.e., the  $H_i$ 's were required to have small intersection with all  $\mathbb{F}_q$ -subspaces, and not just  $\mathbb{F}_h$ -subspaces. Thus, our task is easier in the sense that we only require intersection with  $\mathbb{F}_h$ -subspaces to be small. However, for our purposes, we will require a better bound on the total intersection dimension than that which is guaranteed by [13]. We also remark that this construction requires linear-sized fields which prevents us from obtaining dimension expanders over fields of subpolynomial size.

The subspace design which yields our degree-proportional expander is more elementary so we describe it first. Essentially, we take the subspace design of [13] and define it over an “intermediate field”  $\mathbb{F}_\ell$ , i.e.,  $\mathbb{F}_q \subseteq \mathbb{F}_\ell \subseteq \mathbb{F}_h$ . By appropriately choosing the degree of the extension we are able to guarantee smaller intersections with  $\mathbb{F}_h$ -subspaces and also allow  $q$  to be smaller (as it is now only  $\ell$  that must be linear in  $n$ , and we can take  $\ell \approx q^{1/\delta}$ ).

Our construction which yields lossless dimension expanders is more involved. We take the construction of [13] and now view it as lying in  $\mathbb{F}_q[Y]_{<\delta n}$  (for an appropriately chosen constant  $\delta > 0$ ), where  $\mathbb{F}_q[Y]_{<\delta n}$  denotes the  $\mathbb{F}_q$ -vector space of polynomials of degree  $< \delta n$ . We then map each of the subspaces into  $\mathbb{F}_h^{n/d}$  by evaluating the polynomials at a tuple of correlated degree  $d$  places (recall that  $h = q^d$ ). Identifying  $\mathbb{F}_h^{n/d}$  with  $\mathbb{F}_{q^n}$  completes the construction. Ideas similar to the linear algebraic list-decoding of folded Reed-Solomon

codes [12, 16] are used to prove the final bound on intersection dimension, which with a careful choice of parameters is good enough to guarantee lossless expansion. For technical reasons, in order to explicitly construct the degree  $d$  place we require  $n = q - 1$ .

### 1.3 Previous work

We now survey previous work on dimension expanders. Previous constructions have followed one of three main approaches: the first uses Cayley graphs of groups satisfying Kazhdan's property  $T$ , the second uses monotone expanders, and the third uses rank condensers.

#### 1.3.1 Property $T$

The problem of constructing dimension expanders was originally proposed by Wigderson [29, 1]. Along with the definition, he conjectured that dimension expanders could be constructed with Cayley graphs. This is in analogy with expander graphs, where such approaches have been very successful. To construct an expanding Cayley graph, one uses a group  $G$  with generating set  $S$  satisfying *Kazhdan's property  $T$* . Wigderson conjectured (see Dvir and Wigderson [7], Conjecture 7.1) that an expanding Cayley graph would automatically yield a dimension expander. More precisely, if one takes any irreducible representation  $\rho : G \rightarrow \text{GL}_n(\mathbb{F})$  of the group  $G$ , then  $\rho(S)$  would provide a dimension expander.

In characteristic zero, Lubotzky and Zelmanov [24] succeeded in proving Wigderson's conjecture. Unfortunately, their approach intrinsically uses the notion of unitarity which does not possess a meaningful definition over positive characteristic. They also provided an example of an expanding group whose linear representation over a finite field does *not* yield a dimension expander, although in the example the characteristic of the field divides the order of the group. In an independent work, Harrow [22] proved the same result in the context of *quantum expanders*, which imply dimension expanders in characteristic zero. The following theorem summarizes this discussion.

► **Theorem 1.5** ([24, 22]). *Let  $\mathbb{F}$  be a field of characteristic zero,  $n \geq 1$  an integer. There exists an explicit  $(1/2, 1 + \Omega(1))$ -dimension expander over  $\mathbb{F}^n$  of constant degree.*

Unfortunately, this approach is inherently unable to construct unbalanced dimension expanders. Moreover, it is unclear to us if it is possible to obtain expansion proportional to the degree via this strategy.

#### 1.3.2 Monotone expanders

Consider a bipartite graph  $G$  with left and right partition given by  $[n]$ , and let  $\Gamma_1, \dots, \Gamma_d : [n] \rightarrow [n]$  denote the neighbor (partial)<sup>6</sup> functions of the graph, i.e., each left vertex  $i \in [n]$  is connected to  $\Gamma_j(i)$  whenever it's defined. One can then define the linear maps  $\Gamma'_1, \dots, \Gamma'_d$  which map  $e_i \mapsto e_{\Gamma_j(i)}$  whenever  $\Gamma_j(i)$  is defined and then extending linearly, where the  $e_i$  are the standard basis vectors. It is easily seen that if  $G$  is an expander, the corresponding collection  $\{\Gamma'_j\}_{j=1}^d$  will expand subspaces of the form  $\text{span}\{e_i : i \in S\}$  for  $S \subseteq [n]$ . To expand all subspaces (and hence obtain dimension expanders), Dvir and Shpilka [6] implicitly observed that it is sufficient for the maps  $\Gamma_j$  to be *monotone* (this observation is made explicit in [7]). Note that the matrices  $\Gamma'_j$  have entries in  $\{0, 1\}$ , and they form a dimension expander over *every* field.

<sup>6</sup> That is,  $\Gamma_j$  need only be defined on a *subset* of  $[n]$ .

Thus, in order to construct dimension expanders, it suffices to construct monotone expander graphs. Unfortunately, constructing monotone expander graphs is a *highly* non-trivial task: indeed, the standard probabilistic arguments seem insufficient to even prove the *existence* of monotone expanders (see [7, 3]). Nonetheless, Dvir and Shpilka [5] succeeded in constructing monotone expanders with logarithmic degree, as well as constant-degree expanders with inverse-logarithmic expansion. Later, using the zig-zag product of Reingold, Vadhan and Wigderson [27], Dvir and Wigderson [7] constructed monotone expanders of degree  $\log^{(c)} n$  (the  $c$ -th iterated logarithm) for any constant  $c$ . Moreover, given any constant-degree monotone expander as a starting point (which is not known to exist via the probabilistic method), their method is capable of constructing a constant degree monotone expander graph. Lastly, by a sophisticated analysis of expansion in the group  $\text{SL}_2(\mathbb{R})$ , Bourgain and Yehudayoff [3] were able to construct explicit monotone expanders of constant degree. Thus, we have the following theorem.

► **Theorem 1.6** ([3]). *Let  $n \geq 1$  be an integer. There exists an explicit  $(1/2, 1 + \Omega(1))$ -dimension expander of degree  $O(1)$  over  $\mathbb{F}^n$ , for every field  $\mathbb{F}$ .*

Unfortunately, just as with the previous approach, it is unclear to us if this argument could be adapted to yield degree-proportional dimension expanders.

### 1.3.3 Rank condensers

This final approach to constructing dimension expanders, developed by Forbes and the first author [8], uses *rank condensers*. Unlike the constructions of the previous sections, it inherently uses ideas from algebraic pseudorandomness and thus is most in the spirit of our work. The construction proceeds in two steps. First, one “trivially” expands the subspaces by a factor of  $d$  by defining  $T_j : \mathbb{F}^n \rightarrow \mathbb{F}^n \otimes \mathbb{F}^d$  mapping  $v \mapsto v \otimes e_j$ . The challenge is then to map  $\mathbb{F}^n \otimes \mathbb{F}^d \cong \mathbb{F}^{nd}$  back to  $\mathbb{F}^n$  such that subspaces do not decrease in dimension too much. This is precisely the problem of *lossy rank condensing*, namely, of constructing a small collection of linear maps  $S_k : \mathbb{F}^{nd} \rightarrow \mathbb{F}^n$  such that, for any subspace  $U$  of bounded degree, there exists some  $S_k$  such that  $\dim S_k(U) \geq (1 - \varepsilon) \dim U$ . To complete the construction, one takes the set of all  $S_k T_j$ . We remark that the construction of the rank condenser from this work used the subspace designs of [13], providing more evidence for the interrelatedness of the objects studied in algebraic pseudorandomness. Unfortunately, the construction of subspace designs used in this work require polynomially large fields. The authors are able to decrease the field size using techniques reminiscent of code-concatenation at the cost of certain logarithmic penalties.

The following theorem was obtained.

► **Theorem 1.7** ([8]).

1. *Let  $n, d \geq 1$ . Assume  $|\mathbb{F}| \geq \Omega(n^2)$ . There exists an explicit  $(\Omega(1/\sqrt{d}), \Omega(\sqrt{d}))$ -dimension expander in  $\mathbb{F}^n$  of degree  $d$ .*
2. *Let  $\mathbb{F}_q$  be a finite field,  $n, d \geq 1$ . There exists an explicit  $(\Omega(1/d \log_q(dn)), \Omega(d))$ -dimension expander in  $\mathbb{F}_q^n$  of degree  $O(d^2 \log_q(dn))$ .*

In order to improve the dependence on the field size, improved subspace designs over small fields were constructed by Guruswami, Xing and Yuan [21]. These subspace designs yield a family of explicit  $(\Omega(1/\log_q \log_q n), 1 + \Omega(1))$ -dimension expander of degree  $O(\log_q n)$  over  $\mathbb{F}_q^n$ .

## 1.4 Organization

In Section 2 we set notation and define the various pseudorandom objects that we use in our construction. We also provide probabilistic arguments ascertaining the existence of good dimension expanders in order to set expectations. In Section 3 we prove that the problem of constructing dimension expanders can be reduced to that of constructing appropriate subspace designs, which is the task we address in Section 4. In Section 5, we put all of the pieces together to deduce our main theorems for balanced dimension expanders (for our results on unbalanced dimension expanders, we refer to the full version of the paper). We summarize our work and list open problems in Section 6.

## 2 Background

### 2.1 Notation

First, we briefly summarize the notation that we will use regularly (other notation will be introduced as needed).  $\mathbb{F}$  will always refer to an arbitrary field,  $q$  always denotes a prime power, and  $\mathbb{F}_q$  denotes the finite field with  $q$  elements. We denote  $[n] := \{1, \dots, n\}$ . We write  $a|b$  to assert that the integer  $a$  divides the integer  $b$  without remainder.

Given a subspace  $U \subseteq \mathbb{F}^n$  and a linear map  $T : \mathbb{F}^n \rightarrow \mathbb{F}^m$ ,  $T(U) = \{Tu : u \in U\}$  denotes the image of the subspace  $U$  under the map  $T$ . Given two subspaces  $U, V \subseteq \mathbb{F}^n$ ,  $U + V = \{u + v : u \in U, v \in V\}$  denotes their sum, which is also a subspace.

The finite field with  $q^n$  elements, i.e.,  $\mathbb{F}_{q^n}$ , has the structure of a vector space over  $\mathbb{F}_q$  of dimension  $n$ . Thus, we often identify  $\mathbb{F}_{q^n}$  with  $\mathbb{F}_q^n$ . Moreover, if  $h = q^d$  is a power of  $q$  and  $d|n$ , so  $\mathbb{F}_h \subseteq \mathbb{F}_{q^n}$ , the field  $\mathbb{F}_{q^n}$  also has the structure of a vector space over  $\mathbb{F}_h$  of dimension  $n/d$ . Throughout this work, we will always assume  $d|n$  and write  $n = md$ .

We will sometimes have subspaces of  $W \subseteq \mathbb{F}_{q^n}$  that are linear over  $\mathbb{F}_h$ , i.e., for all  $w \in W$  and  $\alpha \in \mathbb{F}_h$  we have  $\alpha w \in W$ . When we wish to emphasize this, we will say that  $W$  is an  $\mathbb{F}_h$ -subspace. Moreover, we will write  $\dim_{\mathbb{F}_q} W$  or  $\dim_{\mathbb{F}_h} W$  if we need to emphasize that the dimension is computed when viewing  $W$  as an  $\mathbb{F}_q$ -subspace or as an  $\mathbb{F}_h$ -subspace, respectively.

A  $q$ -linearized polynomial  $f$  is a polynomial of the form  $f(X) = \sum_{i=0}^{k-1} f_i X^{q^i}$ . We denote the space of  $q$ -linearized polynomials with coefficients in  $\mathbb{F}_{q^n}$  as  $\mathbb{F}_{q^n}[X; (\cdot)^q]$ . The  $q$ -degree of a linearized polynomial  $f(X) = \sum_{i=0}^{k-1} f_i X^{q^i}$  is the maximum  $i$  such that  $f_i \neq 0$ , and is denoted  $\deg_q f$ . We denote  $\mathbb{F}_{q^n}[X; (\cdot)^q]_{<k} = \{f \in \mathbb{F}_{q^n}[X; (\cdot)^q] : \deg_q f < k\}$ , which we remark is a  $k$ -dimensional vector space over  $\mathbb{F}_{q^n}$ .

Note that if  $\alpha, \beta \in \mathbb{F}_{q^n}$  and  $a, b \in \mathbb{F}_q$  then for any  $f \in \mathbb{F}_{q^n}[X; (\cdot)^q]$ ,  $f(a\alpha + b\beta) = af(\alpha) + bf(\beta)$ , i.e.,  $f$  gives an  $\mathbb{F}_q$ -linear map from  $\mathbb{F}_{q^n} \rightarrow \mathbb{F}_{q^n}$ . Moreover, the space of roots of such an  $f$  is an  $\mathbb{F}_q$ -subspace of dimension at most  $\deg_q f$  (assuming  $f \neq 0$ ).

### 2.2 Dimension expanders

We now formally define dimension expanders and provide an alternate characterization that we find easier to reason about.

► **Definition 2.1** (Dimension expander). Let  $n, d \geq 1$  be an integer,  $\eta > 0$  and  $\beta > 1$ . Let  $\Gamma_1, \dots, \Gamma_d : \mathbb{F}^n \rightarrow \mathbb{F}^n$  be linear maps. The collection  $\{\Gamma_j\}_{j=1}^d$  forms a  $(\eta, \beta)$ -dimension expander if for all subspaces  $U \subseteq \mathbb{F}^n$  of dimension at most  $\eta n$ ,

$$\dim \left( \sum_{j=1}^d \Gamma_j(U) \right) \geq \beta \dim U .$$

The *degree* of the dimension expander is  $d$ .

When clear from context we refer to a dimension expander just as an *expander*. The following proposition follows easily from the definitions.

► **Proposition 2.2** (Contrapositive characterization). *Let  $n \geq 1$  be an integer,  $\eta > 0$  and  $\beta > 1$ . Let  $\Gamma_1, \dots, \Gamma_d : \mathbb{F}^n \rightarrow \mathbb{F}^n$  be linear maps. Suppose that for all  $V \subseteq \mathbb{F}^n$  of dimension at most  $\eta\beta n$ ,*

$$\dim \{u \in \mathbb{F}^n : \Gamma_j(u) \in V \ \forall j \in [d]\} \leq \frac{1}{\beta} \dim V .$$

*Then  $\{\Gamma_j\}_{j=1}^d$  forms an  $(\eta, \beta)$ -dimension expander.*

Next, we define a slight generalization of dimension expanders, wherein the domain and codomain may no longer have the same dimension. That is, the linear maps  $\Gamma_j$  now map  $\mathbb{F}^N \rightarrow \mathbb{F}^n$ , where  $N, n$  may not be equal. We parametrize the “unbalancedness” of the dimension expander by  $b = \frac{N}{n}$ . In our construction we will assume for simplicity that  $b \in \mathbb{Z}$ , although we note that this is not a fundamental restriction. The formal definition is as follows.

► **Definition 2.3** (Unbalanced dimension expanders). *Let  $N, n, d \geq 1$  be integers,  $\eta > 0$  and  $\beta > 1$ . Let  $\Gamma_1, \dots, \Gamma_d : \mathbb{F}^N \rightarrow \mathbb{F}^n$  be linear maps. Set  $b = \frac{N}{n}$ . The collection  $\{\Gamma_j\}_{j=1}^d$  forms a  $b$ -unbalanced  $(\eta, \beta)$ -dimension expander if for all subspaces  $U \subseteq \mathbb{F}^N$  of dimension at most  $\eta N$ ,*

$$\dim \left( \sum_{j=1}^d \Gamma_j(U) \right) \geq \beta \dim U .$$

The *degree* of the unbalanced dimension expander is  $d$ .

Lastly, we state the parameters achievable via the probabilistic method in order to set expectations.

► **Proposition 2.4** (Simple generalization of Proposition C.10 of [8]). *Let  $\mathbb{F}_q$  be a finite field,  $N, n$  positive integers and put  $b := \frac{N}{n}$ . Let  $\beta > 1$  and  $\eta \in (0, \frac{1}{b\beta})$ . Then, assuming*

$$d \geq \beta + \frac{b}{1 - b\beta\eta} + \log_q 16 ,$$

*there exists a collection of linear maps  $\Gamma_1, \dots, \Gamma_d : \mathbb{F}_q^N \rightarrow \mathbb{F}_q^n$  forming a  $(\eta, \beta)$ -unbalanced dimension expander.*

Thus, for  $b = 1$ , if we wish to have  $\beta = (1 - \varepsilon)d$  and  $\eta = \frac{1-\varepsilon}{d}$  we may take  $d = O(1/\varepsilon^2)$ . We remark that in Theorem 5.2, we obtain  $d = O(1/\varepsilon^3)$ .

### 2.3 Subspace design

A crucial ingredient in our construction of dimension expanders are subspace designs. They were originally introduced by two of the authors [20] in order to obtain algebraic codes list-decodable up to the Singleton bound. As in [18], we will be concerned with a slight weakening of this notion, where we are only concerned with having small intersection with subspaces which are linear over an extension of the base field, although we will also require the intersection dimension to be smaller.

► **Definition 2.5.** Let  $V$  be a  $\mathbb{F}_{q^d}$ -vector space. A collection  $H_1, \dots, H_k \subseteq V$  of  $\mathbb{F}_q$ -subspaces is called a  $(s, A, d)$ -subspace design in  $V$  if for every  $\mathbb{F}_{q^d}$ -subspace  $W \subseteq V$  of  $\mathbb{F}_{q^d}$ -dimension  $s$ ,

$$\sum_{i=1}^k \dim_{\mathbb{F}_q}(H_i \cap W) \leq As .$$

We call a subspace design *explicit* if there is an algorithm outputting  $\mathbb{F}_q$ -bases for each subspace  $H_i$  in  $\text{poly}(n)$  field operations.

► **Remark.** In previous works, what we have termed a  $(s, A, d)$ -subspace design would have been called a  $(s, As, d)$ -subspace design. We find it more convenient in this work to remove the multiplicative factor of  $s$  from the parameter in the definition.

## 2.4 Periodic subspaces

We now abstract the kind of structure that will be found in the subspace of  $\mathbb{F}_q^n$  which is mapped entirely into a low-dimensional subspace of  $\mathbb{F}_q^n$  by the  $d$  linear transformations in our dimension expander construction. We note that our definition here is slightly different in form and notation than earlier ones in [20, 18].

► **Definition 2.6** (Periodic subspaces). For positive integers  $n, k, s, d$  with  $d|n$ , an  $\mathbb{F}_q$ -subspace  $T$  of  $\mathbb{F}_q^n$  is said to be  $(s, d)$ -periodic if there exists an  $\mathbb{F}_{q^d}$ -subspace  $W \subseteq \mathbb{F}_{q^d}^n$  of dimension at most  $s$  such that for all  $j$ ,  $1 \leq j \leq k$ , and all  $\xi_1, \xi_2, \dots, \xi_{j-1} \in \mathbb{F}_{q^d}$ , the  $\mathbb{F}_q$ -affine subspace

$$\{\xi_j : \exists v \in T \text{ with } v_\iota = \xi_\iota \text{ for } 1 \leq \iota \leq j\} \subseteq \mathbb{F}_q^n$$

belongs to a coset of  $W$ . In other words, for every *prefix*  $(\xi_1, \dots, \xi_{j-1})$ , the possible extensions  $\xi_j$  to the  $j$ 'th symbol that can belong to a vector in  $T$  are contained in a coset of  $W$ .

An important property of periodic subspaces is that they have small intersection with subspace designs. This is captured by the following proposition.

► **Proposition 2.7** ([18], Proposition 3.9). *Let  $T$  be a  $(s, d)$ -periodic  $\mathbb{F}_q$ -subspace of  $\mathbb{F}_q^n$ , and  $H_1, \dots, H_k \subseteq \mathbb{F}_q^n$  be  $\mathbb{F}_q$ -subspaces forming a  $(s, A, d)$  subspace design in  $\mathbb{F}_q^n$ . Then  $T \cap (H_1 \times \dots \times H_k)$  is an  $\mathbb{F}_q$ -subspace of dimension at most  $As$ .*

## 3 Dimension expander construction

As discussed in the introduction (Section 1), the construction of our dimension expander is inspired by recent constructions of variants of Gabidulin codes for list-decoding in the rank-metric. Indeed, the analysis of our dimension expander proceeds similarly to the analysis of list-decodability of the rank-metric codes presented in [18]. The presentation here is self-contained algebraically, and does not refer to any coding-theoretic context or language.

### 3.1 Construction

Our dimension expanders map  $\mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ . We view the domain as

$$\mathcal{F} := \left\{ f(X) = \sum_{i=0}^{k-1} f_i X^{q^i} : f_i \in H_i, i = 0, \dots, k-1 \right\}$$

where  $H_0, \dots, H_{k-1}$  give a collection of  $\mathbb{F}_q$ -subspaces of  $\mathbb{F}_{q^n}$ , each of  $\mathbb{F}_q$ -dimension  $\frac{n}{k}$  (thus, we assume  $k|n$ ). We will choose  $H_1, H_2, \dots, H_k$  forming a subspace design. We view the image space as  $\mathbb{F}_{q^n}$ . Let  $h = q^d$ , and let  $\alpha_1, \dots, \alpha_d$  give a basis for  $\mathbb{F}_h$  over  $\mathbb{F}_q$ . We assume  $d|n$  and write  $md = n$ . For  $j = 1, \dots, d$ , we define

$$\Gamma_j : \mathcal{F} \rightarrow \mathbb{F}_{q^n} \quad \text{by} \quad f \mapsto f(\alpha_j) . \tag{1}$$

That is, each  $\Gamma_j(f)$  is just the evaluation of  $f$  at the basis element  $\alpha_j$ . These maps are clearly linear over  $\mathbb{F}_q$ .

### 3.2 Analysis

We now state the steps involved in showing that the collection  $\{\Gamma_j\}_{j=1}^d$  forms a dimension expander. We have omitted the proofs; they can be found in the full version of the paper.

For a positive integers  $D, s$  with  $s \leq m$ , we define  $\mathcal{L}_{D,s}$  to be the space of polynomials  $Q \in \mathbb{F}_{q^n}[Z_0, \dots, Z_{s-1}]$  of the form  $Q(Z_0, \dots, Z_{s-1}) = A_0(Z_0) + \dots + A_{s-1}(Z_{s-1})$  with each  $A_i \in \mathbb{F}_{q^n}[X; (\cdot)^q]_{<D}$ , i.e., each  $A_i$  is a  $q$ -linearized polynomial of  $q$ -degree at most  $D - 1$ .

► **Lemma 3.1.** *Let  $V \subseteq \mathbb{F}_{q^n}$  be an  $\mathbb{F}_q$ -subspace of dimension  $B$ . If  $Ds > B$ , there exists a nonzero polynomial  $Q \in \mathcal{L}_{D,s}$  such that*

$$\forall v \in V, \quad Q(v, v^h, \dots, v^{h^{s-1}}) = 0 . \tag{2}$$

Given a polynomial  $g(X) = g_0 + g_1X + \dots + g_rX^r$  and an automorphism  $\tau$  of  $\mathbb{F}_{q^n}$ , we write  $g^\tau$  for the polynomial  $g^\tau(X) = \tau(g_0) + \tau(g_1)X + \dots + \tau(g_r)X^r$ , and let  $g^{\tau^i} = (g^{\tau^{i-1}})^\tau$ . We let  $\sigma : \gamma \mapsto \gamma^h$ , i.e.,  $\sigma$  is the Frobenius automorphism of  $\mathbb{F}_{h^m} = \mathbb{F}_{q^n}$  over  $\mathbb{F}_h$ .

► **Lemma 3.2.** *Let  $f \in \mathbb{F}_{q^n}[X]$  be a  $q$ -linearized polynomial with  $q$ -degree at most  $k - 1$ . Let  $V \subseteq \mathbb{F}_{q^n}$  be an  $\mathbb{F}_q$ -subspace, and  $Q \in \mathcal{L}_{D,s}$  a polynomial satisfying (2). Suppose that  $f(\alpha) \in V$  for all  $\alpha \in \mathbb{F}_h = \mathbb{F}_{q^d}$  and that  $D \leq d - k + 1$ . Then*

$$A_0(f(X)) + A_1(f^\sigma(X)) + \dots + A_{s-1}(f^{\sigma^{s-1}}(X)) = Q(f(X), f^\sigma(X), \dots, f^{\sigma^{s-1}}(X)) = 0 . \tag{3}$$

► **Lemma 3.3.** *The set of solutions to Equation (3), for any nonzero  $Q \in \mathcal{L}_{D,s}$  (for arbitrary  $D$ ), is an  $(s - 1, d)$ -periodic subspace.*

Equipped with these lemmas, we are in position to deduce our main theorem for this section.

► **Theorem 3.4.** *Let  $\{H_i\}_{i=0}^{k-1}$  give a  $(s, A, d)$ -subspace design for all  $s \leq \mu n$  for some  $0 < \mu < 1/d$ . Then  $\{\Gamma_j\}_{j=1}^d$  is a  $(\mu A, \frac{d-k+1}{A})$ -dimension expander. Moreover if the subspace design is explicit then the dimension expander is explicit.*

Thus, we have that subspaces of dimension  $As$  are expanded to subspaces of dimension  $(d - k + 1)s/A$ . This informs what we should hope for from our subspace designs. In particular, obtaining  $A = O(1)$  is enough to obtain a degree proportional expander (by setting  $k = \Theta(d)$ ), while if  $A \approx 1 + \varepsilon$  and  $k \approx \varepsilon d$  we can obtain a *lossless* expander. With these goals in mind, we turn our attention to constructing subspace designs.



## 4 Constructions of subspace designs

For the case of  $d = 1$ , explicit constructions of subspace designs have been given in previous works. The first explicit construction was given in [13], using ideas which had been developed in constructions of list-decodable codes. This construction was subsequently improved over fields of small size in [21].

A previous construction of a subspace design for  $d > 1$  was given in [18]. In this work, a subspace design over the base field (i.e., for  $d = 1$ ) was intersected with a *subspace evasive set* from [4]. However, for our purposes, the size of the intersection dimension (i.e., the product  $As$ ) of this construction is too large. In that work, the authors were more concerned with ensuring that the  $H_i$ 's had large dimension; however, we only require that the  $H_i$ 's have dimension  $n/k$ .

We provide two constructions of subspace designs in this work, yielding our two constructions of dimension expanders. The first construction yields a *degree-proportional* dimension expander over fields of size  $n^\delta$  (for arbitrarily small constant  $\delta$ ). The next yields a *lossless* dimension expander. The only drawback is that it requires a field of size linear in  $n$ .<sup>7</sup> We present our first construction in Section 4.1 and our second construction in Section 4.2. The full version contains all of the proofs that we have removed from this section.

Both of our constructions use as a black box a subspace design provided in [13]. Specifically, by taking  $r = 2$  in Theorem 7 of [13], we obtain a subspace design with the following parameters.

► **Lemma 4.1.** *For all positive integers  $s, t, m$  and prime powers  $\ell$  satisfying  $s \leq t \leq m < \ell$ , there is an explicit collection of  $M \geq \frac{\ell^2}{4t}$   $\mathbb{F}_\ell$ -spaces  $V_1, V_2, \dots, V_M \subseteq \mathbb{F}_\ell^m$ , each of codimension  $2t$ , which forms an  $(s, \frac{m-1}{2(t-s+1)}, 1)$  subspace design in  $\mathbb{F}_\ell^m$ .*

### 4.1 Subspace designs via an intermediate field

This first construction takes the subspace design of Lemma 4.1 defined over an intermediate field  $\mathbb{F}_\ell$ . That is, we fix an integer  $1 < c < d$  such that  $c|d$  so that, for  $\ell = q^c$ ,  $\mathbb{F}_q \subseteq \mathbb{F}_\ell \subseteq \mathbb{F}_h$ . Then, if  $\omega_1, \dots, \omega_m$  gives a basis for  $\mathbb{F}_{h^m}/\mathbb{F}_h$ , define

$$L = \left\{ \sum_{i=1}^m a_i \omega_i : a_i \in \mathbb{F}_\ell \right\}.$$

This is an  $\mathbb{F}_\ell$ -subspace of  $\mathbb{F}_{h^m} = \mathbb{F}_{q^n}$  of  $\mathbb{F}_\ell$ -dimension  $m$ , as  $\omega_1, \dots, \omega_m$  are linearly independent over  $\mathbb{F}_h$  and so *a fortiori* are linearly independent over the subfield  $\mathbb{F}_\ell$ . Thus,  $L \simeq \mathbb{F}_\ell^m$ , and we fix an  $\mathbb{F}_\ell$ -linear isomorphism  $\psi : \mathbb{F}_\ell^m \rightarrow L$ . Note that an  $\mathbb{F}_\ell$ -linear map is automatically  $\mathbb{F}_q$ -linear, so, in particular, the dimension of  $\mathbb{F}_q$ -subspaces in  $\mathbb{F}_\ell^m$  are preserved by  $\psi$ . Then, if  $V_1, \dots, V_k$  give the subspace design from Lemma 4.1, we define  $H_i := \psi(V_i)$  for  $i = 1, \dots, k$ .

► **Proposition 4.2.** *Let  $\ell = q^c$  with  $c = \frac{d}{k} \cdot \frac{m}{m-2t}$ , where  $1 \leq k < d$ . For all  $1 \leq s < t < \ell$  and  $1 \leq k < d$  such that  $\ell^2 \geq 4kt$ ,  $k|d$ ,  $m|k(m-2t)$  and  $k(m-2t)|n$ , there is an explicit construction of  $\{H_i\}_{i=1}^k$  that forms a  $(s, \frac{d}{k} \cdot \frac{m-1}{m-2t} \cdot \frac{m}{2(t-s)}, d)$ -subspace design in  $\mathbb{F}_{q^n}$ . Furthermore  $\dim_{\mathbb{F}_q} H_i = \frac{n}{k}$  for all  $i = 1, \dots, k$ .*

We now fix parameters in such a way to show that we can obtain a subspace design over fields of size  $n^\delta$  for any constant  $\delta > 0$ .

<sup>7</sup> In fact, in order to ensure our construction is algorithmically explicit, we take  $q - 1 = n$ .



► **Corollary 4.3.** *Let  $\delta > 0$  be given and choose an integer  $r$  such that  $\frac{1}{2\delta} < r \leq \frac{1}{\delta}$ . Let  $k, d$  be integers such that  $d = 2k$  and  $r|k$ . Assume moreover that  $2r|m$ . Then, assuming  $q \geq n^\delta$ , there exists an explicit construction of  $\{H_i\}_{i=1}^k$  that forms a  $(s, \frac{s}{\delta}, d)$ -subspace design in  $\mathbb{F}_{q^n}$  for all  $s \leq \frac{1-2\delta}{4d}n$ . Moreover  $\dim_{\mathbb{F}_q} H_i = \frac{n}{k}$  for all  $i = 1, \dots, k$ .*

## 4.2 Construction via correlated high-degree places

This next construction utilizes techniques developed in the context of linear algebraic list-decoding of folded Reed-Solomon codes [12, 16]. Briefly, we take a subspace design in the space of polynomials of bounded degree, and then map it into  $\mathbb{F}_h^m$  in a manner reminiscent of the encoding map of the folded Reed-Solomon code. As we are concerned with bounding the intersection dimension with  $\mathbb{F}_h$ -linear spaces, we in fact evaluate the polynomial at degree  $d$  places. The details follow.

Let  $\zeta$  be a primitive root of the finite field  $\mathbb{F}_q$ . Choose a real  $\delta \in (0, 1)$  such that  $\delta > \frac{1}{k}$  and  $\delta n < q - 1$ , where we recall  $0 < k < d$  and  $n = md$ . Denote by  $\sigma$  the automorphism of the function field  $\mathbb{F}_q(Y)$  sending  $Y$  to  $\zeta Y$ . The order of  $\sigma$  is  $q - 1 \geq m$ . Given  $g \in \mathbb{F}_q(Y)$ , we abbreviate  $g^\sigma := \sigma(g(Y)) = g(\zeta Y)$ .<sup>8</sup>

Denote by  $\mathbb{F}_q[Y]_{<\delta n}$  the set of polynomials of degree less than  $\delta n$ . By Lemma 4.1, there exist  $V_1, V_2, \dots, V_k$  of  $\mathbb{F}_q[Y]_{<\delta n}$ , each of codimension  $\delta n - \frac{n}{k}$ , which forms a  $(r, \frac{\delta n - 1}{\delta n - \frac{n}{k} - 2r + 2}, 1)$  subspace design.

Let  $P(Y)$  be an irreducible polynomial of degree  $d$  such that  $P, P^\sigma, \dots, P^{\sigma^{m-1}}$  are pairwise coprime. Consider the map

$$\pi : \mathbb{F}_q[Y]_{<\delta n} \rightarrow \mathbb{F}_{q^d}^m, \quad f \mapsto (f(P), f(P^\sigma), \dots, f(P^{\sigma^{m-1}})),$$

where  $f(P^{\sigma^j})$  is viewed as the residue of  $f$  in the residue field  $\mathbb{F}_q[Y]/(P^{\sigma^j}) \cong \mathbb{F}_{q^d} = \mathbb{F}_h$ . The Chinese Remainder Theorem guarantees that  $\pi$  is injective. We define

$$\tilde{H}_i = \pi(V_i) = \left\{ (f(P), f(P^\sigma), \dots, f(P^{\sigma^{m-1}})) : f \in V_i \right\} \subseteq \mathbb{F}_h^m \tag{4}$$

for  $i = 1, 2, \dots, k$ .

We remark that this  $\pi$  is reminiscent of the encoding map of the folded Reed-Solomon code (recall that  $P^\sigma = P(\zeta Y)$ ), although in this case we evaluate  $f$  at the high-degree place  $P$ .

► **Proposition 4.4.** *If  $s < (1 - \delta)m = (1 - \delta)\frac{n}{d}$ , then the subspaces  $\tilde{H}_1, \tilde{H}_2, \dots, \tilde{H}_k$  defined above is an  $(s, \frac{\delta}{1-\delta} \cdot \frac{m}{(\delta - \frac{1}{k})m - \frac{2s}{d(1-\delta)}})$ -subspace design in  $\mathbb{F}_h^m$ . Moreover  $\dim_{\mathbb{F}_q} \tilde{H}_i = \frac{n}{k}$  for all  $i = 1, \dots, k$ .*

*Lastly, when  $n = q - 1$ , the subspace design can be constructed explicitly.*

By choosing  $k, d$  and appropriately we obtain the following corollary.

► **Corollary 4.5.** *Let  $\delta > 0$  be such that  $1/\delta \in \mathbb{Z}$  and put  $k = 1/\delta^2$ ,  $d = 1/\delta^3$ . Assume that  $q - 1 = n$ . There exist  $H_1, \dots, H_k$  which form an explicit  $(s, \frac{1}{1-2\delta-\delta^2+2\delta^3})$ -subspace design in  $\mathbb{F}_{q^n}$  for all  $s \leq \frac{1-2\delta}{d}n$ . Moreover  $\dim_{\mathbb{F}_q} H_i = \frac{n}{k}$  for all  $i = 1, \dots, k$ .*

<sup>8</sup> Note that in Section 3 we wrote  $g^\sigma$  to denote the polynomial obtained by applying  $\sigma$  to the coefficients of  $g$ . We hope that this notation does not cause any confusion.

## 5 Explicit instantiations of dimension expanders

As outlined in Section 3, our approach for obtaining explicit constructions of dimension expanders is by reducing to the construction of subspace designs. Specifically, we will apply Theorem 3.4 with the constructions of Section 4. These results yield Theorems 1.2 and 1.1, respectively.

First, using the subspace design constructed in Corollary 4.3, we obtain a degree-proportional dimension expander over fields of arbitrarily small polynomial size.

► **Theorem 5.1.** *Let  $\delta > 0$  be given and assume  $|\mathbb{F}_q| \geq n^\delta$ . Let  $r$  be an integer satisfying  $\frac{1}{2\delta} \leq r < \frac{1}{\delta}$ , let  $k$  be a multiple of  $r$ , and let  $d = 2k$ . There exists an explicit construction of a  $(\eta, \beta)$ -dimension expander of degree  $d$  over  $\mathbb{F}_q^n$  whenever  $2dr|n$ , where  $\eta = \Omega\left(\frac{1}{\delta d}\right)$  and  $\beta = \Omega(\delta d)$ .*

Next, we use the subspace design constructed in Corollary 4.5 to obtain an explicit construction of a lossless dimension expander.

► **Theorem 5.2.** *Fix  $\varepsilon > 0$ , and choose  $\delta = \Theta(\varepsilon)$  sufficiently small and such that  $1/\delta \in \mathbb{Z}$ . Let  $d = 1/\delta^3$  and  $k = 1/\delta^2$  and assume that  $q - 1 = n$  and  $d|n$ . Then there exists an explicit construction of a  $(\frac{1-\varepsilon}{d}, (1-\varepsilon)d)$ -dimension expander with degree  $d$  over  $\mathbb{F}_q^n$ .*

We remark that this construction has degree  $d = O(1/\varepsilon^3)$ . Recalling Proposition 2.4, we know that one could hope for  $d = O(1/\varepsilon^2)$  when  $\eta = \frac{1-\varepsilon}{d}$  and  $\beta = (1-\varepsilon)d$ . Hence, the dependence of the degree on  $\varepsilon$  is just a factor of  $\varepsilon$  away from the randomized construction.

## 6 Conclusion

In this work we provide the first explicit construction of a lossless dimension expander. Our construction uses ideas from recent constructions of list-decodable rank-metric codes, which is in analogy with the approach taken by [15] in the “Boolean” world. Our approach is sufficiently general to achieve lossless expansion even in the case that the expander is “unbalanced”, i.e., when the codomain has dimension smaller than the domain.

The main open problem that remains is to achieve similar constructions over fields of smaller size. Our construction of lossless expanders requires fields of size  $q > n$ , whereas our construction of degree-proportional expanders requires fields of size  $n^\delta$  for arbitrarily small (constant)  $\delta$ . The constraints on the field size arise largely from the constructions of subspace designs that we employed. Thus, we believe that a fruitful avenue of attack on this problem would be to obtain constructions of subspace designs over smaller fields.<sup>9</sup>

The authors of [21] addressed precisely this challenge. In this work the authors do manage to construct subspace designs over all fields, but the intersection size now grows with  $\log_q n$ . If  $q = O(1)$ , then instantiating our approach with these subspace designs only guarantees expansion if the degree is logarithmic. One could also have  $q$  grow polynomially with  $n$  and achieve degree-proportional expanders, but as this does not improve over the intermediate fields approach of Section 4.1 we have not included it.

Lastly, we recall that our construction of a  $(\frac{1-\varepsilon}{d}, (1-\varepsilon)d)$ -dimension expander had degree  $d = \Theta(1/\varepsilon^3)$ , while the probabilistic argument shows  $d = O(1/\varepsilon^2)$  is sufficient. Moreover

<sup>9</sup> In [13] there is also an “extension field” construction that allows for smaller field sizes, but only guarantees the existence of “weak” subspace designs, which does not suffice for the dimension expander application.

if one is satisfied with a  $(\frac{1}{2d}, (1 - \varepsilon)d)$ -dimension expander then it is sufficient to have  $d = O(1/\varepsilon)$ . Thus, constructing lossless expanders whose degree has even better dependence on  $\varepsilon$  would also be interesting.

---

### References

---

- 1 Boaz Barak, Russell Impagliazzo, Amir Shpilka, and Avi Wigderson. Personal Communication to Dvir-Shpilka [6], 2004.
- 2 Jean Bourgain. Expanders and dimensional expansion. *Comptes Rendus Mathematique*, 347(7-8):357–362, 2009. doi:10.1016/j.crma.2009.02.009.
- 3 Jean Bourgain and Amir Yehudayoff. Expansion in  $SL_2(\mathbb{R})$  and monotone expanders. *Geometric and Functional Analysis*, 23(1):1–41, 2013. Preliminary version in the *44th Annual ACM Symposium on Theory of Computing (STOC 2012)*. This work is the full version of [2]. doi:10.1007/s00039-012-0200-9.
- 4 Zeev Dvir and Shachar Lovett. Subspace evasive sets. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 351–358. ACM, 2012.
- 5 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007. doi:10.1137/05063605X.
- 6 Zeev Dvir and Amir Shpilka. Towards dimension expanders over finite fields. *Combinatorica*, 31(3):305–320, 2011.
- 7 Zeev Dvir and Avi Wigderson. Monotone expanders: Constructions and applications. *Theory of Computing*, 6(1):291–308, 2010. doi:10.4086/toc.2010.v006a012.
- 8 Michael A Forbes and Venkatesan Guruswami. Dimension expanders via rank condensers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 800–814, 2015.
- 9 Michael A Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 163–172. ACM, 2012.
- 10 Ernst M. Gabidulin. Theory of codes with maximum rank distance. *Probl. Inform. Transm.*, 21(1):1–12, 1985. URL: <http://www.mathnet.ru/eng/ppi967>.
- 11 Ariel Gabizon. Deterministic extractors for affine sources over large fields. In *Deterministic Extraction from Weak Random Sources*, pages 33–53. Springer, 2011.
- 12 Venkatesan Guruswami. Linear-algebraic list decoding of folded reed-solomon codes. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011*, pages 77–85. IEEE Computer Society, 2011. doi:10.1109/CCC.2011.22.
- 13 Venkatesan Guruswami and Swastik Kopparty. Explicit subspace designs. *Combinatorica*, 36(2):161–185, 2016.
- 14 Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Information Theory*, 54(1):135–150, 2008. doi:10.1109/TIT.2007.911222.
- 15 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM (JACM)*, 56(4):20, 2009.
- 16 Venkatesan Guruswami and Carol Wang. Linear-algebraic list decoding for variants of Reed–Solomon codes. *IEEE Transactions on Information Theory*, 59(6):3257–3268, 2013.
- 17 Venkatesan Guruswami and Carol Wang. Evading subspaces over large fields and explicit list-decodable rank-metric codes. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014*,

- Barcelona, Spain*, volume 28 of *LIPICs*, pages 748–761. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPICs.APPROX-RANDOM.2014.748.
- 18 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.
  - 19 Venkatesan Guruswami and Chaoping Xing. Folded codes from function field towers and improved optimal rate list decoding. 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 339–350. ACM, 2012. doi:10.1145/2213977.2214009.
  - 20 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 (STOC 2013)*, pages 843–852. ACM, 2013.
  - 21 Venkatesan Guruswami, Chaoping Xing, and Chen Yuan. Subspace designs based on algebraic function fields. *Transactions of the AMS*, 2017. To appear. Available as arXiv:1704.05992.
  - 22 Aram W. Harrow. Quantum expanders from any classical cayley graph expander. *Quantum Information & Computation*, 8(8–9):715–721, 2008. URL: <http://www.rintonpress.com/journals/qiconline.html#v8n89>.
  - 23 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):333–364, 2011.
  - 24 Alexander Lubotzky and Efim Zelmanov. Dimension expanders. *Journal of Algebra*, 319(2):730–738, 2008.
  - 25 Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 285–294, 2005.
  - 26 Pavel Pudlák and Vojtěch Rödl. Pseudorandom sets and explicit constructions of Ramsey graphs. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 327–346. Dept. Math., Seconda Univ. Napoli, Caserta, 2004.
  - 27 Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2002.
  - 28 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.
  - 29 Avi Wigderson. Expanders: Old and new applications and problems. Lecture at the Institute for Pure and Applied Mathematics (IPAM), 2004.

# NP-hardness of Minimum Circuit Size Problem for OR-AND-MOD Circuits

**Shuichi Hirahara**<sup>1</sup>

Department of Computer Science, The University of Tokyo  
Tokyo, Japan  
hirahara@is.s.u-tokyo.ac.jp

**Igor C. Oliveira**

Department of Computer Science, University of Oxford  
Oxford, United Kingdom  
igor.carboni.oliveira@cs.ox.ac.uk

**Rahul Santhanam**

Department of Computer Science, University of Oxford  
Oxford, United Kingdom  
rahul.santhanam@cs.ox.ac.uk

---

## Abstract

---

The Minimum Circuit Size Problem (MCSP) asks for the size of the smallest boolean circuit that computes a given truth table. It is a prominent problem in NP that is believed to be hard, but for which no proof of NP-hardness has been found. A significant number of works have demonstrated the central role of this problem and its variations in diverse areas such as cryptography, derandomization, proof complexity, learning theory, and circuit lower bounds.

The NP-hardness of computing the minimum numbers of terms in a DNF formula consistent with a given truth table was proved by W. Masek [31] in 1979. In this work, we make the first progress in showing NP-hardness for more expressive classes of circuits, and establish an analogous result for the MCSP problem for depth-3 circuits of the form OR-AND-MOD<sub>2</sub>. Our techniques extend to an NP-hardness result for MOD<sub>m</sub> gates at the bottom layer under inputs from  $(\mathbb{Z}/m\mathbb{Z})^n$ .

**2012 ACM Subject Classification** Theory of computation → Problems, reductions and completeness

**Keywords and phrases** NP-hardness, Minimum Circuit Size Problem, depth-3 circuits

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.5

**Acknowledgements** Most of this work was done during a visit of the first author to Oxford, supported by ACT-I, JST.

## 1 Introduction

### 1.1 The Minimum Circuit Size Problem

In the Minimum Circuit Size Problem (MCSP), we are given the truth table of a Boolean function as input together with a positive integer  $s$ , and the question is whether a circuit of size at most  $s$  exists for the function represented by this truth table. It is easy to see that

---

<sup>1</sup> Supported by ACT-I, JST and JSPS KAKENHI Grant Numbers JP16J06743



MCSP is in NP: simply guess a circuit  $C$  of size at most  $s$ , and check that  $C$  computes each entry of the truth table correctly.

When solving MCSP deterministically, though, it is unclear how to avoid exhaustive search over the space of circuits of size at most  $s$ . A natural question arises: is MCSP NP-complete? The answer to this problem remains far from clear. MCSP is one of the very few natural problems in NP for which we have no strong evidence *for* or *against* NP-completeness. This is despite the fact that MCSP has long been recognized as a fundamental problem since the earliest research on complexity theory in the Soviet Union in the 1950s [41]. Indeed, it is reported in [8] that Levin delayed the publication of his NP-completeness results for Satisfiability because he was hoping to show similar results for MCSP.

The difficulty of showing MCSP to be NP-hard was explicitly addressed in the work of Kabanets and Cai [27]. Roughly speaking, suppose we have a polynomial-time reduction  $f$  from Satisfiability to MCSP that is “natural”, in the sense that the output length and output parameter depend only on the input length, and the input length is polynomially bounded in the output length – this is a property that all standard reductions have. Kabanets and Cai argued that by applying  $f$  to a trivial family of unsatisfiable formulas, we can show that the class E of problems solvable in linear exponential time requires superpolynomial circuit size. Given that the question of proving super-polynomial circuit lower bounds for explicit functions is a longstanding open question in complexity theory, this provides a significant obstacle to showing NP-hardness of MCSP via natural reductions. Note, though, that the Kabanets-Cai result does not give any evidence *against* NP-hardness of MCSP – it only suggests that NP-hardness might be hard to *establish*. There has been a long sequence of works [7, 32, 24, 23, 6] building on this result to give further evidence of the difficulty of showing NP-hardness of MCSP.

One way around the Kabanets-Cai obstacle is to study the complexity of MCSP for circuit classes for which strong circuit lower bounds are *already known*. Given a class  $\mathcal{C}$  of circuits, let  $\mathcal{C}$ -MCSP be the problem where, given a truth table and a number  $s$ , we wish to know if there is a  $\mathcal{C}$ -circuit of size  $s$  computing the given truth table.

Studying  $\mathcal{C}$ -MCSP for restricted classes  $\mathcal{C}$  of circuits is independently motivated by algorithmic applications in circuit minimization, proof complexity [30, Chapter 30], learning theory (cf. [38, 5, 18, 11]), and cryptography and lower bounds [39] (see also [10]). It was shown already in 1979 by Masek [31] that DNF-MCSP is NP-hard.<sup>2</sup> There have been different proofs of this result [15, 5], and extensions to hardness of approximation [5, 18, 29]. Nevertheless, almost four decades after Masek’s result, and despite the significant attention that the MCSP problem has received (see also [2, 3, 4, 36]), NP-hardness of  $\mathcal{C}$ -MCSP is still not known for any natural class  $\mathcal{C}$  of circuits more expressive than DNFs. To quote Allender et al. [5], “Thus an important open question is to resolve the NP-hardness of both learnability results as well as function minimization results above for classes that are stronger than DNF.”

## 1.2 Our Result

The main contribution in this work is the first NP-hardness result for  $\mathcal{C}$ -MCSP for a class  $\mathcal{C}$  of depth-3 circuits, namely the class of (unbounded fan-in)  $\text{OR} \circ \text{AND} \circ \text{MOD}_m$  circuits, where  $m$  is any integer.

---

<sup>2</sup> For a self-contained presentation of a proof of NP-hardness of DNF-MCSP, see [5].



► **Theorem 1 (Main Result).** *For every  $m \geq 2$ , given the truth table of a function  $f: \mathbb{Z}_m^n \rightarrow \{0, 1\}$ , where  $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z} = \{0, 1, \dots, m-1\}$ , it is NP-hard under polynomial-time deterministic many-one reductions to determine the size of the smallest  $\text{OR} \circ \text{AND} \circ \text{MOD}_m$  circuit  $C$  that computes  $f$ , where circuit size is measured as the top fan-in of  $C$ .<sup>3</sup>*

A few comments are in order. First, we elaborate on our computational model and complexity measure. We work with circuits which have an OR gate at the top, AND gates at the middle level, and  $\text{MOD}_m$  gates at the bottom level. We refer to such circuits as OR-AND-MOD circuits, or equivalently, DNF-MOD circuits. Such circuits operate in a natural way on inputs from  $\mathbb{Z}_m^n$ . We allow arbitrary constants from  $\mathbb{Z}_m$  to feed in to gates at the bottom layer, and insist that inputs to the middle AND layer are Boolean. In other words, a  $\text{MOD}_m$  gate outputs 1 if and only if its corresponding linear equation over  $\mathbb{Z}_m$  is satisfied, and the computations beyond the first layer are all Boolean. For  $m = 2$ , this is precisely the traditional model of DNF of Parities (cf. [14], [25], [26, Section 11.9], [1]).

The complexity measure we use is the top fan-in of the circuit, i.e., fan-in to the top OR gate. The main reason we work with this measure is naturalness and convenience. As argued in [14], top fan-in is the preferred measure for OR-AND-MOD<sub>2</sub> circuits because: (i) it measures the number of affine subspaces required to cover the 1s of the function, and thus has a nice combinatorial meaning; (ii) the number of MOD<sub>2</sub> gates feeding in to any middle layer AND gate can be assumed to be at most  $n$  without loss of generality, by using basic linear algebra, and thus the top fan-in approximates the total number of gates to within a factor of  $n$ ; and (iii) the size of a DNF is often measured by the number of terms in it, and analogously it makes sense to measure the size of a DNF of Parities by the top fan-in of the circuit.

Our results are not however critically dependent on the complexity measure we use, and admit different extensions. Indeed, we demonstrate the robustness of our techniques by adapting them to show a hardness result for computing the number of gates in OR-AND-MOD <sub>$p$</sub>  formulas, where  $p$  is prime (Appendix B). Moreover, we mention that our approach can be modified to show a hardness of approximation result (Appendix C).

The strategy for the proof of Theorem 1 is explained in Section 1.3. In short, we reduce from a variant of the well-known set cover problem [28]. The reduction consists of two stages, and it is initially presented as a randomized reduction. As one ingredient in the derandomization of our approach, we show the existence of near-optimal (seed length  $O(\log n + \log 1/\varepsilon)$ ) pseudorandom generators against  $\text{AND} \circ \text{MOD}_m$  circuits over  $\mathbb{Z}_m^n$  of arbitrary size. This result might be of independent interest, and we refer to the discussion in Section 1.3 for more details.

Before further exploring the ideas of our proof, we give some perspective on the result and the possibility of extending it to more expressive circuit classes. Using the Kabanets-Cai [27] connection between NP-hardness and circuit lower bounds mentioned before, it is not hard to show that our reduction yields a  $2^{\Omega(n)}$  lower bound on the size of DNF-MOD<sub>2</sub> circuits for a function in  $\text{E} = \text{DTIME}[2^{O(n)}]$ . Such strong exponential lower bounds for explicit functions have long been known for the model we consider (see e.g. [22], and also [13, 12]). On the other hand, extending the NP-hardness result even to slightly different classes such as depth-3 AC<sup>0</sup> circuits might be a challenge. It is still unknown if E requires depth-3 AC<sup>0</sup> circuits of size  $2^{\Omega(n)}$ , and using the Kabanets-Cai connection, natural approaches to an NP-hardness result would imply such a lower bound.

<sup>3</sup> As stated, Theorem 1 refers to the complexity of the optimization problem of finding the smallest circuit size for a given truth table, rather than the MCSP decision problem as defined. Note however that these two computational problems are easily seen to be polynomial-time equivalent to each other.

What might be more feasible though is showing NP-hardness of  $\mathfrak{C}$ -MCSP for other related classes  $\mathfrak{C}$  of circuits, and under weaker kinds of reductions, such as quasi-polynomial time reductions or non-uniform reductions. For instance, it might be possible to extend our techniques to classes such as  $\text{THR} \circ \text{AND} \circ \text{MOD}$  and depth-3  $\text{AC}^0$  circuits of small bottom fan-in. In these cases, exponential lower bounds of the form  $2^{\Omega(n)}$  have been obtained (cf. [22], [37]).

More broadly, we believe that showing NP-hardness of MCSP for more expressive classes  $\mathfrak{C}$  is an important direction in better *understanding* circuit classes from the perspective of *meta-complexity*, i.e., complexity questions about computational problems involving circuits and algorithms. There are various criteria for measuring our understanding of a circuit class, for example, (i) Can we design non-trivial satisfiability algorithms for circuits in the class? (ii) Can we unconditionally construct pseudo-random generators secure against circuits in the class? (iii) Can we learn the class using membership queries under the uniform distribution? (iv) Can we prove lower bounds against proof systems whose lines are encoded by circuits in the class? We suggest that the NP-hardness of  $\mathfrak{C}$ -MCSP is another strong indication that we understand a circuit class  $\mathfrak{C}$  well.

### 1.3 Overview of the Proof of Theorem 1

The rest of the paper is dedicated to the proof of Theorem 1, which will be completed in Section 4. Here we provide a high-level description of the reduction. For simplicity, our exposition mostly focuses on the case  $m = 2$ . After that, we explain the main difficulties in extending the result to general  $m$ , and how these are addressed in our proof.

As mentioned above, Masek [31] was the first to establish the NP-hardness of DNF minimization, and Theorem 1 can be interpreted as an extension of Masek's result to the more expressive DNF-MOD circuits. The structure of our argument follows however a *two-step* reduction introduced by Gimpel (cf. Allender et al. [5]), brought to our attention thanks to an alternative proof of Masek's result from [5]. More precisely, their work presents a new proof of the first stage of Gimpel's reduction, and provides a self-contained exposition of the entire argument.

Our NP-hardness proof for DNF-MOD circuits heavily builds on ideas of Gimpel and [5], but the extension to depth-3 requires new ideas and makes the argument much more involved. Let  $(\text{DNF} \circ \text{XOR})\text{-MCSP}$  be the computational problem described in Theorem 1 when  $m = 2$ , and let  $(\text{DNF} \circ \text{XOR})\text{-MCSP}^*$  be its natural generalization to *partial* boolean functions. In other words, an input to  $(\text{DNF} \circ \text{XOR})\text{-MCSP}^*$  encodes the truth table of a function  $f: \{0, 1\}^n \rightarrow \{0, 1, *\}$ , and we are interested in the size of the minimum  $(\text{DNF} \circ \text{XOR})$ -circuit that agrees with  $f$  on  $f^{-1}(\{0, 1\})$ . Let  $r \in \mathbb{N}$  be a large enough constant. Our proof reduces from the NP-complete problem  $r$ -Bounded Set Cover (cf. [19]): Given a set system  $\mathcal{S} \subseteq \binom{[n]}{\leq r}$  that covers  $[n]$ , determine the minimum number  $\ell$  of sets  $S_1, \dots, S_\ell \in \mathcal{S}$  such that  $\bigcup_{i=1}^{\ell} S_i = [n]$ . (We refer to Section 2.2 for a precise formulation of these computational problems.)

In a bit more detail, we present a *randomized* (2-approximate) reduction from  $r$ -Bounded Set Cover to  $(\text{DNF} \circ \text{XOR})\text{-MCSP}^*$ , and a *randomized* reduction from  $(\text{DNF} \circ \text{XOR})\text{-MCSP}^*$  to  $(\text{DNF} \circ \text{XOR})\text{-MCSP}$ . These reductions are then efficiently derandomized using an appropriate pseudorandom generator. As opposed to previous works on the NP-hardness of DNF minimization, our proof crucially explores the fact that  $r$ -Bounded Set Cover is NP-hard even to *approximate* (by roughly a  $\ln r$ -factor), a result from [17, 42] (see Theorem 5, Section 2.2).



We discuss each reduction in more detail now. Common to both of them is a convenient characterization of the sets  $C^{-1}(1) \subseteq \{0, 1\}^n$  of inputs that can be accepted by non-trivial AND  $\circ$  XOR circuits  $C$ . If  $m$  is prime, it is not hard to show that this is precisely the class of affine subspaces of  $\{0, 1\}^n$ . Consequently, for a non-trivial partial function  $f: \{0, 1\}^n \rightarrow \{0, 1, *\}$ , its corresponding  $\text{DNF}_{\text{XOR}}(f)$  complexity is exactly the minimum number  $t$  of affine subspaces  $A_1, \dots, A_t \subseteq \{0, 1\}^n$  such that  $f^{-1}(1) \subseteq \bigcup_{i=1}^t A_i$  and  $\bigcup_{i=1}^t A_i \subseteq f^{-1}(\{1, *\})$  (see Section 2.1). The analysis of our polynomial-time reductions, which will not be covered in this section, relies on this characterization in fundamental ways.

**Step 1.** *A randomized reduction from  $r$ -Bounded Set Cover to (DNF  $\circ$  XOR)-MCSP\* (Section 3.1).*

Given a set-system  $\mathcal{S} \subseteq \binom{[n]}{\leq r}$ , we define a partial boolean function  $f: \{0, 1\}^t \rightarrow \{0, 1, *\}$ , where  $t = O(r \log n)$ . This function is *probabilistically* constructed as follows. First, we associate to each  $i \in [n]$  a *random* vector  $v^i \in \{0, 1\}^t$ . For  $S \in \mathcal{S}$ , let  $v^S = \{v^i \mid i \in S\}$ . Then, we let  $f$  be 1 on each input  $v^i$ , 0 on inputs that are *not* in the linear span of  $v^S$  for every  $S \in \mathcal{S}$ , and  $*$  elsewhere.

Using this construction, we are able to show by a delicate analysis that if  $t$  is sufficiently large, the following holds with high probability: if  $\mathcal{S}$  admits a cover of size  $K$ , then  $\text{DNF}_{\text{XOR}}(f) \leq K$ ; moreover, if  $\text{DNF}_{\text{XOR}}(f) \leq K$ , then  $\mathcal{S}$  admits a cover of size  $\leq 2K$ . (We discuss the intuition for this claim in Section 3.1.) This construction and the hardness of approximation result for  $r$ -Bounded Set Cover imply that (DNF  $\circ$  XOR)-MCSP\* is NP-hard under many-one randomized reductions.

**Step 2.** *A randomized reduction from (DNF  $\circ$  XOR)-MCSP\* to (DNF  $\circ$  XOR)-MCSP (Section 3.2).*

Let  $f: \{0, 1\}^t \rightarrow \{0, 1, *\}$  be an instance of (DNF  $\circ$  XOR)-MCSP\*. We *probabilistically* construct from  $f$  a related *total* function  $g: \{0, 1\}^t \times \{0, 1\}^s \rightarrow \{0, 1\}$ , where  $r = t + 2$  and  $s = O(r + t)$ . In more detail, we encode for each  $x \in \{0, 1\}^t$  its corresponding value  $f(x) \in \{0, 1, *\}$  as a *boolean function*  $g_x$  on a hypercube  $\{0, 1\}^s$ . For an input  $x$  such that  $f(x) \in \{0, 1\}$ , we let  $g(x0^s) = g_x(0^s) = f(x)$ , where  $g_x(\cdot) = 0$  elsewhere. On the other hand, if  $f(x) = *$ , we pick a *random* linear subspace  $L_x \subseteq \{0, 1\}^s$  of dimension  $r$ , and we encode  $f(x)$  as the characteristic function of  $L_x$ .

Again, a careful argument allows us to establish the following connection between the partial function  $f$  and the total function  $g$ : with high probability over the choice of the random linear subspaces  $(L_x)_{x \in f^{-1}(*)}$ ,  $\text{DNF}_{\text{XOR}}(g) = \text{DNF}_{\text{XOR}}(f) + |f^{-1}(*)|$ . (We discuss the intuition for this claim in Section 3.2.) Consequently, it follows from this and the previous reduction that (DNF  $\circ$  XOR)-MCSP is NP-hard under many-one randomized reductions.

**Step 3.** *Efficient derandomization of the reductions (Section 4.1).*

It is possible to prove that the first reduction is always correct provided that the collection of random vectors  $v^i$  is *nice* with respect to the set-system  $\mathcal{S}$  (Definition 12). Similarly, we can prove that the second reduction is correct whenever the collection  $(L_x)_{x \in f^{-1}(*)}$  of linear subspaces is *scattered* (Definition 18). It turns out that both conditions can be checked in polynomial time. This implies that the previously discussed reductions are in fact *zero-error* reductions. Consequently, if we can efficiently construct nice vectors and scattered families of linear subspaces, the reductions can be made deterministic.

In order to achieve this, we use in both cases a subtle derandomization argument that relies on (polynomial-time computable)  $\varepsilon$ -biased distributions [33]. Recall that such distributions

can fool arbitrary linear tests. By a more careful analysis, it is also known that they fool  $\text{AND} \circ \text{XOR}$  circuits. We do *not* describe an  $\text{AND} \circ \text{XOR}$  circuit to check if a collection of vectors is nice, or to check if a collection of linear subspaces is scattered. Still, we are able to show that if  $\varepsilon < 2^{-s}$  then some scattered collection of linear subspaces is encoded by a string in the support of an  $\varepsilon$ -biased distribution, and that the same holds with respect to a nice collection of vectors if  $\varepsilon < 2^{-t}$ . In particular, trying all possible seeds of an  $\varepsilon$ -biased generator produces the combinatorial and algebraic objects that are sufficient to derandomize our reductions. (We refer to Section 4.1 for more details.)

Overall, combining the (derandomized) reductions and using the hardness of approximation result for  $r$ -Bounded Set Cover mentioned above, it follows that  $(\text{DNF} \circ \text{XOR})\text{-MCSP}$  is NP-hard under many-one deterministic polynomial-time reductions.

**The argument for arbitrary  $m \geq 2$ .** Let  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$  and  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$  be the corresponding computational problems with respect to an arbitrary  $m \geq 2$ . (Recall that the input boolean functions in this case are defined over  $\mathbb{Z}_m^n$ .) As we explain next, additional difficulties are present for general  $m$ .

An immediate challenge is that it is no longer clear if the analogous characterization (via affine subspaces) of the class of subsets of  $\mathbb{Z}_m^n$  accepted by non-trivial  $\text{AND} \circ \text{MOD}_m$  circuits holds, and this is crucially exploited when  $m = 2$ . The main issue is that, while in the latter case the result can be established by elementary techniques using that  $\mathbb{Z}_2^n$  is a *vector space* over  $\mathbb{Z}_2$ , for an arbitrary  $m$  the underlying structure might be just a *module*. Without a *basis*, the result is less clear.

Nevertheless, it is possible to prove that the analogous result for  $\text{AND} \circ \text{MOD}_m$  circuits holds (cf. Lemma 2). The alternative and more general argument relies on a property of double orthogonal complements in  $\mathbb{Z}_m^n$  (Appendix A), and we refer to Section 2.1 for more details. Armed with this characterization, the reductions discussed before can be adapted to arbitrary  $m$ . Finding the right generalization of each definition requires some work, but after that, the *randomized* reductions for  $m = 2$  and arbitrary  $m \geq 2$  can be presented in a unified and transparent way.

In order to conclude the proof of Theorem 1, we need to derandomize the new reductions. For  $m = 2$ , the argument was based on an efficient construction of  $\varepsilon$ -biased distributions supported over  $\{0, 1\}^n$ , and the fact that such distributions are also able to fool  $\text{AND} \circ \text{XOR}$  circuits over  $\{0, 1\}^n$ . Without going into further details, we mention that for arbitrary  $m$  it is sufficient to use a pseudorandom generator that fools  $\text{AND} \circ \text{MOD}_m$  circuits over  $\mathbb{Z}_m^n$ . However, a generator with *near-optimal* dependency on  $n$  and  $\varepsilon$  is needed if we are hoping to obtain a polynomial-time reduction. We were not able to find such a result in the literature.<sup>4</sup>

We show in Section 4.2 that, for every  $m \geq 2$ , there is an efficient pseudorandom generator  $G_n: \{0, 1\}^{O(\log n + \log 1/\varepsilon)} \rightarrow \mathbb{Z}_m^n$  that  $\varepsilon$ -fools  $\text{AND} \circ \text{MOD}_m$  circuits of arbitrary size. Our construction relies on the efficient  $\varepsilon$ -biased generators for  $\mathbb{Z}_m^n$  from [9], together with a proof of the following result: If  $G$  is an  $\varepsilon$ -biased generator against  $\mathbb{Z}_m^n$ , then  $G$  ( $m\varepsilon$ )-fools  $\text{AND} \circ \text{MOD}_m$  circuits. Again, we cannot rely on an adaptation of the similar claim for  $m = 2$ , which requires a basis. Our proof proceeds instead by a careful analysis of certain exponential sums encoding the behaviour of the circuit, and that can be used to connect the distinguishing probability to the guarantees offered by the  $\varepsilon$ -biased generator. We refer to Section 4.2 for more details.

<sup>4</sup> Existing generators seem to generate *bits* only, or are restricted to prime modulus, or can handle larger classes of functions but are not efficient enough for our purposes. We refer to [21] and the references therein for related results.

## 2 Preliminaries

**Notation.** For an integer  $n \geq 1$ , let  $[n]$  denote  $\{1, \dots, n\}$ .

**Some notions from group theory.** Let  $m \geq 2$  be a constant. Let  $\mathbb{Z}_m := \mathbb{Z}/m\mathbb{Z}$  denote the integers modulo  $m$ , where all operations on elements in  $\mathbb{Z}_m = \langle +, \{0, 1, \dots, m-1\} \rangle$  are taken mod  $m$ . For any integer  $t \geq 1$ , we regard  $\mathbb{Z}_m^t$  as an additive group with component-wise addition. A non-empty subset  $H \subseteq \mathbb{Z}_m^t$  is called a *linear subspace* if  $H$  is a subgroup, that is,  $0 \in H$  and  $x + y \in H$  for any  $x, y \in H$ . A subset  $A \subseteq \mathbb{Z}_m^t$  is called an *affine subspace* if  $A$  is a coset, that is, there exist  $a \in \mathbb{Z}_m^t$  and a linear subspace  $H \subseteq \mathbb{Z}_m^t$  such that  $A = H + a := \{h + a \mid h \in H\}$ .

We stress that  $\mathbb{Z}_m^t$  gives rise to a *module* and not to a *vector space* when  $m$  is a composite number; however, we borrow some standard notation; for example, for a scalar  $c \in \mathbb{Z}_m$  and a “vector”  $v \in \mathbb{Z}_m^t$ , let  $cv$  denote the scalar multiplication. Let  $\langle x, y \rangle := \sum_{i=1}^t x_i y_i$  ( $\in \mathbb{Z}_m$ ) for any  $x, y \in \mathbb{Z}_m^t$  and  $t \in \mathbb{N}$ .

### 2.1 Circuit Size Measure and Its Characterization

For any integer  $m \geq 2$ , an  $\text{OR} \circ \text{AND} \circ \text{MOD}_m$  ( $= \text{DNF} \circ \text{MOD}_m$ ) circuit is a DNF formula whose terms are  $\text{AND} \circ \text{MOD}_m$  circuits. Here, the  $\text{MOD}_m$  gate is a Boolean function such that  $\text{MOD}_m(x) = 1$  if and only if  $\sum_{i=1}^n x_i \bmod m = 0$  on input  $x \in \{0, 1\}^n$ . We allow multiple input wires and access to constant input bits in the circuit. Note that this allows for more general equations to be computed by a bottom-layer modular gate.

The size of a circuit is usually defined as the number of gates. However, for us it is important to define the size of a  $\text{DNF} \circ \text{MOD}_m$  circuit as the top fan-in of the circuit, or equivalently, its number of  $\text{AND} \circ \text{MOD}_m$  terms. (Note that the same size measure was used in [14] in the case  $m = 2$ .) For a Boolean function  $f: \{0, 1\}^t \rightarrow \{0, 1\}$ , define  $\text{DNF}_{\text{MOD}_m}(f)$  as the minimum number of terms of a  $\text{DNF} \circ \text{MOD}_m$  circuit computing  $f$ , i.e., the fan-in of its OR gate.

In order to present our results in a unified way for any integer  $m \geq 2$ , we extend the input  $\{0, 1\}^t$  of a  $\text{DNF} \circ \text{MOD}_m$  circuit to the larger domain  $\mathbb{Z}_m^t$  in a natural way: that is, we regard the bottom  $\text{MOD}_m$  gate as a function  $\text{MOD}_m: \mathbb{Z}_m^* \rightarrow \{0, 1\}$  that outputs 1 if and only if the sum of its input elements is congruent to 0 mod  $m$ . Again, more general equations can be obtained using multiple input wires and access to constants in  $\mathbb{Z}_m$ .

An  $\text{AND} \circ \text{MOD}_m$  circuit  $C$  *accepts* the set  $X \subseteq \mathbb{Z}_m^t$  if for any  $x \in \mathbb{Z}_m^t$ ,  $x \in X$  if and only if  $C$  outputs 1 on  $x$ . There is a nice combinatorial characterization of the set of inputs that such circuits can accept.

► **Lemma 2** (Characterization of the power of  $\text{AND} \circ \text{MOD}_m$  circuits). *Let  $X \subseteq \mathbb{Z}_m^t$  be a nonempty set. Then, an  $\text{AND} \circ \text{MOD}_m$  circuit accepts  $X$  if and only if  $X$  is an affine subspace of  $\mathbb{Z}_m^t$ .*

This is a standard fact when  $m$  is a prime (cf. [14] for  $m = 2$ ), in which case  $\mathbb{Z}_m^t$  is a vector space. However, the same characterization holds when  $m \geq 2$  is an arbitrary composite number, as established below. The proof relies on the following fact about orthogonal complements in the more general context of modules.

► **Fact 3** (Double orthogonal complement). *Let  $H \subseteq \mathbb{Z}_m^t$  be a linear subspace, and let  $H^\perp := \{x \in \mathbb{Z}_m^t \mid \sum_{i=1}^t x_i y_i = 0 \text{ for any } y \in H\}$  be its orthogonal complement. Then,  $(H^\perp)^\perp = H$ .*

For completeness, we include a proof of this result in Appendix A. Assuming Fact 3, we proceed to a proof of Lemma 2.

**Proof of Lemma 2.** Let  $x := (x_1, \dots, x_t) \in \mathbb{Z}_m^t$  denote the input to the circuit.

Suppose that an  $\text{AND} \circ \text{MOD}_m$  circuit  $\bigwedge_{k=1}^K C_k$  accepts  $X$ , where each  $C_k$  is a  $\text{MOD}_m$  gate. Each  $\text{MOD}_m$  gate  $C_k$  in the circuit defines a linear equation over  $(x_1, \dots, x_t)$ . That is, there are coefficients  $a_k^1, \dots, a_k^t \in \mathbb{Z}_m$  and an element  $b_k \in \mathbb{Z}_m$  such that  $\sum_{i=1}^t a_k^i x_i = b_k$  if and only if  $C_k$  accepts the input  $x$ . Therefore, the circuit  $\bigwedge_{k=1}^K C_k$  accepts the intersection of such linear equations over  $\mathbb{Z}_m$ . Specifically, for a matrix  $A := (a_k^i)_{k \in [K], i \in [t]}$  and a vector  $b := (b_k)_{k \in [K]}$ , the circuit accepts all inputs  $x \in \mathbb{Z}_m^t$  such that  $Ax = b$ ; namely,  $X = \{x \in \mathbb{Z}_m^t \mid Ax = b\}$ . Since  $X$  is nonempty, we can take some element  $x_0 \in X$ . Now, we can rewrite  $X$  as

$$X = \{x \in \mathbb{Z}_m^t \mid A(x - x_0) = 0\} = \{y \in \mathbb{Z}_m^t \mid Ay = 0\} + x_0,$$

which is an affine subspace of  $\mathbb{Z}_m^t$ .

For the converse direction, we use the notion of orthogonal complement. Suppose that  $X \subseteq \mathbb{Z}_m^t$  is an affine subspace. By definition, we can decompose  $X$  into a linear subspace  $H \subseteq \mathbb{Z}_m^t$  and a shift  $a \in \mathbb{Z}_m^t$  so that  $X = H + a$ .

We first claim that  $H$  can be accepted by some  $\text{AND} \circ \text{MOD}_m$  circuit. To prove this, it is sufficient to show the existence of some matrix  $A \in \mathbb{Z}_m^{K \times t}$  such that  $H = \{x \in \mathbb{Z}_m^t \mid Ax = 0\}$ . Since  $H$  is a linear subspace, by Fact 3, for any  $x \in \mathbb{Z}_m^t$ ,

$$x \in H \quad \text{if and only if} \quad \sum_{i=1}^t x_i \cdot y_i = 0 \quad \text{for every } y \in H^\perp.$$

That is, we can define a matrix  $A \in \mathbb{Z}_m^{|H^\perp| \times t}$  as  $(y_i)_{y \in H^\perp, i \in [t]}$ . (In other words, for each  $y \in H^\perp$ , we add a  $\text{MOD}_m$  gate that checks if  $\sum_{i=1}^t x_i \cdot y_i = 0$ , where each coefficient  $y_i$  is simulated using multiple input wires.)

To accept  $X$ , we just need to shift  $H$  by  $a$ . Indeed, for a vector  $b := Aa$ , we have  $X = H + a = \{x \in \mathbb{Z}_m^t \mid Ax = b\}$ ; thus we can construct an  $\text{AND} \circ \text{MOD}_m$  circuit accepting  $X$  by simulating the condition  $Ax = b$ .  $\blacktriangleleft$

As a consequence of Lemma 2, for a function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1\}$ , the minimum size of a  $\text{DNF} \circ \text{MOD}_m$  circuit computing  $f$  equals the minimum number  $S$  of affine subspaces  $T_1, \dots, T_S \subseteq \mathbb{Z}_m^t$  such that  $\bigcup_{i=1}^S T_i = f^{-1}(1)$ .

## 2.2 Computational Problems

The starting point of our NP-hardness results is the set cover problem on instances where each set has size at most  $r$ .

**► Definition 4** (*r*-Bounded Set Cover Problem). For an integer  $r \in \mathbb{N}$ , the *r*-Bounded Set Cover Problem is defined as follows:

- *Input.* An integer  $n \in \mathbb{N}$  and a collection  $\mathcal{S} \subseteq 2^{[n]}$  of nonempty subsets of the universe  $[n]$  such that  $|S| \leq r$  for each  $S \in \mathcal{S}$ , and  $\bigcup_{S \in \mathcal{S}} S = [n]$ .
- *Output.* The minimum number  $\ell$  of subsets  $S_1, \dots, S_\ell \in \mathcal{S}$  such that  $\bigcup_{i=1}^\ell S_i = [n]$ .

For this problem, a tight inapproximability result based on NP-hardness is known.

► **Theorem 5** (Feige [17], Trevisan [42]). *Let  $r$  be a sufficiently large constant. It is NP-hard (under polynomial-time many-one reductions) to approximate the solution of the  $r$ -bounded set cover problem within a factor of  $\ln r - O(\ln \ln r)$ . That is, for any language  $L \in \text{NP}$ , there exists a polynomial-time machine that, on input  $x$ , outputs a threshold  $\theta$  and an instance  $\mathcal{S}$  of the  $r$ -bounded set cover problem such that if  $x \in L$  then  $\mathcal{S}$  has a cover of size at most  $\theta$ , and if  $x \notin L$  then  $\mathcal{S}$  does not have a cover of size at most  $\theta \cdot (\ln r - O(\ln \ln r))$ .*

We stress that the *inapproximability* result is essential for us; we will present a reduction from a 2-factor approximation of the  $r$ -bounded set cover problem to the minimum  $\text{DNF} \circ \text{MOD}_m$  circuit minimization problem.

► **Definition 6** (Minimum Circuit Size Problem for  $\text{DNF} \circ \text{MOD}_m$ ). For an integer  $m \geq 2$ , the Minimum Circuit Size Problem for  $\text{DNF} \circ \text{MOD}_m$ , abbreviated as  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$ , is defined as follows:

- *Input.* A Boolean function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1\}$ , represented as a truth table of length  $m^t$ .
- *Output.*  $\text{DNF}_{\text{MOD}_m}(f)$ .

While our final theorem confirms that  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$  is NP-hard, we will first prove NP-hardness of the circuit minimization problem on instances of a partial function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$ . That is, we regard any input  $x \in f^{-1}(*)$  as “undefined.” For a partial function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$ , we say that a circuit  $C$  computes  $f$  if  $C(x) = f(x)$  for any  $x \in f^{-1}(\{0, 1\})$ . We extend the definition of  $\text{DNF}_{\text{MOD}_m}(f)$  to the size of the minimum  $\text{DNF} \circ \text{MOD}_m$  circuit computing the partial function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$ . The following problem is concerned with the circuit size of partial functions, and we distinguish it from the problem above by adding a superscript  $*$ .

► **Definition 7** (Minimum Circuit Size Problem for Partial Functions). For an integer  $m \geq 2$ , the Minimum Circuit Size Problem $*$  for  $\text{DNF} \circ \text{MOD}_m$ , abbreviated as  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$ , is defined as follows:

- *Input.* A Boolean function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$ , represented as a string of length  $m^t$  over the alphabet  $\{0, 1, *\}$ .
- *Output.*  $\text{DNF}_{\text{MOD}_m}(f)$ .

### 3 Hardness of $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$ Under Randomized Reductions

#### 3.1 Reduction from $r$ -Bounded Set Cover to $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$

This subsection is devoted to proving the following theorem.

► **Theorem 8.**  *$(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$  is NP-hard under (zero-error) randomized polynomial-time many-one reductions.*

Let  $r$  be a large enough constant so that the approximation factor of  $\ln r - O(\ln \ln r)$  in Theorem 5 is larger than 2. We present a reduction from a 2-factor approximation of the  $r$ -bounded set cover problem to  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$ .

Let us prepare some notation. Let  $\mathcal{S}$  be an instance of the  $r$ -bounded set cover problem over the universe  $[n]$  (in particular,  $\bigcup_{S \in \mathcal{S}} S = [n]$ ). Let  $t \in \mathbb{N}$  be a parameter chosen later. For each  $i \in [n]$ , pick  $v^i \in_R \mathbb{Z}_m^t$  independently and uniformly at random. For any  $S \subseteq [n]$ , let  $v^S$  denote  $\{v^i \mid i \in S\}$ . Let  $\text{span}(v^S) := \{\sum_{i \in S} c_i \cdot v^i \mid c_i \in \mathbb{Z}_m \text{ for any } i \in S\}$  denote the linear span of  $v^S$ . (Note that  $\text{span}(v^S)$  is a linear subspace of  $\mathbb{Z}_m^t$  whenever  $S \neq \emptyset$ .) In our reduction, an element  $i \in [n]$  is mapped to a random point  $v^i$  of  $\mathbb{Z}_m^t$ , and a set  $S \in \mathcal{S}$  corresponds to a linear subspace  $\text{span}(v^S)$ .

## 5:10 NP-hardness of MCSP for OR-AND-MOD Circuits

For any set cover instance  $\mathcal{S}$ , we define a function  $f : \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$  as

$$f(x) := \begin{cases} 1 & (\text{if } x = v^i \text{ for some } i \in [n]) \\ 0 & (\text{if } x \notin \bigcup_{S \in \mathcal{S}} \text{span}(v^S)) \\ * & (\text{otherwise}) \end{cases}$$

for any  $x \in \mathbb{Z}_m^t$ . The truth table of  $f$  is the output of our reduction.

It is not hard to see that  $\text{DNF}_{\text{MOD}_m}(f)$  is at most the minimum set cover size for  $\mathcal{S}$  (Claim 9 below). Of course, the difficulty is in proving a circuit lower bound for  $f$  (Claim 10 below).

The idea is as follows: For simplicity of the exposition, let us focus on the case of  $m = 2$ , and moreover let us first consider the case of a  $\text{DNF} \circ \text{MOD}_2$  circuit  $C$  for  $f$  that accepts a union of *linear* subspaces (instead of affine subspaces). More precisely, let  $C^{-1}(1)$  be a union of linear subspaces  $\{T_k\}_{k \in [K]}$ . Then  $T_k$  is a subset of  $C^{-1}(1) \subseteq f^{-1}(\{1, *\}) = \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ ; furthermore, each  $\text{span}(v^S)$  is a random linear subspace of small dimension  $r$ ; therefore, it is possible to show that, with high probability, the set  $\{i \in [n] \mid v^i \in T_k\}$  of points covered by  $T_k$  is contained in some legal set  $S \in \mathcal{S}$  of the set cover instance; hence the circuit size  $K$  is at least the minimum set cover size.

In the case that a circuit  $C$  accepts the union of *affine* subspaces, it is no longer true that, for any affine subspace  $T$  such that  $T \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ , the set  $\{i \in [n] \mid v^i \in T\}$  is covered by some legal set  $S \in \mathcal{S}$ ; indeed, for any two points  $v^i$  and  $v^j$ , the set  $\{v^i, v^j\} (= v^i \oplus \{0, v^i \oplus v^j\})$  is an affine subspace of  $\mathbb{Z}_2^t$ , whereas  $\{i, j\}$  is not necessarily legal in the set cover instance  $\mathcal{S}$ . Nonetheless, we can still prove that, with high probability, the set  $\{i \in [n] \mid v^i \in T\}$  is covered by two legal sets  $S_1, S_2 \in \mathcal{S}$ . As a consequence, the minimum number of affine subspaces needed to cover  $v^1, \dots, v^n$  gives us a 2-factor approximation of the minimum set cover size for  $\mathcal{S}$ . By Theorem 5, it follows that  $(\text{DNF} \circ \text{XOR})\text{-MCSP}^*$  is NP-hard under randomized reductions. Details follow.

► **Claim 9** (Easy part). *Suppose that  $\mathcal{S}$  has a set cover of size  $K$ . Then  $\text{DNF}_{\text{MOD}_m}(f) \leq K$ .*

**Proof.** Let  $\mathcal{C} \subseteq \mathcal{S}$  be a set cover of size  $K$ . For each  $S \in \mathcal{C}$ , by Lemma 2, there exists an  $\text{AND} \circ \text{MOD}_m$  circuit  $C_S$  such that  $C_S$  accepts  $\text{span}(v^S)$ . Define a  $\text{DNF} \circ \text{MOD}_m$  circuit  $C := \bigvee_{S \in \mathcal{C}} C_S$ . It is easy to see that  $C$  computes  $f$ . ◀

Conversely, we prove the following:

► **Claim 10** (Hard part). *For some parameter  $t$  such that  $m^t = (nm)^{O(r)}$ , the following holds with probability at least  $\frac{1}{2}$  (over the choice of  $(v^i)_{i \in [n]}$ ):*

*Let  $K := \text{DNF}_{\text{MOD}_m}(f)$ . Then  $\mathcal{S}$  has a set cover of size  $2K$ .*

The two claims above imply that  $2\text{DNF}_{\text{MOD}_m}(f)$  is a 2-factor approximation for the set cover problem: indeed, let  $s$  be the minimum set cover size for  $\mathcal{S}$ ; then we have  $s \leq 2\text{DNF}_{\text{MOD}_m}(f) \leq 2s$ . It thus remains to prove Claim 10.

To prove Claim 10, let us clarify the desired condition that random objects  $(v^i)_{i \in [n]}$  should satisfy. For any  $I \subseteq [n]$ , define the *affine span* of  $v^I$  as

$$\text{affine-span}(v^I) := \left\{ \sum_{i \in I} c_i v^i \mid c_i \in \mathbb{Z}_m \text{ for } i \in I \text{ and } \sum_{i \in I} c_i = 1 \right\}.$$

The important property of the affine span is that, if an affine subspace  $A$  covers the set  $v^I$  of points in  $I \subseteq [n]$ , then its affine span must also be covered by  $A$ .



► **Claim 11** (Property of the affine span). *For any affine subspace  $A$  of  $\mathbb{Z}_m^t$  and any  $I \subseteq [n]$ , if  $v^I \subseteq A$  then  $\text{affine-span}(v^I) \subseteq A$ .*

**Proof.** Let us write  $A = H + a$  for some linear space  $H \subseteq \mathbb{Z}_m^t$  and vector  $a \in \mathbb{Z}_m^t$ . Since  $v^i \in v^I \subseteq A$  for each  $i \in I$ , there exists some vector  $h^i \in H$  such that  $v^i = h^i + a$ . Take any coefficients  $(c_i)_{i \in I}$  such that  $c_i \in \mathbb{Z}_m$  and  $\sum_{i \in I} c_i = 1$ . Then,

$$\sum_{i \in I} c_i v^i = \sum_{i \in I} c_i (h^i + a) = \sum_{i \in I} c_i h^i + a \in H + a. \quad \blacktriangleleft$$

By Lemma 2, the circuit size of  $f$  equals the minimum number of affine subspaces  $A_1, \dots, A_K \subseteq f^{-1}(\{1, *\})$  such that  $\bigcup_{i=1}^K A_i \supseteq f^{-1}(1)$ . Intuitively, we would like to require that, if the set  $v^I$  ( $\subseteq f^{-1}(1)$ ) of points is covered by some affine subspace  $A \subseteq f^{-1}(\{1, *\})$ , then there exist two legal sets  $S_1, S_2$  of the set cover instance  $\mathcal{S}$  such that  $I \subseteq S_1 \cup S_2$ . In fact, one of these sets can be taken as a singleton:

► **Definition 12.** We say that  $(v^i)_{i \in [n]}$  is *nice* (with respect to  $\mathcal{S}$ ) if, for any  $I \subseteq [n]$ ,

$$\text{affine-span}(v^I) \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S) \implies I \subseteq S_I \cup \{i_I\} \quad (1)$$

for some  $S_I \in \mathcal{S}$  and  $i_I \in [n]$ .

We will prove that  $(v^i)_{i \in [n]}$  is nice with probability at least  $\frac{1}{2}$ , and that for any nice  $(v^i)_{i \in [n]}$ , the minimum size of  $\text{DNF} \circ \text{MOD}_m$  is a 2-factor approximation of the minimum set cover size. We prove the latter first:

► **Claim 13.** *Let  $(v^i)_{i \in [n]}$  be nice, and  $K := \text{DNF}_{\text{MOD}_m}(f)$ . Then  $\mathcal{S}$  has a set cover of size  $2K$ .*

**Proof.** Let  $C = \bigvee_{k=1}^K C_k$  be a  $\text{DNF} \circ \text{MOD}_m$  circuit computing  $f$ , where each  $C_k \in \text{AND} \circ \text{MOD}_m$  is nontrivial. By Lemma 2,  $C_k^{-1}(1)$  is an affine subspace of  $\mathbb{Z}_m^t$ . For each  $C_k$ , we will choose 2 sets from  $\mathcal{S}$  so that the union of all these sets cover the universe  $[n]$ .

Fix any  $C_k$  and let  $I_k := \{i \in [n] \mid C_k(v^i) = 1\}$  be the set of all points covered by  $C_k$ . Since  $C_k^{-1}(1)$  is an affine subspace of  $\mathbb{Z}_m^t$  and  $v^{I_k} \subseteq C_k^{-1}(1)$ , we have  $\text{affine-span}(v^{I_k}) \subseteq C_k^{-1}(1)$  by Claim 11. Since the circuit  $C$  computes  $f$ ,  $C_k^{-1}(1) \subseteq C^{-1}(1) \subseteq f^{-1}(\{1, *\}) = \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ . Thus we have  $\text{affine-span}(v^{I_k}) \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ , which means that the hypothesis of niceness (1) is satisfied; hence there exist some subset  $S_{k1} \in \mathcal{S}$  and some element  $i_k \in [n]$  such that  $I_k \subseteq S_{k1} \cup \{i_k\}$ . Take any set  $S_{k2} \in \mathcal{S}$  such that  $i_k \in S_{k2}$  (such a set  $S_{k2}$  must exist because we assumed  $\bigcup_{S \in \mathcal{S}} S = [n]$ ). Then  $I_k \subseteq S_{k1} \cup S_{k2}$ .

Now we claim that  $\bigcup_{k=1}^K S_{k1} \cup S_{k2} = [n]$  (and hence the set cover instance  $\mathcal{S}$  has a cover of size  $2K$ ). Indeed, for any  $i \in [n]$ , we have  $f(v^i) = 1$  and hence  $C(v^i) = 1$ , which means that there exists some subcircuit  $C_k$  such that  $C_k(v^i) = 1$ . Thus  $i \in I_k \subseteq S_{k1} \cup S_{k2}$  for some  $k \in [K]$ .  $\blacktriangleleft$

It remains to show that a random choice of  $(v_i)_{i \in [n]}$  is nice with high probability:

► **Claim 14.** *For each  $i \in [n]$ , pick  $v^i \in_R \mathbb{Z}_m^t$  uniformly at random and independently. If  $t \geq r + ((r+2) \log n + \log |\mathcal{S}| + 1) / \log m$ , then  $(v^i)_{i \in [n]}$  is nice with probability at least  $\frac{1}{2}$ .*

To prove Claim 14, we will use a union bound over all relevant subsets  $I \subseteq [n]$ ; however, the definition of niceness (1) appears to suggest that we need to take a union bound over exponentially many subsets  $I$ . The next claim shows that this is in fact *not* the case.

► **Claim 15** (Characterization of niceness).  $(v^i)_{i \in [n]}$  is not nice (with respect to  $\mathcal{S}$ ) if and only if there exists some subset  $I \subseteq [n]$  such that all the following conditions hold:

1.  $|I| \leq r + 2$ ,
2.  $I \not\subseteq S \cup \{i\}$  for any  $S \in \mathcal{S}$  and  $i \in [n]$ , and
3.  $\text{affine-span}(v^I) \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ .

In particular, there are at most  $n^{r+2}$  subsets  $I \subseteq [n]$  over which we need to take a union bound.

**Proof.** By the definition of niceness,  $(v^i)_{i \in [n]}$  is not nice if and only if there exists some subset  $I \subseteq [n]$  such that  $\text{affine-span}(v^I) \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$  whereas  $I \not\subseteq S \cup \{i\}$  for any  $S \in \mathcal{S}$  and  $i \in [n]$ . Therefore, it is clear that the three conditions imply that  $(v^i)_{i \in [n]}$  is not nice; we prove below the converse direction (the “only if” part of Claim 15).

A crucial observation is that, for any subset  $I \subseteq [n]$  of size at least  $r + 2$ , the second condition always holds: Indeed, recall that  $\mathcal{S}$  is an instance of the  $r$ -bounded set cover instance; that is,  $|S| \leq r$  for any  $S \in \mathcal{S}$ . Hence, for any  $S \in \mathcal{S}$  and  $i \in [n]$ , we have  $|S \cup \{i\}| \leq r + 1$ ; thus  $I$  cannot be a subset of  $S \cup \{i\}$  simply because  $|I| \geq r + 2$ .

Now suppose that there exists some subset  $I \subseteq [n]$  satisfying the second and third conditions, but not the first one, that is,  $|I| > r + 2$ . Take any subset  $I' \subseteq I$  such that  $|I'| = r + 2$ . We claim that  $I'$  satisfies all three conditions: The first condition ( $|I'| \leq r + 2$ ) is obvious. The second condition holds because of the observation above. To see the third condition, by assumption, we have  $\text{affine-span}(v^I) \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ ; hence, we also have  $\text{affine-span}(v^{I'}) \subseteq \text{affine-span}(v^I) \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ . ◀

Now let us proceed to a proof of Claim 14.

**Proof of Claim 14.** We will bound the probability that a random  $(v^i)_{i \in [n]}$  is not nice, by using the union bound over all the subsets  $I \subseteq [n]$  such that the first and second conditions in Claim 15 hold. To this end, fix any subset  $I \subseteq [n]$  such that  $|I| \leq r + 2$  and  $I \not\subseteq S \cup \{i\}$  for any  $S \in \mathcal{S}$  and  $i \in [n]$  (in particular,  $I$  is not empty). We would like to bound the probability that the affine subspace of  $v^I$  is a subset of  $\bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ .

Take an arbitrary (e.g. the smallest) element  $i_0 \in I$ . Define coefficients  $(c_i)_{i \in I}$  as follows:  $c_i := 1 \in \mathbb{Z}_m$  for any  $i \in I \setminus i_0$  and  $c_{i_0} := (2 - |I|) \bmod m \in \mathbb{Z}_m$ . By this definition, we have  $\sum_{i \in I} c_i = 1$ ; hence,  $\sum_{i \in I} c_i v^i \in \text{affine-span}(v^I)$ . Therefore,

$$\begin{aligned} \Pr_{v^1, \dots, v^n} \left[ \text{affine-span}(v^I) \subseteq \bigcup_{S \in \mathcal{S}} \text{span}(v^S) \right] &\leq \Pr \left[ \sum_{i \in I} c_i v^i \in \bigcup_{S \in \mathcal{S}} \text{span}(v^S) \right] \\ &\leq \sum_{S \in \mathcal{S}} \Pr \left[ \sum_{i \in I} c_i v^i \in \text{span}(v^S) \right]. \end{aligned}$$

By the assumption on  $I$ , we have  $I \not\subseteq S \cup \{i_0\}$  for any  $S \in \mathcal{S}$ ; that is, there exists some index  $j_S \in I \setminus \{i_0\} \setminus S$ . Note that  $c_{j_S} = 1$  because  $j_S \in I \setminus \{i_0\}$ . Therefore, the last probability is



$$\begin{aligned}
\sum_{S \in \mathcal{S}} \Pr \left[ \sum_{i \in I} c_i v^i \in \text{span}(v^S) \right] &= \sum_{S \in \mathcal{S}} \Pr \left[ v^{j_S} \in \text{span}(v^S) - \sum_{i \in I \setminus \{j_S\}} c_i v^i \right] \\
&= \sum_{S \in \mathcal{S}} \Pr \left[ v^{j_S} = \sum_{i \in S} d_i v^i - \sum_{i \in I \setminus \{j_S\}} c_i v^i \text{ for some } (d_i)_{i \in S} \right] \\
&= \sum_{S \in \mathcal{S}} \sum_{(d_i)_{i \in S}} \Pr \left[ v^{j_S} = \sum_{i \in S} d_i v^i - \sum_{i \in I \setminus \{j_S\}} c_i v^i \right] \\
&\leq |\mathcal{S}| \cdot m^r \cdot m^{-t},
\end{aligned}$$

where the last inequality holds because the random vector  $v^{j_S}$  does not appear in the right summations.

Finally, by taking the union bound over all  $I$  such that  $|I| \leq r + 2$  (and  $I \not\subseteq S \cup \{i\}$  for any  $S \in \mathcal{S}$  and  $i \in [n]$ ), the probability that  $(v^i)_{i \in [n]}$  is not nice is bounded from above by  $n^{r+2} \cdot |\mathcal{S}| \cdot m^{r-t} \leq \frac{1}{2}$ . ◀

Given these claims above, it is immediate to complete the whole proof.

**Proof of Claim 10.** We may assume without loss of generality that  $|\mathcal{S}| \leq n^r$  since  $\mathcal{S}$  is an instance of the  $r$ -bounded set cover problem. We set  $t \in \mathbb{N}$  to be the smallest integer such that  $t \geq r + ((r + 2) \log n + \log |\mathcal{S}| + 1) / \log m$ ; then  $t = O(r \log(nm) / \log m)$ . (Here the  $O$  notation hides only a universal constant.) Combining Claims 13 and 14, we immediately obtain Claim 10. ◀

**Proof of Theorem 8.** The encoding of the function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$  is of size  $O(m^t) = (nm)^{O(r)}$ , which is a polynomial in the input size  $\text{poly}(n, |\mathcal{S}|)$ .

Moreover, it is possible to make the reduction zero-error: Indeed, the condition of the niceness can be checked in polynomial time, by using the characterization of Claim 15.

Finally, recall that the  $r$ -bounded set cover problem is NP-hard to approximate within a factor of 2 by Theorem 5 for a sufficiently large constant  $r \in \mathbb{N}$ . Hence, NP-hardness of  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$  follows from Claims 9 and 10. ◀

### 3.2 Reduction from $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$ to $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$

Next, we present a reduction for the minimum circuit size problem for partial functions to that for total functions:

► **Theorem 16.** *There is a (zero-error) randomized polynomial-time many-one reduction from  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$  to  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$ .*

Let  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$  be an instance of  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$ . Let  $r := t + 2$  and  $s := \lceil (2r + 2t) \log m + 2 \rceil = \lceil 4(t + 1) \log m + 2 \rceil$ . We encode each value  $f(x) \in \{0, 1, *\}$  of the partial function  $f$  as a function on a “hypercube”  $\mathbb{Z}_m^s$ : namely, we construct a new *total* function  $g: \mathbb{Z}_m^t \times \mathbb{Z}_m^s \rightarrow \{0, 1\}$  such that  $f(x)$  corresponds to  $(g(x, y))_{y \in \mathbb{Z}_m^s}$ . Specifically, if  $f(x) \neq *$ , then  $f(x)$  is encoded as a hypercube whose origin<sup>5</sup>  $0^s$  is assigned  $f(x)$  and

<sup>5</sup>  $0^s$  denotes the zero of  $\mathbb{Z}_m^s$  for any  $s \in \mathbb{N}$ .

other points are assigned 0; if  $f(x) = *$ , then we pick a random linear subspace  $L_x \subseteq \mathbb{Z}_m^s$  of dimension  $r$  and we encode  $f(x)$  as the characteristic function of  $L_x$ .

Formally, for each  $x \in f^{-1}(*)$ , we pick  $v_x^1, \dots, v_x^r \in_R \mathbb{Z}_m^s$  uniformly and independently at random, and define a random linear subspace  $L_x := \text{span}(v_x^1, \dots, v_x^r)$ . Then the output  $g : \mathbb{Z}_m^t \times \mathbb{Z}_m^s \rightarrow \{0, 1\}$  of our reduction is defined as

$$g(x, y) := \begin{cases} f(x) & (\text{if } f(x) \in \{0, 1\} \text{ and } y = 0^s) \\ 1 & (\text{if } f(x) = * \text{ and } y \in L_x) \\ 0 & (\text{otherwise}) \end{cases}$$

for any  $(x, y) \in \mathbb{Z}_m^t \times \mathbb{Z}_m^s$ .

The idea is as follows: Let us imagine how a minimum  $\text{DNF} \circ \text{MOD}_m$  circuit  $C$  computing  $g$  looks like. We need to cover  $g^{-1}(1)$  by as few affine subspaces as possible. Note that  $g^{-1}(1)$  consists of two parts:  $\{(x, 0^s)\}$  for each  $x \in f^{-1}(1)$ , and  $\{x\} \times L_x$  for each  $x \in f^{-1}(*)$ . In order to cover the latter one, it is likely that we need to use the affine subspace  $\{x\} \times L_x$  itself for each  $x \in f^{-1}(*)$ ; indeed, since each  $L_x$  is a random linear subspace, under our constraints with high probability there is no affine subspace which simultaneously covers (a large fraction of) two random affine subspaces  $\{x\} \times L_x$  and  $\{x'\} \times L_{x'}$  for  $x \neq x' \in f^{-1}(*)$  (Claim 21 below). Therefore, the minimum circuit  $C$  should contain a subcircuit which accepts  $\{x\} \times L_x$  for each  $x \in f^{-1}(*)$ . Now it remains to cover  $\{(x, 0^s)\}$  for each  $x \in f^{-1}(1)$ , but here we can *optionally* cover  $\{(x, 0^s)\}$  for each  $x \in f^{-1}(*)$  (which has been already covered by  $\{x\} \times L_x$ ). This is exactly the same situation as  $(\text{DNF} \circ \text{MOD}_m)$ -MCSP\*; thus with high probability we have  $\text{DNF}_{\text{MOD}_m}(g) = \text{DNF}_{\text{MOD}_m}(f) + |f^{-1}(*)|$ . Details follow.

► **Claim 17.**  $\text{DNF}_{\text{MOD}_m}(g) \leq \text{DNF}_{\text{MOD}_m}(f) + |f^{-1}(*)|$ .

**Proof.** Suppose that a  $\text{DNF} \circ \text{MOD}_m$  circuit  $C = \bigvee_{k=1}^K C_k$  computes  $f$ . For each  $x^* \in f^{-1}(*)$ , take an  $\text{AND} \circ \text{MOD}_m$  formula  $C_{x^*}$  such that  $C_{x^*}^{-1}(1) = \{x^*\} \times L_{x^*}$  (by Lemma 2). Define  $C'(x, y) := \bigvee_{k=1}^K (C_k(x) \wedge (y_1 = 0) \wedge \dots \wedge (y_s = 0)) \vee \bigvee_{x^* \in f^{-1}(*)} C_{x^*}(x, y)$ . It is easy to see that  $C'(x, y) = g(x, y)$  for any  $(x, y) \in \mathbb{Z}_m^t \times \mathbb{Z}_m^s$ . ◀

In order to prove the other direction, let us clarify the desired condition for random linear spaces. We require that  $(L_x)_{x \in f^{-1}(*)}$  is pairwise “disjoint” and that each  $L_x$  is nondegenerated.

► **Definition 18.** We say that  $(L_x)_{x \in f^{-1}(*)}$  is *scattered* if  $|L_x| = m^r$  and  $L_x \cap L_{x'} = \{0^s\}$  for any distinct  $x, x' \in f^{-1}(*)$ .

It is easy to prove that the collection of random linear spaces satisfies the condition above.

► **Claim 19.**  $(L_x)_{x \in f^{-1}(*)}$  is scattered with probability at least  $\frac{1}{2}$ , provided that  $s \geq (2r + 2t) \log m + 2$ .

**Proof.** We first bound the probability that  $(L_x)_{x \in f^{-1}(*)}$  is not pairwise disjoint.

$$\begin{aligned} & \Pr [ L_x \cap L_{x'} \neq \{0^s\} \text{ for some distinct } x, x' \in f^{-1}(*) ] \\ & \leq \sum_{x \neq x' \in f^{-1}(*)} \Pr [ L_x \cap L_{x'} \neq \{0^s\} ] \\ & \leq \sum_{x \neq x' \in f^{-1}(*)} \Pr \left[ \sum_{i=1}^r c_i v_x^i = \sum_{i=1}^r d_i v_{x'}^i, \text{ for some nonzero } (c_i)_{i \in [r]}, (d_i)_{i \in [r]} \right] \\ & < m^{2t} \cdot m^{2r} \cdot 2^{-s} \leq \frac{1}{4}, \end{aligned}$$

where, in the last line, we used the fact that the probability that  $\sum_{i=1}^r c_i v_x^i = \sum_{i=1}^r d_i v_x^i$  is at most  $2^{-s}$  for nonzero (i.e.  $c_i \neq 0, d_j \neq 0$  for some  $i, j \in [r]$ ) coefficients  $(c_i)_{i \in [r]}, (d_i)_{i \in [r]}$ .<sup>6</sup>

Next, we bound the probability that  $|L_x| < m^r$ . Indeed,

$$\begin{aligned} & \Pr [ |L_x| < m^r \text{ for some } x \in f^{-1}(*) ] \\ & \leq \sum_{x \in f^{-1}(*)} \Pr \left[ \sum_{i=1}^r c_i v_x^i = 0^s \text{ for some nonzero } (c_i)_{i \in [r]} \right] \\ & \leq m^t \cdot m^r \cdot 2^{-s} \leq \frac{1}{4}. \end{aligned}$$

Overall, the probability that  $(L_x)_{x \in f^{-1}(*)}$  is not scattered is less than  $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ . ◀

Note that the condition of being scattered can be checked in polynomial time. Indeed, for each  $x \in f^{-1}(*)$ , one can enumerate all the elements of  $L_x$ , which are at most polynomially many in the input size  $m^{O(t)}$ . Thus, our zero-error randomized reduction picks random linear subspaces  $(L_x)_{x \in f^{-1}(*)}$  until we obtain a scattered collection of linear subspaces.

In the rest of the proof, we can thus assume that  $(L_x)_{x \in f^{-1}(*)}$  is scattered. The next claim gives the reverse inequality of Claim 17.

► **Claim 20.**  $\text{DNF}_{\text{MOD}_m}(g) \geq \text{DNF}_{\text{MOD}_m}(f) + |f^{-1}(*)|$  if  $(L_x)_{x \in f^{-1}(*)}$  is scattered.

Let  $C = \bigvee_{k=1}^K C_k$  be a minimum  $\text{DNF} \circ \text{MOD}_m$  circuit computing  $g$ . (In particular,  $K = \text{DNF}_{\text{MOD}_m}(g) \leq \text{DNF}_{\text{MOD}_m}(f) + |f^{-1}(*)| \leq m^{t+1}$ .) For each  $x \in f^{-1}(*)$ , we first extract a subcircuit  $C_{l(x)}$  that covers (a large fraction of) the random linear subspace  $L_x$ . Let  $l(x) \in [K]$  be one of the indices such that  $|C_{l(x)}^{-1}(1) \cap (\{x\} \times L_x)|$  is maximized. That is,  $C_{l(x)}$  covers the largest fraction of the affine subspace  $\{x\} \times L_x$ ; in particular, since  $\bigcup_{k \in [K]} C_k^{-1}(1) \supseteq \{x\} \times L_x$ , there are at least  $|L_x|/K$  ( $= m^r/K \geq m^{r-t-1} \geq 2$ ) points in the set  $C_{l(x)}^{-1}(1) \cap (\{x\} \times L_x)$ . Intuitively, the subcircuits  $\{C_{l(x)} \mid x \in f^{-1}(*)\}$  are supposed to cover random linear subspaces, and the rest of the subcircuits computes  $f$ .

To make the intuition formal, we will prove the following two claims. The first asserts that, under our constraints, no affine subspace can cover a large fraction of two distinct random linear subspaces.

► **Claim 21.**  $l: f^{-1}(*) \rightarrow [K]$  is injective.

The second claim asserts that, if an affine subspace  $C_{l(x')}^{-1}(1)$  covers a large fraction of  $\{x'\} \times L_{x'}$ , then it cannot cover a point  $(x, 0^s)$  such that  $f(x) = 1$ .

► **Claim 22.**  $C_{l(x')}(x, 0^s) = 0$  for any  $x \in f^{-1}(1)$  and  $x' \in f^{-1}(*)$ .

Assuming these two claims, it is easy to prove Claim 20.

**Proof of Claim 20.** For each  $k \in [K]$ , define an  $\text{AND} \circ \text{MOD}_m$  circuit  $C'_k$  as  $C'_k(x) := C_k(x, 0^s)$  on input  $x \in \mathbb{Z}_m^t$ . Define a  $\text{DNF} \circ \text{MOD}_m$  circuit  $C' := \bigvee_{k \in [K] \setminus \{l(x) \mid f(x)=*\}} C'_k$ . By Claim 21, the number of subcircuits in  $C'$  is  $K - |f^{-1}(*)|$ .

We claim that  $C'$  computes  $f$ . Indeed, for any  $x \in f^{-1}(1)$ , we have  $C(x, 0^s) = g(x, 0^s) = f(x) = 1$ ; hence, there is some  $k \in [K]$  such that  $C_k(x, 0^s) = 1$ , which implies that  $C'_k(x) = 1$  by the definition of  $C'_k$ . Claim 22 implies  $k \notin \{l(x') \mid f(x') = *\}$ ; thus  $C'(x) = 1$ . On the other hand, for any  $x \in f^{-1}(0)$ , we have  $C(x, 0^s) = g(x, 0^s) = f(x) = 0$ ; in particular, for any  $k \in [K]$ ,  $C_k(x, 0^s) = 0$ . Thus  $C'_k(x) = 0$  for any  $k \in [K]$ , which implies  $C'(x) = 0$ . ◀

<sup>6</sup> Note that any equation  $ax = b \pmod{m}$  with  $a \neq 0$  is satisfied with probability  $\leq 1/2$  over a random choice of  $x$ .

It remains to prove Claims 21 and 22. We prove the latter first.

**Proof of Claim 22.** Assume, by way of a contradiction, that  $C_{l(x')}(x, 0^s) = 1$  for some  $x \in f^{-1}(1)$  and  $x' \in f^{-1}(*)$ . By the definition of  $l(x')$ , there are at least 2 distinct points  $(x', a)$  and  $(x', b)$  in  $C_{l(x')}^{-1}(1) \cap (\{x'\} \times L_{x'})$ . Since  $C_{l(x')}^{-1}(1)$  is an affine subspace, we have  $(x', a) - (x', b) + (x, 0^s) = (x, a - b) \in C_{l(x')}^{-1}(1)$  (as in the proof of Claim 11). It follows that  $C(x, a - b) = 1$ . Since  $C$  computes  $g$ , we also have  $g(x, a - b) = 1$ , which contradicts the fact that  $a - b \neq 0^s$  and the definition of  $g$ . ◀

**Proof of Claim 21.** Assume that  $l(x_1) = l(x_2) =: k$  for distinct inputs  $x_1, x_2 \in f^{-1}(*)$ . Take any 2 distinct points  $(x_1, a)$  and  $(x_1, b)$  from  $C_k^{-1}(1) \cap (\{x_1\} \times L_{x_1})$  and any point  $(x_2, c)$  from  $C_k^{-1}(1) \cap (\{x_2\} \times L_{x_2})$ . Since  $C_k^{-1}(1)$  is an affine subspace, we have  $(x_1, a) - (x_1, b) + (x_2, c) = (x_2, a - b + c) \in C_k^{-1}(1)$ . We also have  $(x_2, a - b + c) \in \{x_2\} \times L_{x_2}$ , since  $C_k^{-1}(1) \cap (\{x_2\} \times \mathbb{Z}_m^s) \subseteq g^{-1}(1) \cap (\{x_2\} \times \mathbb{Z}_m^s) = \{x_2\} \times L_{x_2}$ . Therefore,  $a - b + c \in L_{x_2}$ . Since  $c \in L_{x_2}$  and this is a linear subspace, it follows that  $a - b \in L_{x_2}$ . On the other hand, by the definition of  $a$  and  $b$ , we have  $0^s \neq a - b \in L_{x_1}$ . However, this is a contradiction because  $0^s \neq a - b \in L_{x_1} \cap L_{x_2} = \{0^s\}$ . ◀

**Proof of Theorem 16.** By Claims 17 and 20, we obtain  $\text{DNF}_{\text{MOD}_m}(g) = \text{DNF}_{\text{MOD}_m}(f) + |f^{-1}(*)|$  for a scattered collection  $(L_x)_{x \in f^{-1}(*)}$ . Since  $s = O(t \log m)$ , the truth table of  $g$  is of length  $m^{t+s} = m^{O(t \log m)}$ , which is a polynomial in the input length for every constant  $m \geq 2$ . Finally, since it is possible to check whether  $(L_x)_{x \in f^{-1}(*)}$  is scattered in polynomial time, the reduction is zero-error. ◀

**On our proof strategy and the restriction to functions over boolean inputs ( $m > 2$ ).** The linear-algebraic and probabilistic techniques employed here naturally suggest to view a set of inputs for the input instance  $f$  as a subset of the algebraic structure  $\mathbb{Z}_m^n$  (a vector space or module, depending on  $m$ ). In order to establish a similar NP-hardness result with respect to functions on the hypercube and AND-OR-MOD $_m$  circuits, one is tempted to encode elements from the structure  $\mathbb{Z}_m^n$  as binary strings, and to consider a bijection  $\varphi: \mathbb{Z}_m^n \leftrightarrow \Gamma \subseteq \{0, 1\}^*$  between vectors and binary strings. However, a binary encoding allows a bottom-layer modular gate to access individual bits of this encoding, and as a consequence, this gate might accept a set  $A \subseteq \{0, 1\}^*$  that does not correspond under  $\varphi$  to the set of solutions of a modular equation over  $\mathbb{Z}_m$ . When this is the case, our argument no longer works.

Another natural approach would be to restrict the input function to boolean inputs, and to directly view such inputs as elements in  $\{0, 1\}^n \subseteq \mathbb{Z}_m^n$ . Here certain technical difficulties are transferred to our probabilistic analysis involving affine subspaces of  $\mathbb{Z}_m^n$ , and it is not immediately clear to us how to modify the argument in this case.

For these reasons, when  $m > 2$  our techniques do not seem to be directly applicable to functions defined over boolean inputs only, and a more complicated argument might be necessary. Note however that this does not exclude the existence of different and potentially simpler reductions among these and other intermediary problems.

#### 4 Derandomization and Pseudorandom Generators for AND ◦ MOD $_m$

In this section, we present a unified way of efficiently derandomizing the zero-error reductions of Section 3. The crucial idea is that certain *subconditions* of being nice or scattered can be checked by AND ◦ MOD $_m$  circuits over  $\mathbb{Z}_m^n$ ; hence, a pseudorandom generator for AND ◦ MOD $_m$  circuits can be used to derandomize the reductions.

In order to achieve this, we show that there exists a quick pseudorandom generator with logarithmic seed length that fools any  $\text{AND} \circ \text{MOD}_m$  circuit (regardless of its size), a result that might be of independent interest.

► **Theorem 23.** *For every  $\epsilon = \epsilon(n) > 0$  and each  $m \geq 2$ , there exists a quick pseudorandom generator  $G = \{G_n : [\Gamma_n] \rightarrow \mathbb{Z}_m^n\}_{n \in \mathbb{N}}$  that  $\epsilon$ -fools any  $\text{AND} \circ \text{MOD}_m$  circuit over  $\mathbb{Z}_m^n$ , where  $\Gamma_n = \text{poly}(n, 1/\epsilon, m)$  is a positive integer.*

Here we say that, for  $\epsilon > 0$  and an integer  $m \geq 2$ , a function  $G_n : [\Gamma_n] \rightarrow \mathbb{Z}_m^n$   $\epsilon$ -fools  $\text{AND} \circ \text{MOD}_m$  circuits if  $|\mathbb{E}_{\gamma \in_R [\Gamma_n]} [C(G_n(\gamma))] - \mathbb{E}_{v \in_R \mathbb{Z}_m^n} [C(v)]| \leq \epsilon$  for every  $\text{AND} \circ \text{MOD}_m$  circuit  $C$ ; such a function  $G_n$  is called an  $\epsilon$ -pseudorandom generator for  $\text{AND} \circ \text{MOD}_m$  circuits. We say that a family  $\{G_n\}_{n \in \mathbb{N}}$  of pseudorandom generators is *quick* if  $G_n$  can be computed in  $\text{poly}(\Gamma_n)$  time. (Recall that  $[\Gamma_n]$  denotes the set  $\{1, \dots, \Gamma_n\}$ , which means that the seed-length of  $G_n$  is logarithmic in  $n$ ,  $m$ , and  $1/\epsilon$  when its input elements are represented as binary strings.)

## 4.1 Derandomizing the Reductions

We defer a proof of Theorem 23 to the next subsection, and present its applications first: The pseudorandom generator implies polynomial-time derandomizations of the reductions presented in Section 3.

► **Theorem 24** (Restatement of Theorem 1). *(DNF  $\circ$  MOD<sub>m</sub>)-MCSP is NP-hard under polynomial-time many-one reductions.*

Our basic strategy is as follows: Each reduction of Section 3 employs random variables that take value on  $\mathbb{Z}_m^k$ , for different choices of  $k$ . To derandomize the reductions, we simply replace these random variables by the output of the pseudorandom generator of Theorem 23; then we try all possible  $\Gamma_n$  seeds of  $G_n$ , and check whether the generated random variables satisfy the desired condition (which can be done in polynomial time). Below we give details for each reduction, starting with the second.

**Derandomizing the second reduction.** We start with the reduction from  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$  to  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$ . The reduction required a scattered collection of linear subspaces, which is provided by the probabilistic argument of Claim 19. Here we present a deterministic construction of such a collection.

► **Theorem 25.** *For any integer  $m \geq 2$ , there exists a deterministic algorithm that, on inputs  $t$  and  $r$ , outputs a scattered collection of  $r$ -dimensional linear subspaces  $(L_h)_{h \in [H]}$  for  $H := m^t$ . Specifically,*

1.  $L_h$  is a linear subspace of  $\mathbb{Z}_m^s$  for  $s := \lceil (2r + 2t) \log m + 2 \rceil$ ,
2.  $|L_h| = m^r$ , and
3.  $L_h \cap L_{h'} = \{0^s\}$  for any distinct  $h, h' \in [H]$ .

*The running time of the algorithm is  $m^{O((r+t) \log m)}$ .*

In the proof of Theorem 16, we picked random vectors  $v_x^1, \dots, v_x^r \in_R \mathbb{Z}_m^s$  and defined  $L_x := \text{span}(v_x^1, \dots, v_x^r)$  for each  $x \in f^{-1}(*) \subseteq \mathbb{Z}_m^t$ . We take a similar approach, but instead of generating vectors uniformly at random, we use the output of the pseudorandom generator as the source of randomness. Specifically, let  $\gamma \in [\Gamma_{rsH}]$  be a seed of the pseudorandom generator of  $G_{rsH}$ ; define vectors  $(v_h^1, \dots, v_h^r)_{h \in [H]} := G_{rsH}(\gamma) \in (\mathbb{Z}_m^s)^H$ ; then, define  $L_h := \text{span}(v_h^1, \dots, v_h^r)$  for each  $h \in [H]$ . We show that the probabilistic argument of Claim 19 still works even if the randomness is replaced in this way:

► **Claim 26.** Let  $G_{rsH}$  be the pseudorandom generator of Theorem 23 with error parameter  $\epsilon = 2^{-s}$ . Pick a seed  $\gamma \in_R [\Gamma_{rsH}]$  uniformly at random, and define a collection  $(L_h)_{h \in [H]}$  of linear subspaces as above. Then,  $(L_h)_{h \in [H]}$  is scattered with nonzero probability.

**Proof.** Note that union bounds hold for any distribution; hence, by using the union bounds as in Claim 19, the probability that  $(L_h)_{h \in [H]}$  is not pairwise disjoint is

$$\begin{aligned} & \Pr [ L_h \cap L_{h'} \neq \{0^s\} \text{ for some distinct } h, h' \in [H] ] \\ & \leq \sum_{h \neq h' \in [H]} \sum_{(c_i), (d_i)} \Pr \left[ \sum_{i=1}^r c_i v_h^i = \sum_{i=1}^r d_i v_{h'}^i \right], \end{aligned} \quad (2)$$

where the second sum is taken over all nonzero coefficient vectors  $(c_i)_{i \in [r]}$  and  $(d_i)_{i \in [r]}$  with entries  $c_i, d_i \in \mathbb{Z}_m$ . If the random vectors  $(v_h^i)_{h,i}$  were uniformly distributed, the probability in (2) could be bounded by  $2^{-s}$  as in Claim 19; Here the probability is taken over a random seed  $\gamma \in_R [\Gamma_{rsH}]$  of the pseudorandom generator  $G_{rsH}$ . The condition that  $\sum_{i=1}^r c_i v_h^i = \sum_{i=1}^r d_i v_{h'}^i$  can be checked by some  $\text{AND} \circ \text{MOD}_m$  circuit that takes  $(v_h^i)_{h,i}$  as input; thus the circuit is  $\epsilon$ -fooled by the pseudorandom generator; as a consequence, the probability (2) is strictly less than  $m^{2t} \cdot m^{2r} \cdot (2^{-s} + \epsilon) \leq \frac{1}{2}$ .

Similarly,

$$\begin{aligned} & \Pr [ |L_h| < m^r \text{ for some } h \in [H] ] \\ & \leq \sum_{h \in [H]} \sum_{(c_i)} \cdot \Pr \left[ \sum_{i=1}^r c_i v_h^i = 0^s \right] \\ & < m^t \cdot m^r \cdot (2^{-s} + \epsilon) \leq \frac{1}{2}. \end{aligned}$$

Overall, the probability that  $(L_h)_{h \in [H]}$  is not scattered is strictly less than  $\frac{1}{2} + \frac{1}{2} = 1$ . ◀

**Proof of Theorem 25.** By Claim 26, there exists some seed  $\gamma \in [\Gamma_{rsH}]$  such that the output  $G_{rsH}(\gamma)$  defines a scattered collection  $(L_h)_{h \in [H]}$  of linear subspaces. By exhaustively searching all the seeds, one can enumerate all the outputs of  $G_{rsH}$  in time  $\text{poly}(\Gamma_{rsH}) = \text{poly}(rsH, 2^s, m)$ . Moreover, one can check whether  $G_{rsH}(\gamma)$  defines a scattered collection for each  $\gamma \in [\Gamma_{rsH}]$  in time  $\text{poly}(H, m^s)$ . Overall, the running time of our construction is  $\text{poly}(m^s) = m^{O((r+t) \log m)}$ . ◀

The randomized reduction of Theorem 16 can be now derandomized, using the deterministic construction of Theorem 25 for  $r := t + 2$ .

► **Corollary 27.** There is a polynomial-time ( $m^{O(t \log m)}$  time on input length  $O(m^t)$ ) many-one reduction from  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$  to  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}$ .

**Derandomizing the first reduction.** We now consider the reduction from the  $r$ -bounded set cover problem to  $(\text{DNF} \circ \text{MOD}_m)\text{-MCSP}^*$ . Let  $[n]$  be the universe, and  $\mathcal{S} \subseteq \binom{[n]}{\leq r}$  be an input to the set cover problem. Derandomizing the reduction amounts to a deterministic construction of a nice collection  $(v^i)_{i \in [n]}$  of vectors. We generate the random vectors using the pseudorandom generator for  $\text{AND} \circ \text{MOD}_m$  circuits, and show that the probabilistic argument of Claim 14 still works.

► **Claim 28 (Revised Claim 14).** Let  $G_{tn}$  be the pseudorandom generator of Theorem 23 with error parameter  $\epsilon < m^{-t}$ . Pick a seed  $\gamma \in_R [\Gamma_{tn}]$  uniformly at random. Define  $(v^1, \dots, v^n) := G_{tn}(\gamma) \in (\mathbb{Z}_m^t)^n$ . If  $t \geq r + ((r+2) \log n + \log |\mathcal{S}| + 1) / \log m$ , then  $(v^i)_{i \in [n]}$  is nice with nonzero probability.

**Proof.** By using union bounds as in Claim 14, it is sufficient to prove

$$n^{r+2} \cdot |\mathcal{S}| \cdot m^r \cdot \Pr \left[ v^{j_S} = \sum_{i \in S} d_i v^i - \sum_{i \in I \setminus \{j_S\}} c_i v^i \right] < 1 \quad (3)$$

for coefficients  $(c_i)_{i \in I}$ ,  $(d_i)_{i \in S}$  and  $j_S \in I \setminus S$ , where the probability is taken over a random seed  $\gamma$ .

The condition  $v^{j_S} = \sum_{i \in S} d_i v^i - \sum_{i \in I \setminus \{j_S\}} c_i v^i$  can be checked by an  $\text{AND} \circ \text{MOD}_m$  circuit that takes  $(v^1, \dots, v^n) \in \mathbb{Z}_m^{tn}$  as input. By Theorem 23, we get

$$\Pr \left[ v^{j_S} = \sum_{i \in S} d_i v^i - \sum_{i \in I \setminus \{j_S\}} c_i v^i \right] \leq m^{-t} + \epsilon.$$

Consequently, due to our choice of  $t$  and using  $\epsilon < m^{-t}$ , the left-hand side of (3) is strictly less than

$$n^{r+2} \cdot |\mathcal{S}| \cdot m^r \cdot 2m^{-t} \leq 1,$$

which completes the proof.  $\blacktriangleleft$

In particular, there exists some seed  $\gamma \in [\Gamma_{tn}]$  such that  $(v^1, \dots, v^n) = G_{tn}(\gamma)$  is nice. The number of seeds is at most  $\Gamma_{tn} = \text{poly}(tn, 1/\epsilon, m) = \text{poly}(n, m^t) = (nm)^{O(r)}$ , which is a polynomial in the input length; hence, in polynomial time, one can try all possible seeds and find a nice collection  $(v^i)_{i \in [n]}$  of vectors. Thus the reduction of Theorem 8 can be derandomized:

► **Corollary 29.** *(DNF  $\circ$  MOD $_m$ )-MCSP\* is NP-hard under polynomial-time many-one reductions.*

**Proof of Theorem 24.** Immediate from Corollaries 29 and 27.  $\blacktriangleleft$

## 4.2 Near-Optimal Pseudorandom Generators for $\text{AND} \circ \text{MOD}_m$

This subsection contains a proof of Theorem 23. We assume basic familiarity with concepts from analysis of boolean functions [35]. For simplicity, we first focus on the case of  $m = 2$ , which admits a simpler proof.

**Proof for  $m = 2$ .** An  $\epsilon$ -biased generator, introduced by Naor and Naor [34], is a pseudorandom generator for XOR functions. That is, we say that a function  $G: \{0, 1\}^s \rightarrow \{0, 1\}^n$  is an  $\epsilon$ -biased generator if  $|\mathbb{E}_{x \in_R \{0, 1\}^n} [\chi_S(x)] - \mathbb{E}_{s \in_R \{0, 1\}^s} [\chi_S(G(s))]| \leq \epsilon$  for any  $S \subseteq [n]$ , where  $\chi_S(x) := \bigoplus_{i \in S} x_i$ . While this definition only requires the generator to fool XOR functions, it can be shown that any Boolean function with small  $\ell_1$  Fourier norm can be fooled by  $\epsilon$ -biased generators.

► **Lemma 30** (see e.g., [16, Lemma 2.5]). *Every function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  can be  $\epsilon \hat{\|f\|_1}$  fooled by any  $\epsilon$ -biased generator. Here,  $\hat{\|f\|_1} := \sum_{S \subseteq [n]} |\hat{f}(S)|$ .*

**Proof Sketch.** Use the Fourier expansion  $f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \chi_S(x)$ , and apply the triangle inequality.  $\blacktriangleleft$

Moreover, it is known that any  $\text{AND} \circ \text{XOR}$  circuit  $f$  has  $\hat{\|f\|_1} = 1$ .



► **Lemma 31** (see e.g., [35, Proposition 3.12]).  $\|f\|_1 = 1$  for any Boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  computable by a nontrivial AND  $\circ$  XOR circuit.

**Proof Sketch.** Let  $H + a \subseteq \{0, 1\}^n$  be the (nonempty) affine subspace accepted by  $f$ . Take a basis of  $H^\perp$ . Write a characteristic function of  $f$  using the basis, and expand it to obtain a Fourier expansion of  $f$ . ◀

Combining these two lemmas, any  $\epsilon$ -biased generator fools AND  $\circ$  XOR circuits. Moreover, Naor and Naor [34] gave an explicit construction of an  $\epsilon$ -biased generator of seed length  $O(\log n + \log(1/\epsilon))$ , from which Theorem 23 follows when  $m = 2$ .

In the proof sketched above, we exploited the fact that  $\{0, 1\}^n = \mathbb{Z}_2^n$  is a vector space: We took a basis of a linear subspace in the proof of Lemma 31. In order to generalize the result to the case of  $m \geq 2$ , we need a more direct proof which does not rely on a basis.

**Proof for any  $m \geq 2$ .** Azar, Motwani and Naor [9] generalized the notion of  $\epsilon$ -biased generator on  $\{0, 1\}^n$  to  $\mathbb{Z}_m^n$  for any integer  $m \geq 2$ , and gave an explicit construction. We review the generalized notion and their result below.

► **Definition 32** ([9]). For a probability distribution  $\mathcal{D}$  over  $\mathbb{Z}_m^n$  and a vector  $a \in \mathbb{Z}_m^n$ ,  $\text{bias}_{\mathcal{D}}(a)$  is defined as follows: for  $g := \gcd(a_1, \dots, a_n, m)$ ,

$$\text{bias}_{\mathcal{D}}(a) := \frac{1}{g} \max_{0 \leq k < m/g} \left| \Pr_{x \sim \mathcal{D}} [\langle a, x \rangle = kg] - \frac{g}{m} \right|.$$

We say that a distribution  $\mathcal{D}$  is  $\epsilon$ -biased if  $\text{bias}_{\mathcal{D}}(a) \leq \epsilon$  for every  $a \in \mathbb{Z}_m^n$ . We say that a function  $G: [\Gamma] \rightarrow \mathbb{Z}_m^n$  is an  $\epsilon$ -biased generator if the distribution  $G(\gamma)$  for a random seed  $\gamma \in_R [\Gamma]$  is  $\epsilon$ -biased.

► **Theorem 33** ([9, Theorem 6.1]). For  $m(n) \geq 2$  and  $\epsilon = \epsilon(n) > 0$ , there exists a quick  $\epsilon$ -biased generator  $G = \{G_n: [\Gamma_n] \rightarrow \mathbb{Z}_m^n\}_{n \in \mathbb{N}}$  for some  $\Gamma_n = \text{poly}(n, 1/\epsilon, m)$ .

We use the same pseudorandom generator  $G$  as in Theorem 33. In what follows, we will show that any  $\epsilon$ -biased generator  $m\epsilon$ -fools AND  $\circ$  MOD $_m$  circuits, which completes the proof of Theorem 23.

Define  $e_m: \mathbb{Z}_m \rightarrow \mathbb{C}^\times$  as  $e_m(k) := \exp(2\pi\sqrt{-1} \cdot k/m)$  for  $k \in \mathbb{Z}_m$ .

► **Lemma 34.** For any distribution  $\mathcal{D}$  on  $\mathbb{Z}_m^n$  and any nonzero vector  $a \in \mathbb{Z}_m^n$ , we have

$$\left| \mathbb{E}_{x \sim \mathcal{D}} [e_m(\langle a, x \rangle)] \right| \leq m \cdot \text{bias}_{\mathcal{D}}(a).$$

**Proof.** The proof follows the same approach of [9, Lemma 4.4]. Let  $g := \gcd(a_1, \dots, a_n, m)$ .

$$\begin{aligned} \left| \mathbb{E}_{x \sim \mathcal{D}} [e_m(\langle a, x \rangle)] \right| &= \left| \sum_{0 \leq k < m/g} e_m(kg) \Pr_{x \sim \mathcal{D}} [\langle a, x \rangle = kg] \right| \\ &= \left| \sum_{0 \leq k < m/g} e_m(kg) \left( \Pr_{x \sim \mathcal{D}} [\langle a, x \rangle = kg] - \frac{g}{m} \right) \right| \\ &\leq \sum_{0 \leq k < m/g} |e_m(kg)| \cdot \left| \Pr_{x \sim \mathcal{D}} [\langle a, x \rangle = kg] - \frac{g}{m} \right| \\ &\leq \frac{m}{g} \cdot 1 \cdot g \cdot \text{bias}_{\mathcal{D}}(a) = m \cdot \text{bias}_{\mathcal{D}}(a), \end{aligned}$$



where the first equality follows from the fact that  $\langle a, x \rangle$  is a multiple of  $g$  for any  $x \in \mathbb{Z}_m^n$ , and in the second equality we used that  $\sum_{0 \leq k < m/g} e_m(kg) = 0$  for  $g < m$ , which is true if  $a \neq 0^n$ .  $\blacktriangleleft$

As a consequence of the previous lemma, we can prove that any affine function can be “fooled”:

► **Lemma 35.** *For any  $\epsilon$ -biased probability distribution  $\mathcal{D}$  on  $\mathbb{Z}_m^n$ , any vector  $a \in \mathbb{Z}_m^n$ , and any scalar  $b \in \mathbb{Z}_m$ ,*

$$\left| \mathbb{E}_{x \sim \mathcal{D}} [e_m(\langle a, x \rangle + b)] - \mathbb{E}_{x \in_R \mathbb{Z}_m^n} [e_m(\langle a, x \rangle + b)] \right| \leq m\epsilon.$$

**Proof.** When  $a = 0^n$ , both expectations are constant, and hence the lemma follows. Otherwise, we have  $\mathbb{E}_{x \in_R \mathbb{Z}_m^n} [e_m(\langle a, x \rangle)] = 0$ , since this expression can be written as a product of expectations, and one of them evaluates to zero. Using Lemma 34, we obtain

$$\begin{aligned} & \left| \mathbb{E}_{x \sim \mathcal{D}} [e_m(\langle a, x \rangle + b)] - \mathbb{E}_{x \in_R \mathbb{Z}_m^n} [e_m(\langle a, x \rangle + b)] \right| \\ &= |e_m(b)| \cdot \left| \mathbb{E}_{x \sim \mathcal{D}} [e_m(\langle a, x \rangle)] - \mathbb{E}_{x \in_R \mathbb{Z}_m^n} [e_m(\langle a, x \rangle)] \right| = 1 \cdot \left| \mathbb{E}_{x \sim \mathcal{D}} [e_m(\langle a, x \rangle)] \right| \\ &\leq m \text{bias}_{\mathcal{D}}(a) \leq m\epsilon. \end{aligned} \quad \blacktriangleleft$$

► **Theorem 36.** *For any  $\epsilon$ -biased probability distribution  $\mathcal{D}$  on  $\mathbb{Z}_m^n$  and any function  $f : \mathbb{Z}_m^n \rightarrow \{0, 1\}$  computable by some  $\text{AND} \circ \text{MOD}_m$  circuit,*

$$\left| \mathbb{E}_{x \sim \mathcal{D}} [f(x)] - \mathbb{E}_{x \in_R \mathbb{Z}_m^n} [f(x)] \right| \leq m\epsilon$$

**Proof.** Suppose that an  $\text{AND} \circ \text{MOD}_m$  circuit computing  $f$  has  $K$   $\text{MOD}_m$  gates, and, for each  $k \in [K]$ , let  $g_k : \mathbb{Z}_m^n \rightarrow \mathbb{Z}_m$  denote the *affine function* that corresponds to the  $k$ th  $\text{MOD}_m$  gate. That is,  $g_k(x) = \langle a_k, x \rangle + b_k$  for some vector  $a_k \in \mathbb{Z}_m^n$  and some scalar  $b_k \in \mathbb{Z}_m$ ; moreover, for any input  $x \in \mathbb{Z}_m^n$ ,  $f(x) = 1$  if and only if  $g_k(x) = 0$  for all  $k \in [K]$ .

We employ the following construction. Let  $p(z)$  be the polynomial over  $\mathbb{C}$  defined as follows.

$$p(z) := \frac{1}{m} \prod_{\alpha \in \mathbb{Z}_m \setminus \{0\}} (z - e_m(\alpha)) \quad (4)$$

$$= \frac{1}{m} \frac{z^m - 1}{z - 1} = \frac{1}{m} \sum_{i=0}^{m-1} z^i, \quad (5)$$

where the second equality holds because the roots of the polynomial  $z^m - 1$  are  $\{e_m(\alpha) \mid \alpha \in \mathbb{Z}_m\}$ . Useful properties of this polynomial are that, by (4), we have  $p(e_m(\alpha)) = 0$  for any  $\alpha \in \mathbb{Z}_m \setminus \{0\}$ , and that  $p(e_m(0)) = p(1) = 1$  because of (5). Using the polynomial, we can write  $f$  as follows:

$$\begin{aligned}
f(x) &= \bigwedge_{k \in [K]} [g_k(x) = 0] \\
&= \bigwedge_{k \in [K]} [p(e_m(g_k(x))) = 1] \\
&= \prod_{k \in [K]} p(e_m(g_k(x))) \\
&= \prod_{k \in [K]} \left( \frac{1}{m} \sum_{j=0}^{m-1} e_m(j \cdot g_k(x)) \right) \\
&= \frac{1}{m^K} \prod_{k \in [K]} \sum_{\alpha_k \in \mathbb{Z}_m} e_m(\alpha_k g_k(x)) \\
&= \frac{1}{m^K} \sum_{\alpha \in \mathbb{Z}_m^K} e_m \left( \sum_{k \in [K]} \alpha_k g_k(x) \right).
\end{aligned}$$

Now, by using Lemma 35, we obtain

$$\begin{aligned}
&\left| \mathbb{E}_{x \sim \mathcal{D}} [f(x)] - \mathbb{E}_{x \in_R \mathbb{Z}_m^n} [f(x)] \right| \\
&\leq \frac{1}{m^K} \sum_{\alpha \in \mathbb{Z}_m^K} \left| \mathbb{E}_{x \sim \mathcal{D}} \left[ e_m \left( \sum_{k \in [K]} \alpha_k g_k(x) \right) \right] - \mathbb{E}_{x \in_R \mathbb{Z}_m^n} \left[ e_m \left( \sum_{k \in [K]} \alpha_k g_k(x) \right) \right] \right| \\
&\leq m\epsilon,
\end{aligned}$$

where in the last inequality we used the fact that  $\sum_{k \in [K]} \alpha_k g_k(x)$  is an affine function. ◀

**Proof of Theorem 23.** The result is immediate from Theorems 33 and 36. ◀

---

## References

- 1 Adi Akavia, Andrej Bogdanov, Siyao Guo, Akshay Kamath, and Alon Rosen. Candidate weak pseudorandom functions in  $AC^0 \circ MOD_2$ . In Moni Naor, editor, *Innovations in Theoretical Computer Science, ITCIS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 251–260. ACM, 2014. doi:10.1145/2554797.2554821.
- 2 Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006. doi:10.1137/050628994.
- 3 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. 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 25–32. Springer, 2014. doi:10.1007/978-3-662-44465-8\_3.
- 4 Eric Allender, Joshua A. Grochow, and Christopher Moore. Graph isomorphism and circuit size. *CoRR*, abs/1511.08189, 2015. arXiv:1511.08189.
- 5 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 J. Comput.*, 38(1):63–84, 2008. URL: <http://dblp.uni-trier.de/db/journals/siamcomp/siamcomp38.html#AllenderHMPS08>.

- 6 Eric Allender and Shuichi Hirahara. New insights on the (non-)hardness of circuit minimization and related problems. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, volume 83 of *LIPICs*, pages 54:1–54:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.MFCS.2017.54.
- 7 Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPICs*, pages 21–33. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPICs.STACS.2015.21.
- 8 Eric Allender, Michal Koucký, Detlef Ronneburger, and Sambuddha Roy. The pervasive reach of resource-bounded kolmogorov complexity in computational complexity theory. *J. Comput. Syst. Sci.*, 77(1):14–40, 2011. doi:10.1016/j.jcss.2010.06.004.
- 9 Yossi Azar, Rajeev Motwani, and Joseph Naor. Approximating probability distributions using small sample spaces. *Combinatorica*, 18(2):151–171, 1998. doi:10.1007/PL00009813.
- 10 Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography.*, pages 79–158. Springer International Publishing, 2017. doi:10.1007/978-3-319-57048-8\_3.
- 11 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.10.
- 12 Arkadev Chattopadhyay and Shachar Lovett. Linear systems over finite abelian groups. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011*, pages 300–308. IEEE Computer Society, 2011. doi:10.1109/CCC.2011.25.
- 13 Arkadev Chattopadhyay and Avi Wigderson. Linear systems over composite moduli. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 43–52. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.17.
- 14 Gil Cohen and Igor Shinkar. The complexity of DNF of parities. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 47–58. ACM, 2016. doi:10.1145/2840728.2840734.
- 15 Sebastian Czort. The complexity of minimizing disjunctive normal form formulas. Master’s Thesis, University of Aarhus, 1999.
- 16 Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:141, 2009. URL: <http://ecc.eccc.hpi-web.de/report/2009/141>.
- 17 Uriel Feige. A threshold of  $\ln n$  for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
- 18 Vitaly Feldman. Hardness of approximate two-level logic minimization and PAC learning with membership queries. *J. Comput. Syst. Sci.*, 75(1):13–26, 2009. doi:10.1016/j.jcss.2008.07.007.
- 19 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- 20 Chris Godsil. Double orthogonal complement of a finite module. MathOverflow (Retrieved 19-01-2018). URL: <https://mathoverflow.net/q/75268>.

- 21 Parikshit Gopalan, Daniel M. Kane, and Raghu Meka. Pseudorandomness via the discrete fourier transform. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 903–922. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.60.
- 22 Vince Grolmusz. A lower bound for depth-3 circuits with MOD  $m$  gates. *Inf. Process. Lett.*, 67(2):87–90, 1998. doi:10.1016/S0020-0190(98)00093-3.
- 23 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.18.
- 24 John M. Hitchcock and Aduri Pavan. On the np-completeness of the minimum circuit size problem. In Prahladh Harsha and G. Ramalingam, editors, *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*, volume 45 of *LIPICs*, pages 236–245. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPICs.FSTTCS.2015.236.
- 25 Stasys Jukna. On graph complexity. *Combinatorics, Probability & Computing*, 15(6):855–876, 2006. doi:10.1017/S0963548306007620.
- 26 Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*. Springer, 2012.
- 27 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79. ACM, 2000. doi:10.1145/335305.335314.
- 28 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. URL: <http://www.cs.berkeley.edu/~luca/cs172/karp.pdf>.
- 29 Subhash Khot and Rishi Saket. Hardness of minimizing and learning DNF expressions. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 231–240. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.37.
- 30 Jan Krajíček. *Forcing with Random Variables and Proof Complexity*, volume 382 of *London Mathematical Society lecture note series*. Cambridge University Press, 2011. URL: <http://www.cambridge.org/de/academic/subjects/mathematics/logic-categories-and-sets/forcing-random-variables-and-proof-complexity?format=PB>.
- 31 William J. Masek. Some NP-complete set covering problems. Unpublished Manuscript, 1979.
- 32 Cody D. Murray and Richard Ryan Williams. On the (non) np-hardness of computing circuit complexity. In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPICs*, pages 365–380. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPICs.CCC.2015.365.
- 33 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. doi:10.1137/0222053.
- 34 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. URL: <http://dblp.uni-trier.de/db/journals/siamcomp/siamcomp22.html#NaorN93>.

- 35 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>.
- 36 Igor Carboni Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In Ryan O’Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPICs*, pages 18:1–18:49. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CCC.2017.18.
- 37 Ramamohan Paturi, Michael E. Saks, and Francis Zane. Exponential lower bounds for depth three boolean circuits. *Computational Complexity*, 9(1):1–15, 2000. doi:10.1007/PL00001598.
- 38 Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988. doi:10.1145/48014.63140.
- 39 Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997. doi:10.1006/jcss.1997.1494.
- 40 Petr Slavík. A tight analysis of the greedy algorithm for set cover. 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 435–441. ACM, 1996. doi:10.1145/237814.237991.
- 41 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.
- 42 Luca Trevisan. Non-approximability results for optimization problems on bounded degree instances. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 453–461. ACM, 2001. doi:10.1145/380752.380839.

### **A** Proof of Fact 3 – Double Orthogonal Complement in $(\mathbb{Z}/m\mathbb{Z})^n$

In this section we present the proof of Fact 3, which for convenience is reformulated as Theorem 37 stated below. Our presentation follows the proof outlined in [20].

Recall the following concepts. We consider the Abelian group  $G := (\mathbb{Z}/m\mathbb{Z})^n$  equipped with component-wise addition modulo  $m$ , and let  $\langle x, y \rangle := \sum_{i \in [n]} x_i y_i \pmod m$ , where  $x, y \in G$ . For a subgroup  $V$  of  $G$ , define  $V^\perp := \{x \in G \mid \langle x, y \rangle = 0 \text{ for all } y \in V\}$ , which is again a subgroup of  $G$ .

► **Theorem 37** (folklore).  $V^{\perp\perp} = V$  for any subgroup  $V$  of  $G = (\mathbb{Z}/m\mathbb{Z})^n$ .

It is easy to see  $V \subseteq V^{\perp\perp}$ : indeed, for any  $x \in V$ , we have  $\langle x, y \rangle = 0$  for each  $y \in V^\perp$  by the definition of  $V^\perp$ ; hence  $x \in V^{\perp\perp}$ . Therefore, it is sufficient to show that the size of  $V^{\perp\perp}$  is equal to that of  $V$ . To this end, we prove the following claim.

► **Claim 38**.  $|V^\perp| = |G|/|V|$  for any subgroup  $V$  of  $G$ .

Note that, applying this claim twice, we obtain  $|V^{\perp\perp}| = |G|/|V^\perp| = |G|/(|G|/|V|) = |V|$ , which completes the proof of Theorem 37. Claim 38 will be proved by combining the three claims below.

Let  $H$  be any finite Abelian group. A *character* of the group  $H$  is a homomorphism  $\chi: H \rightarrow \mathbb{C}^\times$ . Let  $\widehat{H}$  denote the dual group of  $H$ , that is, the group of all characters of  $H$ . (See e.g. [35, Section 8.5] for more details.) It is known that the order of a group  $H$  and the order of its dual group  $\widehat{H}$  are the same.

► **Claim 39** ([35, Corollary of Proposition 8.55 and Exercise 8.35]).  $|H| = |\widehat{H}|$  for any finite Abelian group  $H$ .

For any subgroup  $V$  of  $G$ , define  $V^* := \{\chi \in \widehat{G} \mid \chi(v) = 1 \text{ for every } v \in V\}$ .

► **Claim 40.**  $\widehat{G/V} \cong V^*$  for any subgroup  $V$  of  $G$ .

**Proof.** We define an isomorphism  $\varphi: \widehat{G/V} \rightarrow V^*$ . Given  $\chi \in \widehat{G/V}$ , we define  $\varphi(\chi): G \rightarrow \mathbb{C}^\times$  by  $\varphi(\chi)(x) := \chi(x+V)$  for  $x \in G$ . We claim that  $\varphi(\chi)$  is indeed in  $V^*$ : First,  $\varphi(\chi): G \rightarrow \mathbb{C}^\times$  is a homomorphism since  $\varphi(\chi)(x+y) = \chi(x+y+V) = \chi((x+V)+(y+V)) = \chi(x+V)\chi(y+V)$  for any  $x, y \in G$ . Second,  $\varphi(\chi)(v) = \chi(v+V) = \chi(V) = 1$  for any  $v \in V$ . (Here, we used the fact that the homomorphism  $\chi$  maps the identity  $0+V \in G/V$  to the identity  $1 \in \mathbb{C}^\times$ .)

We claim that  $\varphi$  is a homomorphism. Indeed,  $\varphi(\chi_1\chi_2)(x) = (\chi_1\chi_2)(x+V) = \chi_1(x+V)\chi_2(x+V) = \varphi(\chi_1)(x)\varphi(\chi_2)(x)$  for any  $x \in G$  and any  $\chi_1, \chi_2 \in \widehat{G/V}$ ; hence  $\varphi(\chi_1\chi_2) = \varphi(\chi_1)\varphi(\chi_2)$ .

In order to prove that  $\varphi$  is a bijection, we construct an inverse map  $\psi: V^* \rightarrow \widehat{G/V}$ . Given  $\chi \in V^*$ , define  $\psi(\chi)(a+V) := \chi(a)$  for any coset  $a+V \in G/V$ . Note that this map is well defined since  $a+V = b+V$  implies  $a-b \in V$ , and thus  $1 = \chi(a-b) = \chi(a)/\chi(b)$ . It is straightforward to see that  $\psi = \varphi^{-1}$ : indeed,  $\psi(\varphi(\chi))(a+V) = \varphi(\chi)(a) = \chi(a+V)$  and  $\varphi(\psi(\chi))(a) = \psi(\chi)(a+V) = \chi(a)$  for any  $a \in G$ . Hence  $\varphi$  is both injective and surjective, and consequently, an isomorphism. ◀

► **Claim 41.**  $V^* \cong V^\perp$  for any subgroup  $V$  of  $G = (\mathbb{Z}/m\mathbb{Z})^n$ .

**Proof.** We first prepare some notation: For any  $i \in [n]$ , let  $e_i \in G$  be the vector whose value is 1 on the  $i$ th coordinate and is 0 on the other coordinates. Let  $\omega := \exp(2\pi\sqrt{-1}/m) \in \mathbb{C}^\times$  denote the  $m$ th root of unity.

We construct an isomorphism  $\varphi: V^\perp \rightarrow V^*$ . Given  $x \in V^\perp$ , define  $\varphi(x) \in V^*$  as  $\varphi(x)(y) := \omega^{\langle x, y \rangle}$  for any  $y \in G$ . Note that the image of  $\varphi$  is contained in  $V^*$ : indeed, for any  $v \in V^\perp$ , we have  $\varphi(x)(v) = \omega^{\langle x, v \rangle} = \omega^0 = 1$ .

We claim that  $\varphi$  is injective. It is easy to see that  $\varphi$  is a homomorphism; thus, it is sufficient to prove that the kernel of  $\varphi$  is just  $0 \in V^\perp$ . If  $\varphi(x)$  is the constant function 1, then  $\langle x, y \rangle = 0$  for any  $y \in G$ ; in particular, letting  $y \in \{e_1, \dots, e_n\}$ , we obtain  $x = 0$ .

Finally, we claim that  $\varphi$  is surjective. For any  $\chi \in V^*$  and any  $i \in [n]$ , there is some  $x_i \in \mathbb{Z}/m\mathbb{Z}$  such that  $\chi(e_i) = \omega^{x_i}$ : indeed, since  $1 = \chi(0) = \chi(m \cdot e_i) = \chi(e_i)^m$ ,  $\chi(e_i)$  is one of the  $m$ th roots of unity. Now we define  $x := \sum_{i=1}^n x_i e_i \in G$ . Then, for any  $y \in G$ ,  $\varphi(x)(y) = \omega^{\langle x, y \rangle} = \prod_{i=1}^n \omega^{x_i y_i} = \prod_{i=1}^n \chi(e_i)^{y_i} = \prod_{i=1}^n \chi(y_i e_i) = \chi(\sum_{i=1}^n y_i e_i) = \chi(y)$ ; hence  $\varphi(x) = \chi$  for some  $x \in G$ . Moreover, for any  $v \in V$ , we have  $\chi(v) = \omega^{\langle x, v \rangle} = 1$  since  $\chi \in V^*$ ; thus we have  $\langle x, v \rangle = 0$ , which implies that  $x \in V^\perp$ . ◀

Combining these three claims, we obtain  $|V^\perp| = |V^*| = |\widehat{G/V}| = |G/V| = |G|/|V|$ , which completes the proof of Claim 38.

## **B** On Different Complexity Measures for DNF $\circ$ MOD $_p$ Circuits

In this section, we provide an example of the robustness of our arguments with respect to variations of the complexity measure. Let  $p \geq 2$  be a fixed prime. We sketch the proof of a hardness result for a variant of the (DNF  $\circ$  MOD $_p$ )-MCSP\* problem, described as follows. We consider layered OR  $\circ$  AND  $\circ$  MOD $_p$  formulas<sup>7</sup> over  $\mathbb{Z}_p^n$ , and measure complexity by the total

<sup>7</sup> Recall that in a formula every non-input gate has fan-out one.



number of (non-input) gates in the formula.<sup>8</sup> A bit more precisely, we adapt the proof of Theorem 8 from Section 3.1, and show that this problem is also NP-hard under randomized reductions.

Since  $\mathbb{Z}_p^t$  is a vector space over the field  $\mathbb{Z}_p$ , we can define the dimension of an affine subspace: For a linear subspace  $H \subseteq \mathbb{Z}_p^t$ , let  $\dim(H)$  denote the dimension of  $H$ , and let  $\text{codim}(H) := \dim(H^\perp) = t - \dim(H)$ ; then, for any  $a \in \mathbb{Z}_p^t$ , define the dimension of an affine subspace  $H + a$  as  $\dim(H + a) := \dim(H)$ , and  $\text{codim}(H + a) := \text{codim}(H)$ . Observe that this notion is well-defined. Using dimension, we can characterize the number of gates in  $\text{AND} \circ \text{MOD}_p$  formulas.

► **Lemma 42.** *Let  $A$  be an affine subspace of  $\mathbb{Z}_p^t$ . Then, the minimum number of gates in any layered  $\text{AND} \circ \text{MOD}_p$  formula accepting  $A$  is exactly  $1 + \text{codim}(A)$ .*

**Proof Sketch.** As in the proof of Lemma 2, a layered  $\text{AND} \circ \text{MOD}_p$  formula  $C$  with  $1 + s$  gates accepts the set  $A = C^{-1}(1)$  of solutions of  $s$  linear equations over  $\text{MOD}_p$ . Let  $B \in \mathbb{Z}_p^{s \times t}$  be the matrix that defines these linear equations. Then, we have  $\dim \ker(B) = \dim(A)$ , and by the rank-nullity theorem, we obtain  $\text{codim}(A) = t - \dim(A) = t - \dim \ker(B) = \text{rank}(B) \leq s$ .

Conversely, let  $A = H + a$  for some linear subspace  $H$  and some  $a \in \mathbb{Z}_p^t$ , and let  $\gamma_1, \dots, \gamma_s$  be a basis of  $H^\perp$ , where  $s := \text{codim}(H)$ . Then, using orthogonal complements, it is easy to check that  $x \in A$  if and only if  $\langle \gamma_i, x \rangle = \langle \gamma_i, a \rangle$  for all  $i \in [s]$ . The latter condition can be written as an  $\text{AND} \circ \text{MOD}_p$  layered formula with  $1 + s$  gates. ◀

As a corollary, for any *optimal* layered  $(\text{DNF} \circ \text{MOD}_p)$ -formula  $C = \bigvee_{k=1}^K C_k$  for a function  $f: \mathbb{Z}_p^n \rightarrow \{0, 1\}$ , where  $C_k$  is an  $\text{AND} \circ \text{MOD}_p$  circuit for each  $k \in [K]$ , the total number of gates in the formula is precisely  $1 + K + \sum_{k=1}^K \text{codim}(C_k^{-1}(1))$ .

For convenience, given a function  $f: \mathbb{Z}_p^t \rightarrow \{0, 1, *\}$ , let  $\text{size}(f)$  denote the complexity of  $f$  according to our size measure. Now let us revise the proof of Theorem 8. Given an instance  $\mathcal{S} \subseteq \binom{[n]}{\leq r}$  of the  $r$ -bounded set cover instance, we construct a function  $f: \mathbb{Z}_p^t \rightarrow \{0, 1, *\}$  in exactly the same way. Below we adapt the corresponding claims from Section 3.1. Then we employ the new claims to argue that the NP-hardness result still holds.

► **Claim 43** (Adaptation of Claim 9). *Assume that  $\mathcal{S}$  has a set cover of size  $K$ . Then  $\text{size}(f) \leq (t + 1)K + 1$ .*

**Proof.** Let  $\mathcal{C} \subseteq \mathcal{S}$  be a set cover of size  $K$ . For each  $S \in \mathcal{C}$ , let  $C_S$  be an  $\text{AND} \circ \text{MOD}_p$  circuit over  $\mathbb{Z}_p^t$  that accepts  $\text{span}(v^S)$ . Define a  $\text{DNF} \circ \text{MOD}_p$  circuit  $C := \bigvee_{S \in \mathcal{C}} C_S$ . Then the circuit size of  $C$  is  $1 + K + \sum_{i=1}^K \text{codim}(C_S^{-1}(1))$ , which is obviously at most  $1 + K(t + 1)$ . ◀

► **Claim 44** (Adaptation of Claim 13). *Let  $(v^i)_{i \in [n]}$  be nice, and  $s := \text{size}(f)$ . Then  $\mathcal{S}$  has a set cover of size  $2(s - 1)/(t - r - (\log |\mathcal{S}|/\log p) + 1)$ .*

**Proof.** Let  $C = \bigvee_{k=1}^K C_k$  be an optimal  $\text{DNF} \circ \text{MOD}_p$  layered formula of size  $s$  computing  $f$ . Then, as discussed above, we have  $s = 1 + K + \sum_{k=1}^K \text{codim}(C_k^{-1}(1))$ . On the other hand, the same analysis from Claim 13 shows that  $\mathcal{S}$  has a set cover of size  $\leq 2K$ . It thus remains to give an upper bound on  $K$ .

Since  $C$  computes  $f$ , we have  $C_k^{-1}(1) \subseteq C^{-1}(1) \subseteq f^{-1}(\{1, *\}) = \bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ . By counting the number of elements in  $C_k^{-1}(1)$  and  $\bigcup_{S \in \mathcal{S}} \text{span}(v^S)$ , we obtain  $p^{\dim(C_k^{-1}(1))} \leq |\mathcal{S}| \cdot p^r$ . Hence, we have  $\text{codim}(C_k^{-1}(1)) \geq t - r - \log |\mathcal{S}|/\log p$ ; therefore,

<sup>8</sup> Under our notion of layered formulas, an  $(\text{AND} \circ \text{MOD}_p)$ -circuit with a single  $\text{MOD}_p$  gate has size 2. While this is convenient for the exposition, it is not particularly important for the result.

$$s \geq 1 + K + \sum_{k=1}^K \text{codim}(C_k^{-1}(1)) \geq 1 + K + K(t - r - \log |\mathcal{S}| / \log p),$$

which implies  $K \leq (s - 1) / (t - r - (\log |\mathcal{S}| / \log p) + 1)$ .  $\blacktriangleleft$

Let  $K$  be the minimum size of a cover for  $\mathcal{S}$ . By the claims above, we have  $\text{size}(f) \lesssim tK$  and  $K \lesssim 2\text{size}(f)/t$ , because  $t$  can be taken large enough compared to the other relevant parameters; hence  $\text{size}(f)/t$  roughly gives us a 2-factor approximation. More precisely, we have  $\text{size}(f) \leq (t+1)K + 1 \leq 2(t+1)K$ , and  $K \leq 2(\text{size}(f) - 1) / ((t+1)/2) \leq 4\text{size}(f) / (t+1)$  for any  $t \geq 2r + 2\log |\mathcal{S}| / \log p - 1$ . That is, the set cover size  $K$  satisfies

$$\frac{\text{size}(f)}{2(t+1)} \leq K \leq \frac{4\text{size}(f)}{t+1},$$

which gives an 8-factor approximation of  $K$ . Since we can take  $r$  to be a sufficiently large constant in Theorem 5, the result holds.

### C A Hardness of Approximation Result for $(\text{DNF} \circ \text{MOD}_m)$ -MCSP

The reduction from  $(\text{DNF} \circ \text{MOD}_m)$ -MCSP\* to  $(\text{DNF} \circ \text{MOD}_m)$ -MCSP presented in Section 3 is not *approximation-preserving*: given a partial function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$ , it produces a total function  $g: \mathbb{Z}_m^{O(t \log m)} \rightarrow \{0, 1\}$  such that  $\text{DNF}_{\text{MOD}_m}(g) = \text{DNF}_{\text{MOD}_m}(f) + |f^{-1}(*)|$ . The reduction introduces an additive term  $|f^{-1}(*)|$ , and hence a (multiplicative) approximation of  $\text{DNF}_{\text{MOD}_m}(g)$  does not give a good approximation of  $\text{DNF}_{\text{MOD}_m}(f)$ . In order to fix this situation, we give an approximation-preserving reduction. Our approach is inspired by a reduction described in [5].

► **Theorem 45** (Approximation-preserving version of Corollary 27). *There is a polynomial-time algorithm that, given the truth table of a partial function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1, *\}$ , produces the truth table of a total function  $g: \mathbb{Z}_m^{2t+2s} \rightarrow \{0, 1\}$  such that*

$$\text{DNF}_{\text{MOD}_m}(g) = |f^{-1}(*)| \cdot (\text{DNF}_{\text{MOD}_m}(f) + 1),$$

where  $s := \lceil (6t + 4) \log m + 2 \rceil$ .

**Proof.** The idea of the proof is to amplify the circuit size for  $f$ ; that is, we would like to force any circuit  $C$  computing  $g$  to also compute sub-functions corresponding to  $|f^{-1}(*)|$  copies of  $f$ .

We can amplify the circuit size as follows. Let  $(L_x)_{x \in f^{-1}(*)}$  be a scattered collection of linear subspaces of  $\mathbb{Z}_m^s$ . Define a function  $g'$  by  $g'(x, z, w) := f(x)$  if  $z \in f^{-1}(*)$  and  $w \in L_z$ ; otherwise  $g'(x, z, w) := 0$ . Then, under an appropriate choice of parameters, it can be shown that  $\text{DNF}_{\text{MOD}_m}(g') = |f^{-1}(*)| \cdot \text{DNF}_{\text{MOD}_m}(f)$ . By combining an analogous reduction and the idea behind the proof of Theorem 16, we can obtain a total function  $g$  such that  $\text{DNF}_{\text{MOD}_m}(g) = \text{DNF}_{\text{MOD}_m}(g') + |f^{-1}(*)| = |f^{-1}(*)| \cdot (\text{DNF}_{\text{MOD}_m}(f) + 1)$ .<sup>9</sup> Details follow.

<sup>9</sup> A black-box application of Corollary 27 produces a function  $g$  such that  $\text{DNF}_{\text{MOD}_m}(g) = \text{DNF}_{\text{MOD}_m}(g') + |g'^{-1}(*)|$ , which is not sufficient for our purpose because  $|g'^{-1}(*)|$  is larger than  $|f^{-1}(*)|$ .



We first obtain a scattered collection  $(L_x)_{x \in f^{-1}(*)}$  of  $r$ -dimensional linear subspaces of  $\mathbb{Z}_m^s$  by using Theorem 25 for  $r := 2t + 2$ . Then we define  $g: \mathbb{Z}_m^{2t+2s} \rightarrow \{0, 1\}$  as

$$g(x, y, z, w) := \begin{cases} f(x) & \text{(if } f(x) \in \{0, 1\} \text{ and } y = 0^s \text{ and } f(z) = * \text{ and } w \in L_z) \\ 1 & \text{(if } f(x) = * \text{ and } y \in L_x) \\ 0 & \text{(otherwise)} \end{cases}$$

for any  $((x, y), (z, w)) \in (\mathbb{Z}_m^s \times \mathbb{Z}_m^t)^2$ .

► **Claim 46** (Analogue of Claim 17).  $\text{DNF}_{\text{MOD}_m}(g) \leq |f^{-1}(*)| \cdot (\text{DNF}_{\text{MOD}_m}(f) + 1)$ .

**Proof.** Suppose that a  $\text{DNF} \circ \text{MOD}_m$  circuit  $C = \bigvee_{k=1}^K C_k$  computes  $f$ . For each  $x^* \in f^{-1}(*)$ , take an  $\text{AND} \circ \text{MOD}_m$  circuit  $C_{x^*}$  accepting  $\{x^*\} \times L_{x^*}$  (by Lemma 2). Define

$$C'(x, y, z, w) := \bigvee_{z^* \in f^{-1}(*)} \bigvee_{k=1}^K (C_k(x) \wedge y_1 = 0 \wedge \dots \wedge y_s = 0 \wedge C_{z^*}(z, w)) \vee \bigvee_{x^* \in f^{-1}(*)} C_{x^*}(x, y).$$

It is easy to see that  $C'$  computes  $g$ . ◀

The rest of the proof is devoted to the reverse direction.

► **Claim 47** (Analogue of Claim 20).  $\text{DNF}_{\text{MOD}_m}(g) \geq |f^{-1}(*)| \cdot (\text{DNF}_{\text{MOD}_m}(f) + 1)$ .

Let  $C = \bigvee_{k=1}^K C_k$  be a minimum  $\text{DNF} \circ \text{MOD}_m$  circuit computing  $g$ . In particular,  $K = \text{DNF}_{\text{MOD}_m}(g) \leq |f^{-1}(*)| \cdot (\text{DNF}_{\text{MOD}_m}(f) + 1) \leq m^{2t+1}$ . For each  $x \in f^{-1}(*)$ , let  $l(x) \in [K]$  be one of the indices such that  $|C_{l(x)}^{-1}(1) \cap (\{x\} \times L_x \times \mathbb{Z}_m^{t+s})|$  is maximized. Since  $\bigcup_{k \in [K]} C_k^{-1}(1) \supseteq \{x\} \times L_x \times \mathbb{Z}_m^{t+s}$ , there are at least  $|L_x| \cdot m^{t+s} / K \geq m^{r+t+s} / m^{2t+1} \geq 2$  points in the set  $C_{l(x)}^{-1}(1) \cap (\{x\} \times L_x \times \mathbb{Z}_m^{t+s})$ .

Define  $T_0 := \{C_{l(x)} \mid f(x) = *\}$ . For each  $z \in f^{-1}(*)$ , let  $T_z$  be the set of all  $C_k$  such that  $k \in [K]$  and  $C_k$  accepts at least 2 elements from  $\{(x, 0^s, z)\} \times L_z$  for some  $x \in f^{-1}(1)$ . We will show that the sets  $T_0, \{T_z\}_{z \in f^{-1}(*)}$  are pairwise disjoint, and hence  $K \geq |T_0| + \sum_{z \in f^{-1}(*)} |T_z|$ . We will also prove that  $|T_0| = |f^{-1}(*)|$  and  $|T_z| \geq \text{DNF}_{\text{MOD}_m}(f)$ , which completes the proof.

► **Claim 48.**  $l: f^{-1}(*) \rightarrow [K]$  is injective (hence  $|T_0| = |f^{-1}(*)|$ ).

► **Claim 49.**  $T_0 \cap T_z = \emptyset$  for any  $z \in f^{-1}(*)$ .

Since the proofs of these claims are essentially the same as in Claims 21 and 22, respectively (except that we have extra coordinates taking values in  $\mathbb{Z}_m^t \times \mathbb{Z}_m^s$ ), we omit them.

► **Claim 50.**  $T_{z_1} \cap T_{z_2} = \emptyset$  for any distinct elements  $z_1, z_2 \in f^{-1}(*)$ .

**Proof.** The proof is basically the argument from Claim 21. For completeness, we briefly repeat it here. Towards a contradiction, assume that there exists a circuit  $C_k$  in  $T_{z_1} \cap T_{z_2}$ . By the definition of  $T_{z_1}$  and  $T_{z_2}$ , there exist elements  $x_1, x_2 \in f^{-1}(1)$ ,  $a \neq b \in L_{z_1}$ , and  $c \in L_{z_2}$  such that  $C_k(x_1, 0^s, z_1, a) = C_k(x_1, 0^s, z_1, b) = C_k(x_2, 0^s, z_2, c) = 1$ . Since  $C_k^{-1}(1)$  is an affine subspace, we have  $(x_1, 0^s, z_1, a) - (x_1, 0^s, z_1, b) + (x_2, 0^s, z_2, c) = (x_2, 0^s, z_2, a - b + c) \in C_k^{-1}(1)$ . Since  $C_k^{-1}(1) \cap (\{(x_2, 0^s, z_2)\} \times \mathbb{Z}_m^s) \subseteq \{(x_2, 0^s, z_2)\} \times L_{z_2}$ , we get  $a - b + c \in L_{z_2}$ . However, given that  $c \in L_{z_2}$ , we obtain  $0^s \neq a - b \in L_{z_1} \cap L_{z_2}$ , which contradicts  $L_{z_1} \cap L_{z_2} = \{0^s\}$ . ◀

Fix any  $z \in f^{-1}(*)$ . For each  $C_k \in T_z$ , define an  $\text{AND} \circ \text{MOD}_m$  circuit  $C'_k$  so that  $C'_k^{-1}(1) = \{x \in \mathbb{Z}_m^t \mid C_k(x, 0^s, z, w) = 1 \text{ for some } w \in \mathbb{Z}_m^s\}$ . (Note that a projection of an affine subspace  $C_k^{-1}(1)$  is again an affine subspace because a projection is a homomorphism.) Now define  $C_z := \bigvee_{C_k \in T_z} C'_k$ .

► **Claim 51.**  $C_z$  computes  $f$  for any  $z \in f^{-1}(*)$ . (In particular,  $|T_z| \geq \text{DNF}_{\text{MOD}_m}(f)$ .)

**Proof.** Fix any  $x \in f^{-1}(1)$ . Since  $\{(x, 0^s, z)\} \times L_z$  is covered by  $\bigcup_{k \in [K]} C_k^{-1}(1)$ , and  $|L_z| = m^r$ ,  $K \leq m^{2t+1}$ , and  $r = 2t + 2$ , there exists  $k \in [K]$  such that there are at least 2 elements in  $(\{(x, 0^s, z)\} \times L_z) \cap C_k^{-1}(1)$ ; hence, by the definition of  $T_z$ , we have  $C_k \in T_z$ . Moreover,  $C'_k(x) = 1$  by the definition of  $C'_k$ ; thus  $C_z(x) = \bigvee_{C_k \in T_z} C'_k(x) = 1$ .

Now fix any  $x \in f^{-1}(0)$ . Since  $g(x, 0^s, z, w) = 0$  for every  $w \in \mathbb{Z}_m^s$ , we get  $C_k(x, 0^s, z, w) = 0$  for any  $C_k \in T_z$ ; thus  $C'_k(x) = 0$ , which implies that  $C_z(x) = 0$ . ◀

Combining the claims above, we obtain

$$\text{DNF}_{\text{MOD}_m}(g) = K \geq |T_0| + \sum_{z \in f^{-1}(*)} |T_z| \geq |f^{-1}(*)| \cdot (\text{DNF}_{\text{MOD}_m}(f) + 1).$$

This completes the proof of Theorem 45. ◀

We can then establish a hardness of approximation result for computing  $\text{DNF}_{\text{MOD}_m}(f)$ . For a function  $f: \mathbb{Z}_m^t \rightarrow \{0, 1\}$ , define  $|f| := m^t$ , which is the number of entries in the truth table of a function  $f$ .

► **Theorem 52.** *There exists a constant  $c > 0$  such that if there is a quasipolynomial-time algorithm which approximates  $\text{DNF}_{\text{MOD}_m}(f)$  to within a factor of  $c \log \log |f|$ , then  $\text{NP} \subseteq \text{DTIME}(2^{(\log n)^{O(1)}})$ .*

**Proof.** As noted by Trevisan [42], by choosing the parameters of Feige's reduction [17], one can obtain hardness of approximation results for the  $r$ -bounded set cover problem. While Trevisan only analyzed the case when  $r$  is constant (cf. Theorem 5), a similar analysis<sup>10</sup> shows that it is NP-hard (under quasipolynomial-time many-one reductions) to approximate the  $r(n)$ -bounded set cover problem on  $n$  points within a factor of  $\gamma \log r(n)$  ( $= \gamma \log \log n$ ) for  $r(n) := \log n$  and some small constant  $\gamma > 0$ .

Suppose that  $\text{DNF}_{\text{MOD}_m}(g)$  can be approximated to within a factor of  $(\gamma/6) \log \log |g|$  by an algorithm  $A$ , where  $g: \mathbb{Z}_m^t \rightarrow \{0, 1\}$  is a total function. We show below that if  $A$  runs in quasipolynomial time, then  $\text{NP} \subseteq \text{DTIME}(2^{(\log n)^{O(1)}})$ .

First, note that in order to conclude this it is enough to describe a quasipolynomial-time algorithm  $B$  that approximates  $r$ -Bounded Set Cover to within a factor of  $\gamma \log r(n)$  for  $r(n) = \log n$ . Let  $([n], \mathcal{S})$  be an instance of the  $r$ -Bounded Set Cover Problem. Algorithm  $B$  applies the deterministic  $n^{O(r(n))}$ -time reduction provided by Corollary 29 to produce a partial Boolean function  $f: \mathbb{Z}_m^{O(r \log n)} \rightarrow \{0, 1, *\}$ . It then invokes the deterministic reduction from Theorem 45 to construct from  $f$  a total function  $g: \mathbb{Z}_m^{O(r \log n)} \rightarrow \{0, 1\}$ . Finally,  $B$  uses the approximation algorithm  $A$  to compute a  $(\gamma/6) \log \log |g|$  approximation to  $\text{DNF}_{\text{MOD}_m}(g)$ . Let  $\tilde{g} \in \mathbb{N}$  be the value output by  $A$ . Algorithm  $B$  outputs  $\tilde{K} := 2\tilde{g}/|f^{-1}(*)|$ .

Note that  $B$  runs in quasipolynomial time under our assumptions. It remains to show that it approximates the solution of the original set cover problem within a factor of  $\gamma \log \log n$ . Let  $K$  be the cost of an optimal solution to the initial set cover instance. Recall that

<sup>10</sup> Specifically, for the parameters and notation in [17], given a 3CNF-5 formula on  $n$  variables, let  $k$  be a sufficiently large constant,  $m := \sqrt{\log n}$ , and  $\ell := c \log \log m$  for a large constant  $c$ . Then the output of Feige's reduction is an instance of the set cover problem on  $N$  ( $= m(5n)^\ell$ ) points such that each set is of size at most  $m2^{O(\ell)} \leq r(N) = \log N$ , and the gap between yes instances and no instances is  $(1 - \frac{4}{k}) \ln m = \Omega(\log \log N)$ .

$2\text{DNF}_{\text{MOD}_m}(f)$  is a 2-factor approximation for  $K$ ; that is,  $K \leq 2 \cdot \text{DNF}_{\text{MOD}_m}(f) \leq 2K$ . On the other hand, the guarantees of the algorithm  $A$  imply that

$$\text{DNF}_{\text{MOD}_m}(g) \leq \tilde{g} \leq \text{DNF}_{\text{MOD}_m}(g) \cdot (\gamma/6) \log \log |g|.$$

Since  $\text{DNF}_{\text{MOD}_m}(g) = |f^{-1}(*)| \cdot (\text{DNF}_{\text{MOD}_m}(f) + 1)$ , we get

$$K \leq \frac{2\tilde{g}}{|f^{-1}(*)|} \leq (\gamma/6) \log \log |g| \cdot (K + 1)$$

Therefore, for large enough  $n$  and on non-trivial instances (i.e.  $K \geq 1$ ), the value  $\tilde{K}$  output by  $B$  approximates  $K$  to within a factor of  $2 \cdot (\gamma/6) \log \log |g| \leq (\gamma/3) \cdot (\log r(n) + \log \log n + O(1)) \leq (\gamma/3) \cdot 3 \log \log n$ . ◀

Finally, we note that when  $m$  is prime, it is possible to design a quasipolynomial-time approximation algorithm for  $\text{DNF}_{\text{MOD}_m}(f)$  with an approximation factor of  $O(\log |f|)$ .

► **Theorem 53.** *Let  $p$  be a prime number. There is a quasipolynomial-time algorithm which approximates  $\text{DNF}_{\text{MOD}_p}(f)$  to within a factor of  $\ln |f|$ .*

**Proof.** Let  $|f| = p^t$  be the number of entries in the truth table of  $f$ , the input function. By the results of Section 2.1, computing  $\text{DNF}_{\text{MOD}_p}(f)$  is equivalent to solving a set cover instance. Recall that set cover admits a polynomial-time approximation algorithm that achieves an approximation factor of  $\ln N$  on instances over a universe of size  $N$  (cf. [40]). Consequently, in order to prove the result it is enough to verify that computing  $\text{DNF}_{\text{MOD}_p}(f)$  reduces to a set cover instance with domain size  $N_f := |f^{-1}(1)| \leq |f|$  and of size at most quasipolynomial in  $|f|$ .

Indeed, for a non-zero function  $f: \mathbb{Z}_p^t \rightarrow \{0, 1\}$ ,  $\text{DNF}_{\text{MOD}_p}(f)$  is exactly the minimum number of affine subspaces that cover  $f^{-1}(1)$ . Therefore, by relabelling elements, computing  $\text{DNF}_{\text{MOD}_p}(f)$  reduces to a set cover instance  $([N_f], \mathcal{S}_f)$ , where a set  $S \in \mathcal{S}_f$  if and only if  $S$  viewed as a subset of  $\mathbb{Z}_p^t$  is an affine subspace contained in  $f^{-1}(1)$ . Each such affine subspace has dimension at most  $t$ , and can be explicitly described by a basis  $v_1, \dots, v_\ell \in \mathbb{Z}_p^t$ , where  $\ell \leq t$ , and a vector  $b \in \mathbb{Z}_p^t$ . Hence there are at most  $p^{O(t^2)}$  such spaces, and consequently,  $|\mathcal{S}_f| \leq p^{O(t^2)}$ . In other words, we get a set cover instance over a ground set of size  $\leq |f|$ , and this instance contains at most  $|f|^{O(\log |f|)}$  sets.

Finally, since the sets in  $\mathcal{S}_f$  can be generated in time at most  $|f|^{O(\log |f|)}$ , and the set cover approximation algorithm runs in time polynomial in its input length, the result holds. ◀




# Limits on Representing Boolean Functions by Linear Combinations of Simple Functions: Thresholds, ReLUs, and Low-Degree Polynomials

Richard Ryan Williams<sup>1</sup>

EECS and CSAIL, MIT, 32 Vassar St., Cambridge MA, USA

rrw@mit.edu

 <https://orcid.org/0000-0003-2326-2233>

---

## Abstract

We consider the problem of representing Boolean functions exactly by “sparse” linear combinations (over  $\mathbb{R}$ ) of functions from some “simple” class  $\mathcal{C}$ . In particular, given  $\mathcal{C}$  we are interested in finding low-complexity functions lacking sparse representations. When  $\mathcal{C}$  forms a basis for the space of Boolean functions (e.g., the set of PARITY functions or the set of conjunctions) this sort of problem has a well-understood answer; the problem becomes interesting when  $\mathcal{C}$  is “overcomplete” and the set of functions is not linearly independent. We focus on the cases where  $\mathcal{C}$  is the set of linear threshold functions, the set of rectified linear units (ReLUs), and the set of low-degree polynomials over a finite field, all of which are well-studied in different contexts.

We provide generic tools for proving lower bounds on representations of this kind. Applying these, we give several new lower bounds for “semi-explicit” Boolean functions. Let  $\alpha(n)$  be an unbounded function such that  $n^{\alpha(n)}$  is time constructible (e.g.  $\alpha(n) = \log^*(n)$ ). We show:

- Functions in  $\text{NTIME}[n^{\alpha(n)}]$  that require super-polynomially many linear threshold functions to represent (depth-two neural networks with sign activation function, a special case of depth-two threshold circuit lower bounds).
- Functions in  $\text{NTIME}[n^{\alpha(n)}]$  that require super-polynomially many ReLU gates to represent (depth-two neural networks with ReLU activation function).
- Functions in  $\text{NTIME}[n^{\alpha(n)}]$  that require super-polynomially many  $O(1)$ -degree  $\mathbb{F}_p$ -polynomials to represent exactly, for every prime  $p$  (related to problems regarding Higher-Order “Uncertainty Principles”). We also obtain a function in  $\text{E}^{\text{NP}}$  requiring  $2^{\Omega(n)}$  linear combinations.
- Functions in  $\text{NTIME}[n^{\text{poly}(\log n)}]$  that require super-polynomially many  $\text{ACC} \circ \text{THR}$  circuits to represent exactly (further generalizing the recent lower bounds of Murray and the author).

We also obtain “fixed-polynomial” lower bounds for functions in  $\text{NP}$ , for the first three representation classes. All our lower bounds are obtained via algorithms for *analyzing* linear combinations of simple functions in the above scenarios, in ways which substantially beat exhaustive search.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Circuit complexity, Computer systems organization  $\rightarrow$  Neural networks

**Keywords and phrases** linear threshold functions, lower bounds, neural networks, low-degree polynomials

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.6

**Acknowledgements** I thank Lijie Chen, Pooya Hatami, Adam Klivans, and Shachar Lovett for useful discussions on the topics of this paper. In particular, I am grateful to Shachar for noticing

---

<sup>1</sup> Supported by NSF CCF-1553288. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.



a gap in a lemma in an earlier version of this paper. I am also grateful to Brynmor Chapman for his proofreading, and patience with my explanations regarding this paper. Thanks also to the CCC'18 reviewers for their comments and corrections.

## 1 Introduction

Given  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and a class  $\mathcal{C}$  of “simple” functions, when can  $f$  be represented exactly as a *short*  $\mathbb{R}$ -linear combination of functions from  $\mathcal{C}$ ? When  $\mathcal{C}$  forms a basis for  $B_n$  (the set of all Boolean functions on  $n$  inputs) the question has a unique answer that is generally easy to obtain, by analyzing the appropriate linear system (the cases where  $\mathcal{C}$  is the set of all parity functions or the set of all conjunctions are canonical examples). For  $|\mathcal{C}| \gg 2^n$ , the situation becomes much more interesting, as there can be many possible representations. The general problem of understanding which functions do and do not have sparse representations for simple  $\mathcal{C}$  arises in many different mathematical topics. Three relevant to TCS are depth-two threshold circuits, depth-two neural networks with various activation functions, and higher-order Fourier analysis. We use the notation

$$\text{SUM} \circ \mathcal{C}$$

to denote the class of  $\mathbb{R}$ -linear combinations of  $\mathcal{C}$ -functions; for example,  $\text{SUM} \circ \text{MOD2}$  denotes  $\mathbb{R}$ -linear combinations of PARITY functions. The relevant complexity measure for a “circuit” in  $\text{SUM} \circ \mathcal{C}$  is the fan-in of the SUM gate, which we call the *sparsity* of the circuit.

### Sums of Threshold Circuits

Let  $\text{SUM} \circ \text{THR}$  be linear combinations of linear threshold functions (LTFs).<sup>2</sup> As there are  $2^{\Theta(n^2)}$   $n$ -variate threshold functions [55], a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  has *many* possible representations as a  $\text{SUM} \circ \text{THR}$ . Such circuits are also known in the machine learning literature as *depth-two neural networks with sign activation functions*.

In 1994, Roychowdhury, Orlitsky, and Siu [38] noted that no interesting size lower bounds were known for computing Boolean functions with  $\text{SUM} \circ \text{THR}$  circuits (beyond the few that are/were known for  $\text{THR} \circ \text{THR}$  [22, 38, 28, 14, 43, 2]). The problem was raised again more recently in CCC'10 by Hansen and Podolskii [23]. In particular, the following remains largely unanswered:

**Problem:** *Find an explicit  $f : \{0, 1\}^* \rightarrow \{0, 1\}$  without polynomially-sparse  $\text{SUM} \circ \text{THR}$ , i.e., every linear combination of LTFs computing  $f$  on  $n$ -bit inputs needs  $n^{\omega(1)}$  LTFs, for infinitely many  $n$ .*

Because of prior lower bounds in weaker settings (such as majority-of-majority [22] and majority-of-thresholds [36]), it is natural to think that correlation bounds against linear threshold functions should help.<sup>3</sup> Correlation bounds do imply lower bounds for  $\text{SUM} \circ \text{THR}$ , but only when the weights in the linear combination are not too large (i.e., the weights must be in  $[-2^{\delta n}, 2^{\delta n}]$  for small  $\delta < 1$ ). However, if arbitrary weights are allowed, interesting lower bounds on  $\text{SUM} \circ \text{THR}$  (beyond  $\Omega(n^{2.5})$  wires [28]) were open, to the best of our knowledge. In Section 4, we prove arbitrary polynomial lower bounds for NP functions:

<sup>2</sup> From here on, “linear combination” means “ $\mathbb{R}$ -linear combination”, unless otherwise specified.

<sup>3</sup> That is, one wants to show that a function cannot be  $(1/2 + \varepsilon(n))$ -approximated by a linear threshold function, for the tiniest  $\varepsilon(n) > 0$  possible.

► **Theorem 1.** *For all  $k$ , there is an  $f_k \in \text{NP}$  without  $\text{SUM} \circ \text{THR}$  circuits of  $n^k$  sparsity. Furthermore, for every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time constructible, there is a function in  $\text{NTIME}[n^{\alpha(n)}]$  that does not have  $\text{SUM} \circ \text{THR}$  circuits of polynomial sparsity.*

Note that for arbitrary circuits (even for  $\text{THR} \circ \text{THR}$  circuits) the best known complexity for such functions without  $n^k$ -size circuits (for fixed  $k$ ) is  $\text{MA}/1$  ([40]) and  $S_2^p$ .

### Sums of ReLU Gates

A ReLU (rectified linear unit) gate is a function  $f : \{0, 1\}^t \rightarrow \mathbb{R}^+$  such that there is a vector  $w \in \mathbb{R}^t$  and scalar  $a \in \mathbb{R}$  such that for all  $x$ ,

$$f(x) = \max\{0, \langle x, w \rangle + a\}.$$

It is important to note that ReLU gates might not be Boolean-valued, but they must output non-negative numbers on all Boolean inputs. Linear combinations of ReLU gates are also known as *depth-two neural networks with ReLU activation functions*, and they are intensely studied in machine learning. Several lower bounds for Sums-of-ReLU functions (which for consistency we call  $\text{SUM} \circ \text{ReLU}$ ) have recently been shown for functions with *real-valued* inputs and outputs (examples include [16, 44, 3, 15, 39]) but none of the methods extend to Boolean functions, to the best of our knowledge. Recently, Mukherjee and Basu [33] have proved  $\Omega(n^{1-\delta})$ -gate lower bounds for  $\text{SUM} \circ \text{ReLU}$  circuits computing the Andreev function, extending ideas in [28, 13].

Observing that for  $|\langle x, w \rangle| \geq 1$  we have

$$\max\{0, \langle x, w \rangle + 1\} - \max\{0, \langle x, w \rangle\} = \text{sign}(\langle x, w \rangle),$$

it follows that every  $\text{SUM} \circ \text{THR}$  circuit can be simulated by a  $\text{SUM} \circ \text{ReLU}$  circuit with only a doubling of the sparsity. In Section 5 we extend our lower bounds to Sums-of-ReLU circuits:

► **Theorem 2.** *For all  $k$ , there is an  $f_k \in \text{NP}$  without  $\text{SUM} \circ \text{ReLU}$  circuits of  $n^k$  sparsity. Furthermore, for every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time constructible, there is a function in  $\text{NTIME}[n^{\alpha(n)}]$  that does not have  $\text{SUM} \circ \text{ReLU}$  circuits of polynomial sparsity.*

### Representing Boolean Functions With Higher-Order Polynomials

Higher-order Fourier analysis of Boolean functions deals with representing Boolean functions by  $\mathbb{R}$ -linear combinations of  $\mathbb{F}_2$ -polynomials of degree higher than one (see [25] for a survey of some applications in CS theory). The question of which (if any) explicit functions lack *sparse* representations, even for degree-two polynomials, has been wide open. Letting  $\text{MOD}_2$  be the class of parity functions, this question asks to find lower bounds for  $\text{SUM} \circ \text{MOD}_2 \circ \text{AND}_2$  circuits (in our notation,  $\text{AND}_k$  denotes ANDs of fan-in at most  $k$ ). Such lower bound problems appear much more difficult than the degree-one case of  $\text{SUM} \circ \text{MOD}_2$ . Even understanding the sparsity of the AND function in the quadratic (and in general, degree- $O(1)$ ) setting is a prominent open problem:

► **Hypothesis 3** (Quadratic Uncertainty Principle [17]). *There is an  $\varepsilon > 0$  such that the AND function on  $n$  variables does not have  $\text{SUM} \circ \text{MOD}_2 \circ \text{AND}_2$  circuits of  $2^{\varepsilon n}$  sparsity.*

Although it is believed that AND needs exponential sparsity, to our knowledge the *only* lower bound known for an explicit function in  $\text{SUM} \circ \text{MOD}_2 \circ \text{AND}_2$  was  $\Omega(n)$ -sparsity. For completeness we include a proof provided to us by Lovett [30]) in Appendix A. Again, when

the weights in the linear combination are required to be small (magnitudes are  $2^{\varepsilon n}$  for small  $\varepsilon > 0$ ), correlation bounds yield some results: one example (among many) is the work of Green [20] showing that a majority vote of quadratic  $\mathbb{F}_3$ -polynomials needs  $2^{\Omega(n)}$  polynomials to compute PARITY. (Other works in this vein include [24, 10, 9, 19]; see Viola [49] for a survey.) However, for arbitrary weights, no non-trivial lower bounds have been reported (to our knowledge).

In Section 6, we prove polynomial sparsity lower bounds for Boolean functions in NP and  $2^{\Omega(n)}$ -size lower bounds for  $E^{\text{NP}}$ , against linear combinations of polynomials over any prime field with any constant degree:

► **Theorem 4.** *For every integer  $k, d \geq 1$  and prime  $p$ , there is an  $f_k \in \text{NP}$  without  $\text{SUM} \circ \text{MOD}_p \circ \text{AND}_d$  circuits of  $n^k$  sparsity. Furthermore, for every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time constructible, there is a function in  $\text{NTIME}[n^{\alpha(n)}]$  that does not have  $\text{SUM} \circ \text{MOD}_p \circ \text{AND}_d$  circuits of polynomial sparsity.*

► **Theorem 5.** *For every  $d \geq 1$  and prime  $p$ , there is an  $\alpha > 0$  and an  $f \in E^{\text{NP}}$  without  $\text{SUM} \circ \text{MOD}_p \circ \text{AND}_d$  circuits of  $2^{\alpha n}$  sparsity.*

Note the “smallest” known complexity class for a function lacking  $2^{\Omega(n)}$ -size circuits is  $E^{\Sigma_2^{\text{P}}}$  [32], and it is a longstanding open problem to reduce the complexity class for such a function, even against depth-3  $\text{AC}^0$  circuits.

## 1.1 Intuition

Here we give an overview of some of the ideas used to prove the lower bounds in this work. The lower bounds of this paper follow the high-level strategy of proving circuit lower bounds by designing circuit-analysis (satisfiability) algorithms [51, 53, 52]. However, in this work we must execute this strategy differently. All previous lower bounds proved in this framework utilize the “polynomial method” from circuit complexity in various ways (representing a circuit by a low-degree polynomial of some kind), combined with fast matrix multiplication and/or fast polynomial evaluation. These approaches do not seem to work for solving SAT on linear combinations of thresholds, low-degree polynomials, or ReLU gates. For example, we do not know how to get a sparse (probabilistic or approximate) polynomial (over any field) for computing an OR of many  $\text{SUM} \circ \text{THR}$ s, and it is likely that any reasonable approach via polynomials would fail to yield non-trivial results. However, we are able to adapt some bits of the polynomial method to the setting of low-degree polynomials (see Section 6).

Another complication is that, in the prior lower bound arguments, a nondeterministic procedure *guesses* a small circuit  $C$  of the kind one wishes to prove a lower bound against, and composes  $C$  with other Boolean circuitry to form a SAT instance. In our case, if we guess some arbitrary  $\text{SUM} \circ C$  circuit, we first need to know if this circuit is actually computing a Boolean function; if not, then the satisfiability question itself is not well-defined, and it will not be possible to meaningfully compose such a circuit with other Boolean circuits. Thus we need a way to efficiently check whether a linear combination is Boolean-valued.

We give a generic way to “lift” non-trivial algorithms for counting SAT assignments to short products of  $C$  circuits to non-trivial algorithms for detecting if a given  $\text{SUM} \circ C$  circuit is Boolean-valued and for counting SAT assignments. More precisely, we show that in order to prove lower bounds for linear combinations of  $C$ -functions, it suffices to solve a certain sum-product task faster than exhaustive search:



**Sum-Product over  $\mathcal{C}$ :** Given  $k$  functions  $f_1, \dots, f_k$  from  $\mathcal{C}$ , each on Boolean variables  $x_1, \dots, x_n$ , compute

$$\sum_{x \in \{0,1\}^n} \prod_{i=1}^k f_i(x).$$

Note the Sum-Product is computed over  $\mathbb{R}$ , and the task makes sense even if the functions  $f_1, \dots, f_k$  output *non-Boolean* values. Further note that if the functions  $f_1, \dots, f_k$  are Boolean-valued, then the product of  $k$  of them is simply the AND of  $k$  of them. In general, the Sum-Product problem will be NP-hard for most interesting representation classes: for example, it is already equivalent to Subset Sum when  $\mathcal{C}$  is the set of *exact* threshold functions (see Section 2 for a definition). Our meta-theorem states that mild improvements over exhaustive search for Sum-Product over  $\mathcal{C}$  imply strong lower bounds for  $\text{SUM} \circ \mathcal{C}$ :

► **Theorem 6.** *Suppose every  $C \in \mathcal{C}$  has a  $\text{poly}(n)$ -bit representation, where each  $C$  can be evaluated on a given input in  $\text{poly}(n)$  time. Assume there is an  $\varepsilon > 0$  and for  $k = 1, \dots, 4$  there is an  $n^{O(1)} \cdot 2^{n-\varepsilon n}$ -time algorithm for computing the Sum-Product of  $k$  functions  $f_1(x_1, \dots, x_n), \dots, f_k(x_1, \dots, x_n)$  from  $\mathcal{C}$ . Then:*

1. *For every  $k$ , there is a function in NP that does not have  $\text{SUM} \circ \mathcal{C}$  circuits of sparsity  $n^k$ .*
2. *For every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time constructible, there is a function in  $\text{NTIME}[n^{\alpha(n)}]$  that does not have  $\text{SUM} \circ \mathcal{C}$  circuits of polynomial sparsity.*

Theorem 6 is used to prove lower bounds against  $\text{SUM} \circ \text{THR}$ ,  $\text{SUM} \circ \text{ReLU}$ , and  $\text{SUM} \circ \text{MOD2} \circ \text{AND}_{O(1)}$ , by providing non-trivial algorithms solving the Sum-Product problem for these various classes. For the  $\text{E}^{\text{NP}}$  lower bounds, we use a closure property of  $\text{SUM} \circ \text{MOD2} \circ \text{AND}_{O(1)}$  combined with standard ideas from this line of work (see Theorem 21).

Theorem 6 (and its components) can also be used to easily “lift” existing circuit lower bounds to *linear combinations* of those circuits:

► **Theorem 7.** *For every  $d, m \geq 1$ , there is a  $b \geq 1$  and an  $f \in \text{NTIME}[n^{\log^b n}]$  that does not have  $\text{SUM} \circ \text{ACC}_d^0[m] \circ \text{THR}$  circuits of  $n^a$  size, for every  $a$ .*

That is, we obtain super-polynomial sparsity lower bounds on representing nondeterministic quasi-polynomial-time functions with  $\mathbb{R}$ -linear combinations of  $\text{ACC} \circ \text{THR}$  circuits (each of quasi-polynomial size). This applies the fact that we can solve the Sum-Product problem on  $\text{ACC} \circ \text{THR}$  circuits (because we can count SAT assignments to them), with an analogous running time as the best SAT algorithm. More details on Theorem 7 can be found in Section 3.

## Outline

The next section is the Preliminaries, which gives background knowledge. Section 3 proves Theorem 6. In Sections 4, 5, and 6, Sum-Product algorithms for  $\text{THR}$ ,  $\text{ReLU}$ , and  $\text{MODp} \circ \text{AND}_d$  (degree- $d$   $\mathbb{F}_p$ -polynomials) are provided which beat exhaustive search. The algorithms for  $\text{THR}$  and  $\text{ReLU}$  (Theorems 24 and 25) build upon and extend old Subset-Sum algorithms (Theorem 9). The algorithm for  $\text{MODp} \circ \text{AND}_d$  (Theorem 26) uses tools from the polynomial method in a new way. Applying Theorem 6 to each of these algorithms, we obtain strong lower bounds for  $\text{SUM} \circ \mathcal{C}$  for all three classes  $\mathcal{C}$ .

## 2 Preliminaries

Let  $\mathcal{C}$  be a class of functions of the form  $f : \{0, 1\}^n \rightarrow \mathbb{R}$ . Each member  $C \in \mathcal{C}$  has a number of inputs  $n$  and a size, which is the length of the representation of  $C$  in bits. For the classes THR,  $\text{MOD2} \circ \text{AND}_{O(1)}$ , and ReLU, the size  $|C|$  of a representation is  $\text{poly}(n)$  bits, without loss of generality; see Proposition 8. (For classes such as  $\text{MOD2} \circ \text{AND}_{\log_2(n)}$ , a member of the class takes  $\Omega(n^{\log n})$  bits to represent, in the worst case.) We assume that for all  $n$ , our class  $\mathcal{C}$  contains the projection functions  $f_i(x_1, \dots, x_n) = x_i$  for all  $i = 1, \dots, n$ . We also assume that  $\mathcal{C}$  is *evaluable*, meaning that there is a universal  $k \geq 1$  such that every  $C \in \mathcal{C}$  can be evaluated on a given input in  $O(|C|^k)$  time. All classes we consider have this property.

As is standard, we let  $\text{ANY}_c$  denote the class of Boolean functions with  $c$  inputs (the class contains “any” such function).

An arbitrary  $\text{SUM} \circ \mathcal{C}$  circuit  $C$  over  $n$  variables represents some function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$ . We say that  $C$  is *Boolean-valued* if for all  $x \in \{0, 1\}^n$ , the output of  $C$  on  $x$  is in  $\{0, 1\}$ . The following proposition is useful to keep in mind, as it shows that every sparse linear combination of Boolean functions implementing another Boolean function has an equivalent linear combination with “reasonable” coefficients.

► **Proposition 8.** *Let  $\mathcal{C}$  be a class of functions with co-domain  $\{0, 1\}$ , and let  $C$  be a  $\text{SUM} \circ \mathcal{C}$  circuit of sparsity  $s$  that is Boolean-valued. There is an equivalent  $\text{SUM} \circ \mathcal{C}$  circuit  $C'$  such that every weight in the linear combination of  $C'$  has the form  $j/k$ , where both  $j$  and  $k$  are integers in  $[-s^{s/2}, s^{s/2}]$ .*

**Proof.** (See also [34, 4].) Let  $C$  be a linear combination of  $s$  functions from  $\mathcal{C}$ . WLOG, the set of  $s$  Boolean functions from  $\mathcal{C}$  is a linearly independent set (otherwise, we could obtain a smaller linear combination representing the same function). The problem of finding coefficients for the Boolean-valued  $C$  is equivalent to solving a certain linear system  $Ax = b$  in  $s$  unknowns over the rationals, where  $b \in \{0, 1\}^{2^n}$  and  $A \in \{0, 1\}^{s \times 2^n}$ . Take a linearly independent subsystem of  $s$  of these  $2^n$  equations. Since the determinant of any  $s \times s$  Boolean matrix is in  $[-s^{s/2}, s^{s/2}]$  [21], the result follows from Cramer’s rule. ◀

The relevant theorem for sums of ReLU gates is more involved, but Maass [31] shows how the weights for a circuit of size  $s$  need only  $\text{poly}(s, n)$  bits of precision. Such “analog-to-digital” results are crucial for our work, as in our lower bound proofs we will need a discrete nondeterministic algorithm to *guess* a  $\text{SUM} \circ \mathcal{C}$  circuit and check various properties of it.

### Useful Results For Thresholds

We draw from several algorithms and representation theorems from past work. For  $\text{SUM} \circ \text{THR}$ , we eventually appeal to a classic result from exact algorithms:

► **Theorem 9** (Horowitz and Sahni [26]). *The number of Subset Sum solutions to any arbitrary instance of  $n$  items with integer weights of magnitude  $[-2^W, 2^W]$  can be computed in  $2^{n/2} \cdot \text{poly}(W)$  time.*

Theorem 9 is usually stated in terms of *finding* a subset sum solution, but the algorithm can be easily adapted to count solutions as well.

A Boolean function  $f$  is called an *exact threshold function* if there are real-valued  $\alpha_1, \dots, \alpha_n$  and  $t$  such that for all  $x$ ,

$$f(x) = 1 \iff \sum_i \alpha_i x_i = t.$$

Let ETHR be the class of exact threshold functions. For our  $\text{SUM} \circ \text{THR}$  circuit results, the following transformation is extremely useful:

► **Theorem 10** (Hansen and Podolskii [23]). *Every linear threshold function in  $n$  variables can be represented as an linear combination of  $\text{poly}(n)$  exact threshold functions, each with coefficient 1.*

It follows that every  $\text{SUM} \circ \text{THR}$  of sparsity  $s$  has an equivalent  $\text{SUM} \circ \text{ETHR}$  of sparsity  $\text{poly}(s)$ . The idea is that a  $\text{THR}$  function defines a set of points in the Boolean hypercube lying on one side of a given hyperplane; we can “cover” all the points lying on one side by a *disjoint* sum of  $\text{poly}(n)$  hyperplanes, which function as  $\text{ETHR}$  gates. Thus each coefficient in the linear combination is simply 1.

Another useful property of  $\text{ETHR}$  gates is that they are closed under AND:

► **Theorem 11** (Hansen and Podolskii [23]). *Every conjunction of  $t$  exact threshold functions in  $n$  variables with integer weights in  $[-W, W]$  can be converted in  $\text{poly}(t, n)$  time to an equivalent single exact threshold gate, with weights in  $[-(nW)^{\Theta(t)}, (nW)^{\Theta(t)}]$ .*

The idea is simple: if we multiply the  $i$ th exact threshold gate’s linear form by the factor  $(nW)^i$ , no linear form will “interfere” with the other sums, and we can determine if all of them are satisfied simultaneously with one exact threshold.

### Useful Results for Finite Field Polynomials

Two tools from the literature will be helpful for our results on linear combinations of polynomials. The first is *modulus-amplifying polynomials*, which have been used in Toda’s Theorem [46], representations of ACC and ACC-SAT algorithms [6, 53], algorithms for All-Pairs Shortest Paths [12], and algorithms for solving polynomial systems [29]:

► **Lemma 12** (Beigel and Tarui [6]). *For all  $\ell \in \mathbb{Z}^+$ , the degree- $(2\ell - 1)$  polynomial (over  $\mathbb{Z}$ )*

$$P_\ell(y) = 1 - (1 - y)^\ell \sum_{j=0}^{\ell-1} \binom{\ell + j - 1}{j} y^j$$

*has the property for all integers  $m \geq 2$ ,*

- *if  $y \equiv 0 \pmod{m}$  then  $P_\ell(y) \equiv 0 \pmod{m^\ell}$ ,*
- *if  $y \equiv 1 \pmod{m}$  then  $P_\ell(y) \equiv 1 \pmod{m^\ell}$ .*

*Furthermore, each coefficient in  $P_\ell$  has magnitude at most  $2^{O(\ell)}$ .*

Recall that a multivariate polynomial is *multilinear* if it contains no powers larger than one. The second tool is a classic result on rapidly evaluating a multilinear polynomial on all points in the Boolean hypercube.

► **Theorem 13** (cf. [8], Section 2.2). *Given the  $2^n$ -coefficient vector of a multilinear polynomial  $p \in \mathbb{Z}[x_1, \dots, x_n]$  where each coefficient is in  $[-W, W]$ , the value of  $p$  on all points in  $\{0, 1\}^n$  can be computed in  $2^n \cdot \text{poly}(n, \log W)$  time.*

The algorithm of Theorem 13 can be obtained by divide-and-conquer (as described in [50]) or by dynamic programming (as in [8], Section 2.2).

### Connections Between Nondeterministic Circuit UNSAT Algorithms and Circuit Lower Bounds

We also appeal to several known connections between circuit UNSAT algorithms that beat exhaustive search and circuit lower bounds against nondeterministic time classes, which build on prior work [51, 27, 41, 7].

► **Theorem 14** ([35]). *If there is an  $\varepsilon > 0$  such that Circuit Unsatisfiability for (fan-in 2) circuits with  $n$  inputs and  $2^{\varepsilon n}$  size is solvable in  $O(2^{n-\varepsilon n})$  nondeterministic time, then for every  $k$  there is a function in NP that does not have  $n^k$ -size (fan-in 2) circuits.*

► **Theorem 15** (Corollary 12 in Tell [45], following [35]). *If there is a  $\delta > 0$  and  $c \geq 1$  such that Circuit Unsatisfiability for (fan-in 2) circuits with  $n$  variables and  $m$  gates is solvable in  $O(2^{n(1-\delta)} \cdot m^c)$  nondeterministic time, then for every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time-constructible, there is a function in NTIME[ $n^{\alpha(n)}$ ] that is not in P/poly.*

► **Theorem 16** ([35]). *If there is an  $\varepsilon > 0$  such that Circuit Unsatisfiability for (fan-in 2) circuits with  $n$  inputs and  $2^{n^\varepsilon}$  size is solvable in  $O(2^{n-n^\varepsilon})$  nondeterministic time, then for every  $k$  there is a function in NTIME[ $n^{\text{poly}(\log n)}$ ] that does not have  $n^{\log^k n}$ -size (fan-in 2) circuits.*

In fact, all of these algorithms-to-lower-bounds connections still hold when we replace Circuit Unsatisfiability with the promise problem of distinguishing unsatisfiable circuits from circuits with  $2^{n-1}$  satisfying assignments.

### The Power of Linear Combinations of Low-Degree Polynomials

We note that classical work suggests that  $\mathbb{R}$ -linear combinations of higher-degree  $\mathbb{F}_2$ -polynomials can be quite powerful. For example, applying Valiant’s depth reduction [47] and using the representation of the AND function in the Fourier basis, it is easy to show that every  $O(n)$ -size  $O(\log n)$ -depth circuit can be represented by a linear combination of  $2^{O(n/\log \log n)}$   $\mathbb{F}_2$ -polynomials of degree  $O(n^\varepsilon)$ , for any desired  $\varepsilon > 0$ . Moreover, one can represent any  $O(n)$ -size “Valiant series-parallel” circuit (see [11]) by a linear combination of  $2^{\varepsilon n}$   $\mathbb{F}_2$ -polynomials of degree  $2^{O(1/\varepsilon)}$ . Hence there is a natural barrier to proving exponential-sparsity lower bounds for linear combinations of “somewhat-low” degree polynomials.

## 3 Meta-Theorem for Lower Bounds on Linear Combinations of Simple Functions

In this section, we prove our generic theorem which is applied in subsequent sections to prove lower bounds against linear combinations of threshold functions, ReLU gates, and constant-degree polynomials. Recall (from the Introduction) the Sum-Product problem:

**Sum-Product over  $\mathcal{C}$ :** Given  $k$  functions  $f_1, \dots, f_k$  from  $\mathcal{C}$ , each on Boolean variables  $x_1, \dots, x_n$ , compute

$$\sum_{x \in \{0,1\}^n} \prod_{i=1}^k f_i(x).$$

► **Reminder of Theorem 6.** *Suppose every  $C \in \mathcal{C}$  has a  $\text{poly}(n)$ -bit representation, where each  $C$  can be evaluated on a given input in  $\text{poly}(n)$  time. Assume there is an  $\varepsilon > 0$  and for  $k = 1, \dots, 4$  there is an  $n^{O(1)} \cdot 2^{n-\varepsilon n}$ -time algorithm for computing the Sum-Product of  $k$  functions  $f_1(x_1, \dots, x_n), \dots, f_k(x_1, \dots, x_n)$  from  $\mathcal{C}$ . Then:*

1. For every  $c$ , there is a function in NP that does not have  $\text{SUM} \circ \mathcal{C}$  circuits of sparsity  $n^c$ .
2. For every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time constructible, there is a function in  $\text{NTIME}[n^{\alpha(n)}]$  that does not have  $\text{SUM} \circ \mathcal{C}$  circuits of polynomial sparsity.

The remainder of this section will prove Theorem 6, and an extension to  $\text{E}^{\text{NP}}$  in some cases. We are able to use much of the earlier arguments [51, 53, 35] as black boxes. However we need several modifications.

The first new component needed is a method for checking that a given linear combination of  $\mathcal{C}$  circuits actually encodes a Boolean function (i.e. is Boolean-valued on all Boolean inputs). This is provided by the following theorem:

► **Theorem 17.** *Assume there is an  $\varepsilon > 0$  and for  $k = 1, \dots, 4$  there is an  $n^{O(1)} \cdot 2^{n-\varepsilon n}$ -time algorithm for computing the Sum-Product of  $k$  functions  $f_1(x_1, \dots, x_n), \dots, f_k(x_1, \dots, x_n)$  from  $\mathcal{C}$ .*

*Then there is a  $2^{n-\varepsilon n} \cdot \text{poly}(n, s)$ -time algorithm that, given  $f(x_1, \dots, x_n)$  which is an arbitrary linear combination of  $s$  functions from  $\mathcal{C}$ , determines whether or not  $f(a) \in \{0, 1\}$  for all  $a \in \{0, 1\}^n$ .*

**Proof.** Suppose we are given  $f = \sum_{i=1}^s \alpha_i c_i$ , where  $\alpha_i \in \mathbb{R}$  and  $c_i \in \mathcal{C}$  each have  $n$  inputs. Consider the polynomial

$$h(x) := f(x)^2 \cdot (1 - f(x))^2 = f(x)^2 - 2f(x)^3 + f(x)^4.$$

Observe that:

- If  $f(a) \in \{0, 1\}$  for all  $a \in \{0, 1\}^n$ , then  $h(a) = 0$  for all  $a$ .
- $f(b) \notin \{0, 1\}$  implies  $h(b) > 0$ .
- For all  $a \in \{0, 1\}^n$ ,  $h(a) \geq 0$ .

Therefore  $\sum_{a \in \{0, 1\}^n} h(a) = 0$  if and only if  $f(a) \in \{0, 1\}$  for all  $a \in \{0, 1\}^n$ . By applying the distributive law to each of  $f(x)^2$ ,  $f(x)^3$ ,  $f(x)^4$ , and exchanging the order of summation, we have

$$\begin{aligned} \sum_{a \in \{0, 1\}^n} h(a) &= \sum_{i_1, i_2} \beta_{i_1, i_2} \left( \sum_{a \in \{0, 1\}^n} f_{i_1}(x) \cdot f_{i_2}(x) \right) \\ &\quad + \sum_{i_1, i_2, i_3} \gamma_{i_1, i_2, i_3} \left( \sum_{a \in \{0, 1\}^n} f_{i_1}(x) \cdot f_{i_2}(x) \cdot f_{i_3}(x) \right) \\ &\quad + \sum_{i_1, i_2, i_3, i_4} \delta_{i_1, i_2, i_3, i_4} \left( \sum_{a \in \{0, 1\}^n} f_{i_1}(x) \cdot f_{i_2}(x) \cdot f_{i_3}(x) \cdot f_{i_4}(x) \right) \end{aligned}$$

for  $\beta_{i_1, i_2} = \alpha_{i_1} \cdot \alpha_{i_2}$ ,  $\gamma_{i_1, i_2, i_3} = -2\alpha_{i_1} \cdot \alpha_{i_2} \cdot \alpha_{i_3}$ ,  $\delta_{i_1, i_2, i_3, i_4} = \alpha_{i_1} \cdot \alpha_{i_2} \cdot \alpha_{i_3} \cdot \alpha_{i_4}$ .

Observe that each sum over  $a \in \{0, 1\}^n$  on the RHS is precisely a Sum-Product task over  $\mathcal{C}$ , with products ranging from  $k = 2$  to  $k = 4$ . Therefore we can check that the sum  $\sum_{a \in \{0, 1\}^n} h(a)$  is zero with  $O(s^4)$  calls to Sum-Product over  $\mathcal{C}$ . By assumption, this can be done in  $O(2^{n-\varepsilon n} \cdot \text{poly}(n, s))$  time. ◀

The second crucial component yields the ability to solve Circuit Unsatisfiability efficiently with nondeterminism, under the hypotheses (in fact, weaker hypotheses). This is provided by the following lemma, which is similar to (but more complicated than) Lemma 3.1 in [53]:

► **Lemma 18.** *Assume:*

- *There is an  $\varepsilon > 0$  and for  $k = 1, \dots, 4$  there is an  $n^{O(1)} \cdot 2^{n-\varepsilon n}$ -time algorithm for computing the Sum-Product of  $k$  functions from  $\mathcal{C}$ .*
- *The Circuit Evaluation problem has  $\text{SUM} \circ \mathcal{C}$  circuits of sparsity  $n^c$ , for some  $c > 0$ . Then there is a nondeterministic  $2^{n-\varepsilon n} \cdot \text{poly}(n, s)$ -time algorithm for Circuit Unsatisfiability, on arbitrary fan-in-2 circuits with  $n$  inputs and  $s$  gates.*

**Proof.** Suppose we are given a circuit  $C$  with  $n$  inputs and  $s$  gates of fan-in 2, and wish to nondeterministically prove it is unsatisfiable. Let us index the gates in topological order, so that gates  $1, \dots, n$  are the input gates, and the  $s$ -th gate is the output gate.

Our nondeterministic algorithm begins by guessing a  $\text{SUM} \circ \mathcal{C}$  circuit  $EVAL$  with  $n + O(\log s)$  inputs and sparsity at most  $(n + s)^{c+1}$ , which is intended to encode the Circuit Evaluation function:

$$EVAL(C, x, i) := \text{Evaluate } C \text{ on } x, \text{ and output the value of the } i\text{-th gate of } C.$$

(Note  $i$  is encoded as an  $O(\log s)$ -bit string.) Let

$$D(x, i) := EVAL(C, x, i),$$

i.e., we think of  $C$  as hard-coded in the function, to simplify the notation. Applying Theorem 17, we can check that  $D$  encodes a Boolean function in  $2^{n-\varepsilon n} \cdot \text{poly}(s, n)$  time.

Next, we check that  $D(a, s) = 0$  for all  $a \in \{0, 1\}^n$ ; in other words,  $D$  claims that  $C$  outputs 0 on every input. Suppose  $D$  has the form

$$D(x, i) = \sum_{j=1}^{(n+s)^{c+1}} \alpha_j \cdot c_j(x, i),$$

for some  $\alpha_j \in \mathbb{R}$  and  $c_j \in \mathcal{C}$ . Since  $D$  has already been determined to be Boolean, it suffices to compute  $\sum_{a \in \{0, 1\}^n} D(a, s)$  to know whether or not  $D(x, s) = 0$  for all  $a$ . By exchanging the order of summation,

$$\begin{aligned} \sum_{a \in \{0, 1\}^n} D(a, s) &= \sum_{a \in \{0, 1\}^n} \left( \sum_j \alpha_j \cdot c_j(a, i) \right) \\ &= \sum_j \alpha_j \cdot \left( \sum_{a \in \{0, 1\}^n} c_j(a, i) \right). \end{aligned}$$

Therefore we only need to make  $(n + s)^{c+1}$  calls to Sum-Product over  $\mathcal{C}$  (with  $k = 1$ ) to determine that  $D(x, s) = 0$  for all  $a \in \{0, 1\}^n$ . This can be done in  $2^{n-\varepsilon n} \cdot \text{poly}(n, s)$  time, by assumption.

Next, we have to check that for every gate  $i = 1, \dots, s$ , and every  $a \in \{0, 1\}^n$ ,  $D(a, i)$  correctly reports the output of the  $i$ -th gate when  $C$  evaluates  $a$ . To check the input gates, we need to check that  $D(x, i) = x_i$  for all  $i = 1, \dots, n$ ; we can do this by checking that

$$\sum_{a \in \{0, 1\}^n} (D(x, i) - x_i)^2 = 0,$$

which (by distributivity and re-arranging the order of summation, as in the proof of Theorem 17) can be computed with  $O((n + s)^{2(c+1)})$  calls to Sum-Product over  $\mathcal{C}$  (with  $k = 2$ ) in  $2^{n-\varepsilon n} \cdot \text{poly}(n, s)$  time.

For all gates  $i$  other than the input gates, the  $i$ th-gate takes inputs from previous gates indexed by some  $i_1 < i$  and  $i_2 < i$ , and computes a function of their two outputs. To check the consistency of gate  $i$ , we can form a degree-3 polynomial  $p_i(A, B, C)$  which outputs 0-1 values on all  $A, B, C \in \{0, 1\}$ , such that  $p_i(A, B, C) = 0$  if and only if  $A$  is the output of gate  $i$ , given that  $B$  is the output of gate  $i_1$  and  $C$  is the output of gate  $i_2$ .

Since  $D$  is Boolean-valued, we have reduced our problem to determining that

$$\sum_{a \in \{0,1\}^n} p(D(a, i), D(a, i_1), D(a, i_2)) = 0,$$

for each gate  $i = n + 1, \dots, s$ , and each gate  $i$ 's corresponding input gates  $i_1$  and  $i_2$ . Applying the distributive law to the LHS and exchanging the order of summation (as before), this results in  $O((n + s)^{3(c+1)})$  Sum-Product-over- $\mathcal{C}$  computations with up to  $k = 3$  products, computable in  $2^{n-\varepsilon n} \cdot \text{poly}(n, s)$  time.

Our nondeterministic algorithm determines that the input circuit  $C$  is unsatisfiable if and only if all of the above checks pass. If  $C$  is satisfiable, then every possible  $D$  guessed will fail some check. If  $C$  is unsatisfiable, then under the hypotheses of the theorem, a  $\text{SUM} \circ \mathcal{C}$  circuit  $D$  simulating every gate of  $C$  always exists. By guessing this  $D$ , and running the assumed Sum-Product algorithm, our nondeterministic algorithm accepts. ◀

After the above preparation, we turn back to the proof of Theorem 6. At this point, it is simply a matter of applying the above Lemma 18 with the known algorithms-to-lower-bound connections:

**Proof of Theorem 6.** Suppose every  $C \in \mathcal{C}$  has a  $\text{poly}(n)$ -bit representation, where each  $C$  can be evaluated on a given input in  $\text{poly}(n)$  time. Recall the hypothesis of the theorem is:

(A) There is an  $\varepsilon > 0$  and for  $k = 1, \dots, 4$  there is an  $n^{O(1)} \cdot 2^{n-\varepsilon n}$ -time algorithm for computing the Sum-Product of  $k$  functions  $f_1(x_1, \dots, x_n), \dots, f_k(x_1, \dots, x_n)$  from  $\mathcal{C}$ .

Furthermore, recall that Lemma 18 states:

Assuming (A) and assuming Circuit Evaluation has  $\text{SUM} \circ \mathcal{C}$  circuits of sparsity  $n^k$  for some  $k$ , there is a nondeterministic  $2^{n-\varepsilon n} \cdot \text{poly}(n, s)$ -time algorithm for Circuit Unsatisfiability, on arbitrary fan-in-2 circuits with  $n$  inputs and  $s$  gates.

We can then prove the lower bounds of the theorem readily, as follows.

- (1) Assume every function in NP has  $\text{SUM} \circ \mathcal{C}$  circuits of  $n^k$  sparsity circuits, for some fixed  $k$ . Then both hypotheses of Lemma 18 are satisfied (note Circuit Evaluation is in P), and the conclusion implies that there is an  $\varepsilon > 0$  such that Circuit Unsatisfiability for (fan-in 2) circuits with  $n$  inputs and  $2^{\varepsilon n}$  size is solvable in  $O(2^{n-\varepsilon n})$  nondeterministic time. Therefore by Theorem 14, for every  $k$  there is a function in NP that *does not* have  $n^k$ -size (fan-in 2) circuits. This is a contradiction because  $\text{SUM} \circ \mathcal{C}$  circuits of  $n^k$  sparsity can be simulated with  $n^{ck}$ -size fan-in-2 circuits, for some universal  $c$ .
- (2) The same argument as in (1) and (2) (but with Theorem 15 applied) shows that for every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time-constructible, there is a function in  $\text{NTIME}[n^{\alpha(n)}]$  that does not have  $\text{SUM} \circ \mathcal{C}$  circuits of polynomial sparsity.

This completes the proof. ◀



### A Note on Lower Bounds for Linear Combinations of ACC Circuits

There are other new lower bound consequences of the arguments in Theorem 6 that we will not study in detail here, because they follow easily from combining known results. Here is an example:

► **Reminder of Theorem 7.** *For every  $d, m \geq 1$ , there is a  $b \geq 1$  and an  $f \in \text{NTIME}[n^{\log^b n}]$  that does not have  $\text{SUM} \circ \text{AC}_d^0[m] \circ \text{THR}$  circuits of  $n^a$  size, for every  $a$ .*

This lower bound can be obtained as follows. First, the argument of Lemma 18 also shows:

► **Theorem 19.** *Assume*

- *There is an  $\varepsilon > 0$  and for  $k = 1, \dots, 4$  there is an  $n^{O(1)} \cdot 2^{n-n^\varepsilon}$ -time algorithm for computing the Sum-Product of  $k$  functions from  $\mathcal{C}$ .*
- *The Circuit Evaluation problem has  $\text{SUM} \circ \mathcal{C}$  circuits of sparsity  $n^a$ , for some  $a > 0$ . Then there is a nondeterministic  $2^{n-n^\varepsilon} \cdot \text{poly}(n, s)$ -time algorithm for Circuit Unsatisfiability, on arbitrary fan-in-2 circuits with  $n$  inputs and  $s$  gates.*

Now we combine this theorem with the following two facts:

1. For every depth  $d$  and integer  $m \geq 2$ , there is an  $\varepsilon > 0$  such that the Sum-Product of  $O(1)$   $\text{AC}_d^0[m] \circ \text{THR}$  circuits of  $2^{n^\varepsilon}$  size can be computed in  $2^{n-n^\varepsilon}$  time. This simply applies the algorithm for counting satisfying assignments of  $\text{AC}_d^0[m] \circ \text{THR}$  circuits ([52]).
2. If for some  $\alpha > 0$  there is a nondeterministic  $2^{n-n^\alpha}$ -time Circuit Unsatisfiability algorithm for  $2^{n^\alpha}$ -size circuits, then for every  $a \geq 1$ , there is a  $b \geq 1$  such that  $\text{NTIME}[n^{\log^b n}]$  does not have  $n^{\log^a n}$ -size circuits (this is a theorem of Murray and Williams [35]).

Theorem 7 is immediate: Assuming  $\text{NTIME}[n^{\log^b n}]$  has  $\text{SUM} \circ \text{AC}_d^0[m] \circ \text{THR}$  circuits of  $n^a$  size for some  $a \geq 1$ , both hypotheses of Theorem 19 are satisfied for  $\mathcal{C} = \text{AC}_d^0[m] \circ \text{THR}$ , and the conclusion of Theorem 19 combined with item 2 above yields a contradiction.

### 3.1 Lower Bounds for Exponential Time With an NP Oracle

For classes  $\mathcal{C}$  with a natural closure property, the lower bounds can be extended to  $2^{\Omega(n)}$  sparsity for a function in  $\text{E}^{\text{NP}}$ . Recall  $\text{ANY}_c$  denotes the class of Boolean functions with  $c$  inputs (the class contains “any” such function).

For an integer  $c \geq 1$ , we say that  $\mathcal{C}$  is *efficiently closed under  $\text{NC}_c^0$*  if there is a polynomial-time algorithm  $A$  such that, given any circuit  $C$  of the form  $\mathcal{C} \circ \text{ANY}_c$ , algorithm  $A$  outputs an equivalent circuit  $D$  from  $\mathcal{C}$  (which is only polynomially larger). We note this property is true of  $O(1)$ -degree polynomials:

► **Proposition 20.** *For every integer  $m \geq 2$  and  $c \geq 1$ , the class  $\mathcal{C} = \bigcup_{d \geq 1} \text{MOD}_m \circ \text{AND}_d$  is efficiently closed under  $\text{NC}_c^0$ .*

**Proof.** Every  $\text{MOD}_m \circ \text{AND}_d \circ \text{ANY}_c$  circuit can be represented by an  $\text{MOD}_m \circ \text{AND}_{dc}$  circuit. In particular, every Boolean function on  $c$  inputs has an exact representation as a sum (modulo  $m$ ) of ANDs of fan-in  $c$ ; composing such a sum with a  $\text{MOD}_m \circ \text{AND}$  circuit and applying the distributive law yields the result. ◀

► **Theorem 21.** *There is a universal  $c \geq 1$  satisfying the following. Suppose  $\mathcal{C}$  is efficiently closed under  $\text{NC}_c^0$ , and suppose every  $C \in \mathcal{C}$  has a  $\text{poly}(n)$ -bit representation, where each  $C$  can be evaluated on a given input in  $\text{poly}(n)$  time.*

*Assume there is an  $\varepsilon > 0$  and for  $k = 1, \dots, 4$  there is an  $n^{O(1)} \cdot 2^{n-\varepsilon n}$ -time algorithm for*



computing the Sum-Product of  $k$  functions  $f_1(x_1, \dots, x_n), \dots, f_k(x_1, \dots, x_n)$  from  $\mathcal{C}$ . Then there is a function in  $\mathbf{E}^{\text{NP}}$  that does not have  $\text{SUM} \circ \mathcal{C}$  circuits of sparsity  $2^{\alpha n}$ , for some  $\alpha > 0$ .

The remainder of this section sketches the proof of Theorem 21; we give only a sketch, as the argument closely resembles others [53, 27]).

Let  $\varepsilon \in (0, 1)$ . Assume  $\mathcal{C}$  is efficiently closed under  $\text{NC}_c^0$ . Furthermore:

- (A) There is an  $\varepsilon > 0$  and an  $O(2^{n-\varepsilon n})$ -time algorithm for computing the Sum-Product of  $k$  functions from  $\mathcal{C}$ , and
- (B) For all functions  $f \in \text{TIME}[2^{O(n)}]^{\text{NP}}$  and all  $\alpha > 0$ ,  $f$  has  $\text{SUM} \circ \mathcal{C}$  circuits of sparsity  $2^{\alpha n}$ .

We wish to establish a contradiction. In particular, we will show that assumptions (A) and (B) together imply that every problem in  $\text{NTIME}[2^n]$  can be simulated by a nondeterministic  $o(2^n)$ -time algorithm, contradicting the (strong) nondeterministic time hierarchy theorem [42, 56].

Let  $L \in \text{NTIME}[2^n]$ . On a given input  $x$ , our nondeterministic  $o(2^n)$ -time algorithm for  $L$  has two parts:

- (i) It guesses a witness for  $x$  of  $o(2^n)$  size.
- (ii) It verifies that witness for  $x$  in  $o(2^n)$  time.

To handle (i), we use assumption (B) to show that one can nondeterministically guess a  $2^{\alpha n} \cdot \text{poly}(n)$ -size  $\text{SUM} \circ \mathcal{C}$  circuit that encodes a witness for  $x$ , applying a simple “easy witness” lemma from [51]:

► **Lemma 22** (Lemma 3.2 in [51]). *Let  $\mathcal{D}$  be any class of circuits. If  $\mathbf{E}^{\text{NP}}$  has circuits of size  $S(n)$  from class  $\mathcal{D}$ , then for every  $L \in \text{NTIME}[2^n]$  and every verifier  $V$  for  $L$ , and every  $x \in L$  of length  $n = |x|$ , there is a  $y$  of length  $O(2^n)$  such that  $V(x, y)$  accepts and the  $\mathcal{D}$ -circuit complexity of  $y$  (construed as a function  $f : \{0, 1\}^{n+O(1)} \rightarrow \{0, 1\}$ ) is at most  $S(n)$ .*

In other words, assumption (B) implies that every yes-instance of  $L$  has  $S(n)$ -size “witness circuits”: a witness of length  $O(2^n)$  that can be represented as an  $S(n)$ -size  $\text{SUM} \circ \mathcal{C}$  Boolean-valued circuit. Furthermore, this holds for every verifier for  $L$ .

To handle (ii), we choose an appropriate verifier, so that verifying witnesses becomes equivalent to a simple Sum-Product call. In particular we use the following extremely “local” reduction from  $L \in \text{NTIME}[2^n]$  to 3SAT instances of  $2^n \cdot \text{poly}(n)$  length:

► **Lemma 23** ([27]). *Every  $L \in \text{NTIME}[2^n]$  can be reduced to 3SAT instances of  $O(2^n \cdot n^4)$  size. Moreover, there is an algorithm that, given an instance  $x$  of  $L$  and an integer  $i \in [O(2^n \cdot n^4)]$  in binary, reads only  $O(1)$  bits of  $x$  and outputs the  $i$ -th clause of the resulting 3SAT formula, in  $O(n^4)$  time.*

Since in Lemma 23 each bit of the output is a function of some  $c \leq O(1)$  inputs, each bit of the output is a member of  $\text{ANY}_c$ . So for every instance  $x$  of length  $n$  for the language  $L$ , we can produce (in deterministic  $\text{poly}(n)$  time) a circuit  $D_x$  which is an ordered collection of  $O(n)$  functions from  $\text{ANY}_c$ . The circuit  $D_x$  takes  $n + O(\log n)$  binary inputs, construes that input as an integer  $i$ , and outputs the  $i$ -th clause of a formula  $F_x$  which is satisfiable if and only if  $x \in L$ .

Our nondeterministic algorithm for  $L$  guesses a  $2^{O(\alpha n)}$ -sparse  $\text{SUM} \circ \mathcal{C}$  circuit  $C_x$  that takes  $n + O(\log n)$  inputs and is meant to encode a satisfying assignment for the formula  $F_x$ . We can check  $C_x$  is Boolean-valued on all  $2^n \cdot \text{poly}(n)$  inputs in  $2^{n-\varepsilon n/2}$  time, by applying Theorem 17 and letting  $\alpha > 0$  be sufficiently small.

Composing  $C_x$  with the  $O(n)$  polynomials forming  $D_x$ , we obtain a  $2^{O(\alpha n)}$ -sparse  $\text{SUM} \circ \mathcal{C} \circ \text{ANY}_c$  circuit  $E$  with  $n + O(\log n)$  inputs (composed of three copies of  $C_x$ , and  $O(n)$  copies of  $D_x$ ) such that

$E$  is unsatisfiable if and only if  $C_x$  encodes a satisfying assignment for  $F_x$ .

(We leave out the details, as they are provided in multiple other papers [51, 53].) To complete the  $o(2^n)$ -time algorithm for  $L$ , it suffices to check unsatisfiability of the resulting  $2^{O(\alpha n)}$ -size circuit  $E$  in  $o(2^n)$  nondeterministic time. This would yield the desired contradiction.

Such a nondeterministic UNSAT algorithm is provided by first converting  $E$  into a  $\text{SUM} \circ \mathcal{C}$  circuit in  $2^{O(\alpha n)}$  time (using the fact that  $\mathcal{C}$  is efficiently closed under  $\text{NC}^0$ ). This yields a sum of  $2^{O(\alpha n)}$   $\mathcal{C}$ -circuits. Analogously to the proof of Lemma 18, checking the unsatisfiability of such an  $E$  can be reduced to  $2^{O(\alpha n)}$  calls to Sum-Product of  $\mathcal{C}$ , by applying distributivity. Applying the Sum-Product algorithm of assumption (A) that runs in  $O(2^{n-\varepsilon n})$  time, and setting  $\alpha > 0$  to be sufficiently small, the running time is  $o(2^n)$ .

This completes the proof of Theorem 21.

#### 4 Sparse Combinations of Threshold Functions

We now turn to proving  $\text{SUM} \circ \text{THR}$  lower bounds. Due to Lemma 6, it suffices to give a  $2^{n-\varepsilon n}$ -time algorithm for the Sum-Product Problem over THR:

**Sum-Product over THR:** Given  $k$  linear threshold functions  $f_1, \dots, f_k$ , each on Boolean variables  $x_1, \dots, x_n$ , compute

$$\sum_{x \in \{0,1\}^n} \prod_{i=1}^k f_i(x).$$

Putting together various pieces (described in the Preliminaries), there is a substantially faster-than- $2^n$  time algorithm:

► **Theorem 24.** *The Sum-Product of  $k$  linear threshold functions on  $n$  variables (with weights in  $[-n^n, n^n]$ ) can be computed in  $2^{n/2} \cdot n^{O(k)}$  time.*

Note that having weights in  $[-n^n, n^n]$  is without loss of generality (in our lower bound proofs, our nondeterministic algorithm can always guess an equivalent circuit with such weights, as described by Proposition 8).

**Proof.** Let  $f_1, \dots, f_k$  be  $n$ -variable threshold functions. Applying Theorem 10, we can write each  $f_i$  as a sum of  $t = \text{poly}(n)$  exact threshold functions:

$$f_i(x) = \sum_{i=1}^t g_i(x),$$

where each  $g_i(x)$  is defined by some weights  $w_{i,1}, \dots, w_{i,n} \in \mathbb{R}$  and a threshold value  $t \in \mathbb{R}$ . Therefore we can write the product  $f_1 \cdots f_k$  as

$$\prod_{i=1}^k f_i = \sum_{(i_1, \dots, i_k) \in [t]^k} g_{i_1} \cdots g_{i_k}.$$

Each term  $g_{i_1} \cdots g_{i_k}$  is a conjunction of  $k$  exact thresholds. Applying Theorem 11, each such term can be replaced with a single exact threshold gate, with weights of magnitude  $n^{O(kn)}$ , i.e., each weight is representable with  $O(kn \log n)$  bits. Thus

$$\prod_{i=1}^k f_i = \sum_{(i_1, \dots, i_k) \in [t]^k} h_{i_1, \dots, i_k}$$

for some exact threshold gates  $h_{i_1, \dots, i_k}$ . The desired sum can therefore be written as

$$\begin{aligned} \sum_{a \in \{0,1\}^n} \prod_{i=1}^k f_i(a) &= \sum_{a \in \{0,1\}^n} \sum_{(i_1, \dots, i_k) \in [t]^k} h_{i_1, \dots, i_k}(a) \\ &= \sum_{(i_1, \dots, i_k) \in [t]^k} \left( \sum_{a \in \{0,1\}^n} h_{i_1, \dots, i_k}(a) \right). \end{aligned}$$

Now observe that each sum  $\sum_{a \in \{0,1\}^n} h_{i_1, \dots, i_k}(a)$  on the RHS is equivalent to an instance of #Subset Sum. In particular, each such sum is counting the number of subsets of a given set of  $n$  weights in  $[-n^{\Omega(kn)}, n^{O(kn)}]$  which sum to zero. By Theorem 9, this can be computed in  $\text{poly}(k, n) \cdot 2^{n/2}$  time. Since there are  $n^{O(k)}$  such sums to compute in the outer sum, the total running time is  $n^{O(k)} \cdot 2^{n/2}$ . ◀

The following are immediate from Theorem 6:

▶ **Reminder of Theorem 1.** *For all  $k$ , there is an  $f_k \in \text{NP}$  without  $\text{SUM} \circ \text{THR}$  circuits of  $n^k$  sparsity. Furthermore, for every unbounded  $\alpha(n)$  such that  $n^{\alpha(n)}$  is time constructible, there is a function in  $\text{NTIME}[n^{\alpha(n)}]$  that does not have  $\text{SUM} \circ \text{THR}$  circuits of polynomial sparsity.*

## 5 Sparse Combinations of ReLU Gates

Recall that a function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  from the class ReLU is defined with respect to a weight vector  $w \in \mathbb{R}^n$  and a scalar  $a \in \mathbb{R}$ , such that for all  $a \in \{0, 1\}^n$ ,

$$f(x) = \max\{0, \langle w, x \rangle + a\}.$$

To prove  $\text{SUM} \circ \text{ReLU}$  lower bounds, we give a  $2^{n-\varepsilon n}$ -time algorithm for the Sum-Product Problem over ReLU:

**Sum-Product over ReLU:** Given  $k$  ReLU functions  $f_1, \dots, f_k$ , each on Boolean variables  $x_1, \dots, x_n$ , compute

$$\sum_{x \in \{0,1\}^n} \prod_{i=1}^k f_i(x).$$

▶ **Theorem 25.** *The Sum-Product of  $k$  ReLU functions on  $n$  variables (with weights in  $[-W, W]$ ) can be computed in  $2^{n/2} \cdot n^{O(k)} \cdot \text{poly}(k, n, \log W)$  time.*

The proof is similar in spirit to the algorithm for Sum-Product of threshold functions (Theorem 24), except that complications arise due to the real-valued outputs of ReLU functions. We end up having to solve a problem generalizing #Subset Sum, but which turns out to have a nice “split-and-list”  $2^{n/2}$ -time algorithm, analogously to #Subset Sum.

**Proof.** Let  $f_1, \dots, f_k$  be  $n$ -variable ReLU functions, defined by weight vectors  $w_1, \dots, w_k \in \mathbb{R}^n$  and scalars  $a_1, \dots, a_k \in \mathbb{R}$ , respectively. Our task is to compute

$$\sum_{x \in \{0,1\}^n} \max\{0, \langle x, w_1 \rangle + a_1\} \cdots \max\{0, \langle x, w_k \rangle + a_k\}.$$

First, we note the above sum is equal to

$$\sum_{x \in \{0,1\}^n} [\langle x, w_1 \rangle \geq -a_1] \cdot (\langle x, w_1 \rangle + a_1) \cdots [\langle x, w_k \rangle \geq -a_k] \cdot (\langle x, w_k \rangle + a_k),$$

where we are using the Iverson bracket notation  $[P]$  to denote a function that outputs 1 if  $P$  is true and 0 otherwise. Applying Theorem 10, each of the threshold functions  $[\langle x, w_i \rangle \geq -a_i]$  can be represented as a linear combination of  $t = \text{poly}(n)$  exact threshold functions. In particular there are exact thresholds  $g_{i,j}$  such that the above sum equals

$$\sum_x \left( \sum_{j=1}^t g_{1,j}(x) \right) \cdot (\langle x, w_1 \rangle + a_1) \cdots \left( \sum_{j=1}^t g_{k,j}(x) \right) \cdot (\langle x, w_k \rangle + a_k).$$

Applying the distributive law, the above sum equals

$$\sum_x \sum_{j_1, \dots, j_k \in [t]^k} g_{1,j_1}(x) \cdots g_{k,j_k}(x) \cdot (\langle x, w_1 \rangle + a_1) \cdots (\langle x, w_k \rangle + a_k).$$

Re-arranging the summation order yields

$$\sum_{j_1, \dots, j_k \in [t]^k} \left( \sum_x g_{1,j_1}(x) \cdots g_{k,j_k}(x) \cdot (\langle x, w_1 \rangle + a_1) \cdots (\langle x, w_k \rangle + a_k) \right).$$

Applying Theorem 11, each  $g_{1,j_1}(x) \cdots g_{k,j_k}(x)$  can be replaced by a single exact threshold  $h_{j_1, \dots, j_k}(x)$ .

Our task has been reduced to  $n^{O(k)}$  computations of the form

$$\sum_{x \in \{0,1\}^n} h_{j_1, \dots, j_k}(x) \cdot (\langle x, w_1 \rangle + a_1) \cdots (\langle x, w_k \rangle + a_k). \quad (1)$$

Without the  $(\langle x, w_1 \rangle + a_1) \cdots (\langle x, w_k \rangle + a_k)$  term, (1) would be exactly a #Subset Sum instance, as in Theorem 24. In this new situation, we need to count a “weighted” sum over the subset sum solutions, where the weights are determined by a product of  $k$  inner products of the solution vectors with some fixed vectors.

Let us now describe how to solve the generalized problem given by (1). To keep the exposition clear, we will walk through an attempted solution and fix it as it breaks.

Suppose the exact threshold function  $h_{j_1, \dots, j_k}(x)$  of (1) is defined by weights  $\alpha_1, \dots, \alpha_n \in \mathbb{R}$  and threshold value  $t \in \mathbb{R}$ , so that

$$h_{j_1, \dots, j_k}(x) = 1 \iff \sum_{i=1}^n \alpha_i x_i = t.$$

As with the Subset Sum problem, we begin by splitting the set of variables  $x$  into two halves,  $\{x_1, \dots, x_{n/2}\}$  and  $\{x_{n/2+1}, \dots, x_n\}$  (WLOG, assume  $n$  is even). Correspondingly, we split each of the  $k$  weight vectors  $w_i \in \mathbb{R}^n$  of (1) into two halves,  $w_i^{(1)} \in \mathbb{R}^{n/2}$  and  $w_i^{(2)} \in \mathbb{R}^{n/2}$  for the first and second halves of variables, respectively.

We list all  $2^{n/2}$  partial assignments to the first half, and all  $2^{n/2}$  partial assignments to the second. For each partial assignment  $A = (A_1, \dots, A_{n/2})$  to the first half of variables  $\{x_1, \dots, x_{n/2}\}$ , we compute a vector  $v_A$ , as follows:

- $v_A[0] := -t + \sum_{i=1}^{n/2} \alpha_i A_i$ ,
- for all  $j = 1, \dots, k$ ,  $v_A[j] := a_j + \langle w_j^{(1)}, (A_1, \dots, A_{n/2}) \rangle$ .

For each partial assignment  $A' = (A_{n/2+1}, \dots, A_n)$  from the second half, we compute a vector  $w_{A'}$ :

- $w_{A'}[0] := \sum_{i=n/2+1}^n \alpha_i A_i$ ,
- for all  $j = 1, \dots, k$ ,  $w_{A'}[j] := \langle w_j^{(2)}, (A_{n/2+1}, \dots, A_n) \rangle$ .

Notice that  $v_A[0] + w_{A'}[0] = 0$  if and only if  $h_{j_1, \dots, j_k}(A, A') = 1$ . Thus in our sum, we only need to consider pairs of vectors  $v_A$  from the first half and vectors  $w_{A'}$  from the second half such that  $v_A[0] + w_{A'}[0] = 0$ . Moreover, note that for all  $j = 1, \dots, k$ ,

$$v_A[j] + w_{A'}[j] = \langle x, w_j \rangle + a_j.$$

It follows that (1) equals

$$\sum_{(v_A, w_{A'}) : v_A[0] + w_{A'}[0] = 0} (v_A[1] + w_{A'}[1]) \cdots (v_A[k] + w_{A'}[k]).$$

The Subset-Sum algorithm of Horowitz and Sahni [26] shows how to efficiently find pairs  $(v_A, w_{A'})$  with  $v_A[0] + w_{A'}[0] = 0$ : sorting all vectors in the second half by their 0-th coordinate, for each vector  $v_A$  from the first half we can compute (in  $\text{poly}(n)$  time) the number of second-half vectors  $w_{A'}$  satisfying  $v_A[0] + w_{A'}[0] = 0$  (even if there are exponentially many such vectors). However it is unclear how to incorporate the odd-looking  $(v_A[1] + w_{A'}[1]) \cdots (v_A[k] + w_{A'}[k])$  multiplicative factors into a weighted sum.

To do so, we modify the vectors  $v_A$  and  $w_B$  as follows. Consider the expansion of  $\prod_{i=1}^k (v_A[i] + w_{A'}[i])$  into a sum of  $2^k$  products: it can be seen as the inner product of two  $2^k$ -dimensional vectors, where one vector's entries is a function solely of  $v_A$  and the other vector's entries is a function solely of  $w_{A'}$ . (Furthermore, note that the number of bits needed to describe entries in these new vectors has increased only by a multiplicative factor of  $k$ .)

Thus we can assign  $(2^k + 1)$ -dimensional vectors  $v'_A$  (in place of the  $v_A$ ) and  $w'_B$  (in place of the  $w_B$ ) such that  $v'_A[0] = v_A[0]$ ,  $w'_A[0] = w_A[0]$ , and for all  $A, A'$  we have

$$(v_A[1] + w_{A'}[1]) \cdots (v_A[k] + w_{A'}[k]) = \sum_{j=1}^{2^k} v'_A[j] \cdot w'_{A'}[j].$$

Now our goal is to compute

$$\sum_{(v'_A, w'_{A'}) : v'_A[0] + w'_{A'}[0] = 0} \left( \sum_{j=1}^{2^k} v'_A[j] \cdot w'_{A'}[j] \right). \tag{2}$$

We can get a more efficient algorithm for the problem defined by (2), by preprocessing the second half of vectors (i.e., the  $w'_{A'}$  vectors). For each distinct value  $e = w'_{A'}[0] \in \mathbb{R}$  among the  $2^{n/2}$  vectors in the second half, we make a new  $(2^k + 1)$ -dimensional vector  $W'_e$  where:

- $W'_e[0] = e$ , and
- for all  $i = 1, \dots, 2^k$ ,  $W'_e[i] = \sum_{w'_A : w'_A[0] = e} w'_A[i]$ .

That is, the coordinates  $1, \dots, 2^k$  of  $W'_e$  are obtained by component-wise summing all vectors  $w'_A$  such that  $w'_A[0] = e$ . The preparation of the vectors  $W'_e$  can be done in  $2^{n/2} \cdot \text{poly}(k, n, \log W)$  time, by partitioning all  $2^{n/2}$  vectors  $w'_A$  from the second half of variables into equivalence classes (where two vectors are equivalent if their 0-coordinates are equal), then obtaining each  $W'_e$  by summing the vectors in one equivalence class.

Finally, we can use the  $W'_{A'}$  vectors to compute the sum (2) in  $2^{n/2} \cdot 2^k \cdot \text{poly}(k, n, \log W)$  time. Have a running sum that is initially 0. Iterate through each vector  $v'_A$  from the first half of variables, look up the corresponding second-half vector  $W'_e$  (with  $v'_A[0] = -W'_e[0]$ ) in  $\text{poly}(k, n, \log W)$  time, and add the inner product

$$\sum_{i=1}^{2^k} v'_A[i] \cdot W'_e[i]$$

to the running sum. Because each vector  $(W'_e[1], \dots, W'_e[2^k])$  is the sum of *all* vectors  $(w'_{A'}[1], \dots, w'_{A'}[2^k])$  such that  $v'_A[0] + w'_{A'}[0] = 0$ , each inner product  $\sum_{i=1}^{2^k} v'_A[i] \cdot W'_e[i]$  contributes

$$\sum_{w'_{A'} : v'_A[0] + w'_{A'}[0] = 0} \left( \sum_{j=1}^{2^k} v'_A[j] \cdot w'_{A'}[j] \right)$$

to the running sum. Therefore after iterating through all vectors  $v'_A$ , our running sum has computed (2) exactly, in only  $2^{n/2} \cdot 2^k \cdot \text{poly}(n, \log W)$  time. ◀

From the algorithm of Theorem 25, we immediately obtain the  $\text{SUM} \circ \text{ReLU}$  lower bounds of Theorem 2.

## 6 Sparse Combinations of Low-Degree Polynomials over Finite Fields

We can also prove lower bounds for linear combinations of low-degree  $\mathbb{F}_p$ -polynomials in  $n$  variables, for any prime  $p$ , by giving a faster Sum-Product algorithm. In this context, the Sum-Product problem becomes:

**Sum-Product over  $\text{MOD}_p \circ \text{AND}_d$ :** Given  $k$  polynomials  $p_1, \dots, p_k \in \mathbb{F}_p[x_1, \dots, x_n]$ , each of degree at most  $d$ , compute

$$\sum_{x \in \{0,1\}^n} \left( \prod_{i=1}^k p_i(x) \right),$$

where the sum over all  $x \in \{0,1\}^n$  is taken over the reals (or rationals).

That is, we treat each  $\prod_{i=1}^k p_i(x)$  as a function from  $\{0,1\}^n$  to  $\{0,1,\dots,p-1\} \subset \mathbb{Q}$ , and wish to compute the sum of these integers over all  $x \in \{0,1\}^n$ .

In related work, Lokshtanov *et al.* [29] showed how to (deterministically) count solutions in  $\mathbb{F}_p^n$  to a system of  $\ell$  degree- $d$   $\mathbb{F}_p$ -polynomials in  $p^{n+o(n)-n/O(dp^{6/7})} \cdot \text{poly}(\ell)$  time. For our Sum-Product problem, we need to compute a “weighted” sum (the terms can take on values in  $\{0, \dots, p-1\}$ ), and we need to count the weighted sum over only *Boolean* assignments. We can achieve this, with a comparable runtime savings involving  $k$  and  $p$ :

► **Theorem 26.** *The Sum-Product of  $k$  degree- $d$  polynomials  $p_1, \dots, p_k \in \mathbb{F}_p[x_1, \dots, x_n]$  can be computed in  $p^{2k} \cdot (1.9^n + 2^{n-n/(6dp)}) \cdot \text{poly}(n)$  time.*

**Proof.** Let  $p_1, \dots, p_k$  be given. We wish to compute

$$\sum_{x \in \{0,1\}^n} \left( \prod_{i=1}^k p_i(x) \right), \tag{3}$$

where each product outputs an integer in  $\{0, 1, \dots, p-1\}$ . We first convert the Sum-Product problem of (3) to an equivalent sum where each “term” in the sum is a small system of polynomial equations.

We say that a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is an *exact  $\mathbb{F}_p$ -polynomial function* if there is a polynomial  $p \in \mathbb{F}_p[x_1, \dots, x_n]$  and  $a \in \mathbb{F}_p$  such that for all  $x \in \{0, 1\}^n$ ,

$$f(x) = 1 \iff p(x) = a.$$

We use the notation  $[p(x) = a]$  to denote such an exact polynomial function. Let us replace each polynomial  $p_i(x)$  in the sum-product expression with an equivalent linear combination (over  $\mathbb{Z}$ ) of exact polynomial functions. In particular, replace each  $p_i(x)$  with the sum *over the integers*

$$\sum_{a \in \mathbb{F}_p} a \cdot [p_i(x) = a].$$

That is, we are replacing  $p_i(a)$  with an equivalent integer-valued sum of  $p$  Boolean functions. Now the desired sum (3) has the form:

$$\begin{aligned} & \sum_{x \in \{0,1\}^n} \left( \prod_{i=1}^k \left( \sum_{a \in \mathbb{F}_p} a \cdot [p_i(x) = a] \right) \right) \\ &= \sum_{x \in \{0,1\}^n} \sum_{(a_1, \dots, a_k) \in \mathbb{F}_p^k} a_1 \cdots a_k \cdot \prod_{i=1}^k [p_i(x) = a_i] \quad (\text{by distributivity}) \end{aligned} \tag{4}$$

$$= \sum_{(a_1, \dots, a_k) \in \mathbb{F}_p^k} a_1 \cdots a_k \cdot \left( \sum_{x \in \{0,1\}^n} [p_1(x) = a_1] \cdots [p_k(x) = a_k] \right). \tag{5}$$

Each inner sum in (5) counts the number of Boolean solutions to a system of polynomial equations  $p_1(x) = a_1, \dots, p_k(x) = a_k$ . We can further reduce this problem to counting the number of Boolean solutions to *one* equation, by applying a simple reduction (from [54]). Namely, we have the equation

$$\begin{aligned} & \sum_{x \in \{0,1\}^n} \prod_{i=1}^k [p_i(x) = a_i] \\ &= \frac{1}{p^k} \sum_{(b_1, \dots, b_k) \in \mathbb{F}_p^k} \sum_{x \in \{0,1\}^n} \left( \left[ \sum_{j=1}^k b_j \cdot (p_j(x) - a_j) = 0 \right] - \left[ \sum_{j=1}^k b_j \cdot (p_j(x) - a_j) = 1 \right] \right). \end{aligned} \tag{6}$$

To see why (6) holds, let  $x \in \{0, 1\}^n$  such that  $[p_1(x) = a_1] \cdots [p_k(x) = a_k] = 1$ . Then for *every*  $(b_1, \dots, b_k) \in \mathbb{F}_p^k$ , we have  $[\sum_{j=1}^k b_j \cdot (p_j(x) - a_j) = 0] = 1$ . So every solution  $x$  to the system of  $k$  equations is counted for  $p^k$  times in (6); since the result is divided by  $p^k$ , each solution contributes 1 to (6). On the other hand, if  $x$  is not a solution to the system, and  $[p_1(x) = a_1] \cdots [p_k(x) = a_k] = 0$ , then for some  $j$ ,  $p_j(x) - a_j \neq 0$ . It follows that there are precisely  $p^{k-1}$  vectors  $(b_1, \dots, b_k) \in \mathbb{F}_p^k$  such that  $[\sum_{j=1}^k b_j \cdot (p_j(x) - a_j) = 0] = 1$ , and there are precisely  $p^{k-1}$  (other) vectors  $(b'_1, \dots, b'_k) \in \mathbb{F}_p^k$  such that  $[\sum_{j=1}^k b'_j \cdot (p_j(x) - a_j) = 1] = 1$ . These two equal counts cancel out in the sum of (6), so non-solutions to the system contribute 0 to the sum of (6).

Putting (5) and (6) together, the original Sum-Product problem (3) can now be reduced to the computation of  $O(p^{2k})$  sums, each of the form

$$\sum_{x \in \{0,1\}^n} [q(x_1, \dots, x_n) = 0],$$

where  $q$  is an  $\mathbb{F}_p$ -polynomial of degree at most  $d$ . That is, to obtain (3), we only need to count the Boolean roots of  $O(p^{2k})$  polynomials  $q$ , and take the appropriate  $\mathbb{R}$ -linear combination of these counts.

Let us now focus on counting roots to a single polynomial  $q(x_1, \dots, x_n)$  of degree  $d$ . Let  $P_\ell(z)$  be the modulus-amplifying polynomial of degree  $2\ell - 1$ , from Theorem 12. Let  $\delta \in (0, 1/2)$  be a parameter, and consider the following “reduced” polynomial in  $n - \delta n$  variables, over the integers:

$$Q(x_1, \dots, x_{n-\delta n}) := \sum_{a_1, \dots, a_{\delta n} \in \{0,1\}} P_{\delta n}(1 - q(x_1, \dots, x_{n-\delta n}, a_1, \dots, a_{\delta n})^{p-1}).$$

Note that  $Q$  has degree less than  $2dp\delta n$ . Set  $\delta = 1/(6dp)$ , and note that  $2dp\delta n < (n - \delta n)/2$ . Over  $\mathbb{F}_p$ , the polynomial  $1 - q(x)^{p-1}$  equals 1 mod  $p$  if  $x$  is a root of  $q$ , and is 0 mod  $p$  otherwise. Applying the modulus-amplifying properties of  $P_{\delta n}$ , we have:

- If  $x$  is a root of  $q$ , then  $P_{\delta n}(1 - q(x)^{p-1}) = 1 \bmod p^{\delta n}$ .
- If  $x$  is not a root of  $q$ , then  $P_{\delta n}(1 - q(x)^{p-1}) = 0 \bmod p^{\delta n}$ .

As the sum in  $Q$  is over only  $2^{\delta n}$  such  $P_{\delta n}(\dots)$  terms, and  $p \geq 2$ , we conclude that for all  $b_1, \dots, b_{n-\delta n} \in \{0,1\}$ , the quantity  $(Q(b_1, \dots, b_{n-\delta n}) \bmod p^{\delta n})$  equals the number of  $a_1, \dots, a_{\delta n} \in \{0,1\}$  such that

$$q(b_1, \dots, b_{n-\delta n}, a_1, \dots, a_{\delta n}) = 0.$$

Therefore if we evaluate the polynomial  $Q$  over all  $2^{n-\delta n}$  Boolean assignments  $(b_1, \dots, b_{n-\delta n})$ , compute each value separately modulo  $p^{\delta n}$ , then sum those values over the integers, we will obtain the number of Boolean roots of  $q$ .

Over Boolean assignments, we may assume without loss of generality that  $Q$  is multilinear (i.e.  $x_i^2 = x_i$  for all  $i$ ). Since  $2dp\delta n < (n - \delta n)/2$ , standard properties of binomial coefficients imply that the number of monomials of  $Q$  is

$$O\left(\binom{n - \delta n}{2dp\delta n}\right).$$

By constructing  $Q$  term-by-term (expanding each  $P_{\delta n}(1 - q(x_1, \dots, x_{n-\delta n}, a_1, \dots, a_{\delta n})^{p-1})$  one-by-one, and adding them to a running sum, similar to [12, 29]), we may represent  $Q$  as a sum of  $O\left(\binom{n - \delta n}{2dp\delta n}\right)$  monomials, constructed in  $\text{poly}(n) \cdot \binom{n - \delta n}{2dp\delta n}$  time. Letting  $\delta = 1/(6dp)$ , the number of monomials of  $Q$  is less than  $\binom{n}{n/3} \leq 1.9^n$ . Applying the fast polynomial evaluation algorithm of Theorem 13,  $Q$  can be evaluated on all  $2^{n-n/(6dp)}$  Boolean assignments in time  $(1.9^n + 2^{n-n/(6dp)}) \cdot \text{poly}(n)$  time. ◀

Therefore, for every *fixed* degree  $d$  and prime  $p$ , there is an  $\varepsilon > 0$  such that the relevant Sum-Product problem is in  $2^{n-\varepsilon n} \cdot \text{poly}(n)$  time. This immediately implies the lower bounds of Theorems 4 and 5. In particular, to prove 5 we apply Theorem 21. Fix an integer degree  $d$ , and let  $c \geq 1$  be the universal constant (from Theorem 21) such that we need to solve Sum-Product for  $\text{MOD}_p \circ \text{AND}_d \circ \text{ANY}_c$  circuits. Converting to  $\text{SUM} \circ \text{MOD}_p \circ \text{AND}_{dc}$ , Theorem 26 says that the Sum-Product problem can be solved in  $2^{n-n/O(dc)}$  time (omitting low-order terms).



## 7 Conclusion

Applying old and new tools, we have established several strong new lower bounds for representing Boolean functions in different regimes. Among the most interesting open problems remaining, we find the Quadratic Uncertainty Principle (the conjecture that AND requires large  $\mathbb{R}$ -linear combinations of quadratic  $\mathbb{F}_2$ -polynomials) to be especially intriguing. Quadratic polynomials have special properties that higher degrees do not; for example, one can count the roots of a given quadratic  $\mathbb{F}_p$ -polynomial in *polynomial time* (see [54] for a recent application of this phenomenon). Therefore in some cases, our  $2^{n-\varepsilon n}$ -time algorithms become  $\text{poly}(n)$ -time algorithms. Intuitively, an extremely efficient counting algorithm *should* imply lower bounds for functions *in polynomial time* against linear combinations of quadratic  $\mathbb{F}_2$ -polynomials, perhaps even lower bounds against the AND function, but so far we have not yet been able to prove such bounds.

### The Constant Degree Hypothesis?

A longstanding problem in circuit complexity – seemingly related to the Quadratic Uncertainty Principle – is the Constant Degree Hypothesis of Barrington, Straubing, and Thérien [5]:

► **Hypothesis 27** (Constant Degree Hypothesis (CDH)). *For every constant  $d \geq 1$  and primes  $p, q$ , there is an  $\varepsilon > 0$  such that the AND function on  $n$  variables cannot be computed by  $\text{MOD}_p \circ \text{MOD}_q \circ \text{AND}_d$  circuits of  $2^{\varepsilon n}$  size.*

The CDH is currently only known to be true for  $d = 1$ , and for  $p = q$ . Can the techniques of this paper say anything about such problems, even for the case of  $d = 2$ ?

### Split-and-List as a Lower Bound Technique?

As noted by a CCC reviewer, the algorithmic approaches applied in this paper (in particular, the “split-and-list” paradigm [18]) were essentially known in the literature, and yet they were already powerful enough to prove strong lower bounds against functions in NP. Is there a more direct method for proving circuit lower bounds that “corresponds” to the split-and-list paradigm, *without* having to go through a generic connection between SAT algorithms and circuit lower bounds?

Intuitively, the algorithmic split-and-list paradigm is related to communication complexity. In split-and-list, the variable space of an instance is “split” into smaller parts, and we find a global solution to the instance by “listing” partial solutions to the variables, and combining partial solutions together in some interesting way. This feels related to the situation where multiple parties hold parts of a global input, and they communicate to determine if the global input is a solution to some problem. Indeed, intuitive connections between the two have been successfully made in several papers, and articulated fairly strongly in [48, 37, 1].

However, there is a sense in which split-and-list seems more powerful. A good example is the algorithm for Subset-Sum: it splits the variables of the solution space into two parts, and uses the ability to *quickly and deterministically sort and search* the list of  $2^{n/2}$  partial solutions to find a Subset-Sum faster. In contrast, deterministic communication between two parties holding  $n/2$  bits each (with public knowledge of a Subset-Sum instance of  $n$  items) cannot always determine with low communication if their joint  $n$ -bit assignment is a solution to the instance. (When the weights of the instance are exponentially large, the communication problem becomes as hard as EQUALITY.)

## References

- 1 Amir Abboud, Aviad Rubinfeld, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In *FOCS*, pages 25–36, 2017.
- 2 Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *FOCS*, pages 467–476, 2016.
- 3 Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.
- 4 László Babai, Kristoffer Arnsfelt Hansen, Vladimir V. Podolskii, and Xiaoming Sun. Weights of exact threshold functions. In *Mathematical Foundations of Computer Science*, pages 66–77, 2010.
- 5 David A. Mix Barrington, Howard Straubing, and Denis Thérien. Non-uniform automata over groups. *Inf. Comput.*, 89(2):109–132, 1990.
- 6 Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, pages 350–366, 1994.
- 7 Eli Ben-Sasson and Emanuele Viola. Short PCPs with projection queries. In *ICALP*, pages 163–173, 2014.
- 8 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009.
- 9 Jean Bourgain. Estimation of certain exponential sums arising in complexity theory. *C.R. Acad. Sci. Paris Ser. I*, 340:627–631, 2005.
- 10 Jin-yi Cai, Frederic Green, and Thomas Thierauf. On the correlation of symmetric functions. *Mathematical Systems Theory*, 29(3):245–258, 1996.
- 11 Chris Calabro. A lower bound on the size of series-parallel graphs dense in long paths. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(110), 2008.
- 12 Timothy M. Chan and Ryan Williams. Deterministic APSP, Orthogonal Vectors, and more: Quickly derandomizing Razborov-Smolensky. In *SODA*, pages 1246–1255, 2016.
- 13 Arkadev Chattopadhyay and Nikhil S. Mande. Weights at the bottom matter when the top is heavy. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:83, 2017.
- 14 Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. In *CCC*, pages 1:1–1:35, 2016.
- 15 Amit Daniely. Depth separation for neural networks. In *Proceedings of COLT*, pages 690–696, 2017.
- 16 Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Proceedings of COLT*, pages 907–940, 2016.
- 17 Yuval Filmus, Hamed Hatami, Steven Heilman, Elchanan Mossel, Ryan O’Donnell, Sushant Sachdeva, Andrew Wan, and Karl Wimmer. Real Analysis in Computer Science: A collection of open problems, Simons Institute, 2014. URL: <https://simons.berkeley.edu/sites/default/files/openprobsmerged.pdf>.
- 18 Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Springer, 2010.
- 19 Anna Gál and Vladimir Trifonov. On the correlation between parity and modular polynomials. *Theory Comput. Syst.*, 50(3):516–536, 2012.
- 20 Frederic Green. The correlation between parity and quadratic polynomials mod 3. *Journal of Computer and System Sciences*, 69(1):28–44, 2004.
- 21 Jacques Hadamard. Résolution d’une question relative aux déterminants. *Bull. Sci. Math.*, 17:30–31, 1893.
- 22 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993.
- 23 Kristoffer Arnsfelt Hansen and Vladimir V Podolskii. Exact threshold circuits. In *CCC*, pages 270–279, 2010.
- 24 Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

- 25 Hamed Hatami, Pooya Hatami, and Shachar Lovett. Higher-order Fourier analysis and applications. Manuscript, 2016. URL: [https://cseweb.ucsd.edu/~slovett/files/survey-higher\\_order\\_fourier.pdf](https://cseweb.ucsd.edu/~slovett/files/survey-higher_order_fourier.pdf).
- 26 Ellis Horowitz and Sartaj Sahni. Computing partitions with applications to the knapsack problem. *JACM*, 21(2):277–292, 1974.
- 27 Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. In *Proceedings of ICALP*, pages 749–760, 2015.
- 28 Daniel M. Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *STOC*, pages 633–643, 2016.
- 29 Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, R. Ryan Williams, and Huacheng Yu. Beating brute force for systems of polynomial equations over finite fields. In *SODA*, pages 2190–2202, 2017.
- 30 Shachar Lovett. Personal communication, 2017.
- 31 Wolfgang Maass. Bounds for the computational power and learning complexity of analog neural nets. *SIAM Journal on Computing*, 26(3):708–732, 1997.
- 32 Peter Bro Miltersen, N. V. Vinodchandran, and Osamu Watanabe. Super-polynomial versus half-exponential circuit size in the exponential hierarchy. In *COCOON, Springer LNCS 1627*, pages 210–220, 1999.
- 33 Anirbit Mukherjee and Amitabh Basu. Lower bounds over Boolean inputs for deep neural networks with ReLU gates. *ArXiv e-prints*, 2017. arXiv:1711.03073.
- 34 S. Muroga, I. Toda, and S. Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271:376–418, 1961.
- 35 Cody Murray and Ryan Williams. Circuit lower bounds for nondeterministic quasipolytime: An easy witness lemma for NP and NQP. *Electronic Colloquium on Computational Complexity (ECCC)*, TR17-188, 2017.
- 36 Noam Nisan. The communication complexity of threshold gates. In *Proceedings of “Combinatorics, Paul Erdos is Eighty”*, pages 301–315, 1994.
- 37 Mihai Pătraşcu and Ryan Williams. On the possibility of faster sat algorithms. In *SODA*, pages 1065–1075, 2010.
- 38 Vwani P. Roychowdhury, Alon Orlitsky, and Kai-Yeung Siu. Lower bounds on threshold and related circuits via communication complexity. *IEEE Transactions on Information Theory*, 40(2):467–474, 1994.
- 39 Itay Safran and Ohad Shamir. Depth-width tradeoffs in approximating natural functions with neural networks. In *International Conference on Machine Learning*, pages 2979–2987, 2017.
- 40 Rahul Santhanam. Circuit lower bounds for Merlin–Arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009.
- 41 Rahul Santhanam and Ryan Williams. On medium-uniformity and circuit lower bounds. In *IEEE Conf. Computational Complexity*, pages 15–23, 2013.
- 42 Joel Seiferas, Michael Fischer, and Albert Meyer. Separating nondeterministic time complexity classes. *Journal of the ACM*, 25(1):146–167, jan 1978.
- 43 Suguru Tamaki. A satisfiability algorithm for depth two circuits with a sub-quadratic number of symmetric and threshold gates. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:100, 2016.
- 44 Matus Telgarsky. benefits of depth in neural networks. In *Proceedings of COLT*, pages 1517–1539, 2016.
- 45 Roei Tell. Proving that  $\text{prBPP}=\text{prP}$  is as hard as “almost” proving that  $P \neq NP$ . *Electronic Colloquium on Computational Complexity (ECCC)*, 18(3), 2018.
- 46 S. Toda. PP is as hard as the polynomial-time hierarchy. *sicomp*, 20(5):865–877, 1991.

- 47 L. G. Valiant. Graph-theoretic arguments in low-level complexity. In J. Gruska, editor, *MFCS*, volume 53 of *LNCS*, pages 162–176, Tatranská Lomnica, Czechoslovakia, sep 1977. Springer.
- 48 Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM Journal on Computing*, 42(3):831–854, 2013.
- 49 Emanuele Viola. Guest column: correlation bounds for polynomials over  $\{0, 1\}$ . *SIGACT News*, 40(1):27–44, 2009.
- 50 Ryan Williams. A casual tour around a circuit complexity bound. *SIGACT News*, 42(3):54–76, 2011.
- 51 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *sicomp*, 42(3):1218–1244, 2013.
- 52 Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *STOC*, pages 194–202, 2014.
- 53 Ryan Williams. Nonuniform ACC circuit lower bounds. *JACM*, 61(1):2, 2014.
- 54 Ryan Williams. Counting solutions to polynomial systems via reductions. In Raimund Seidel, editor, *1st Symposium on Simplicity in Algorithms (SOSA 2018)*, pages 6:1–6:15, 2018.
- 55 R. O. Winder. *Threshold Logic*. PhD thesis, Princeton University, 1962. Preliminary version in FOCS’60.
- 56 Stanislav Žák. A Turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327–333, 1983.

## A

 Linear Lower Bound for AND With Sums of Quadratic Polynomials

For reference, we report a folklore  $\Omega(n)$  lower bound on representing AND with linear combinations of quadratic  $\mathbb{F}_2$ -polynomials (recall it is conjectured that the sparsity lower bound is  $2^{\Omega(n)}$ ). The below proof was communicated to us by Shachar Lovett.

► **Theorem 28** (Lovett [30]). *The AND function on  $n$  inputs does not have  $\text{SUM} \circ \text{MOD}_2 \circ \text{AND}_2$  circuits of sparsity less than  $n/2$ .*

**Proof.** Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be the NOR function (which by DeMorgan’s laws has the same sparsity as AND). Suppose we can write

$$f(x) = \sum_{i=1}^s \alpha_i (-1)^{q_i(x)},$$

where the  $q_i(x)$  are quadratic  $\mathbb{F}_2$ -polynomials, and all  $\alpha_i \in \mathbb{R}$ . Note that without loss of generality we may assume  $q_i(0) = 0$  for all  $i$  (if  $q_i(0) = 1$ , then replacing  $\alpha_i$  by  $-\alpha_i$  and  $q_i(x)$  by  $q_i(x) + 1$  yields an equivalent expression). If  $s < n/2$ , then by the Chevalley–Warning theorem, the number of common roots of  $\{q_1, \dots, q_r\}$  is divisible by 2. But then there is another common root  $x^*$ , so  $f(0) = f(x^*)$ , contradicting the definition of NOR. ◀

# The Power of Natural Properties as Oracles

**Russell Impagliazzo**

Department of Computer Science, University of California San Diego, La Jolla, CA, USA  
russell@cs.ucsd.edu

**Valentine Kabanets**

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

**Ilya Volkovich**

Department of EECS, CSE Division, University of Michigan, Ann Arbor, MI, USA  
ilyavol@umich.edu

---

## Abstract

We study the power of randomized complexity classes that are given oracle access to a natural property of Razborov and Rudich (*JCSS*, 1997) or its special case, the Minimal Circuit Size Problem (MCSP). We show that in a number of complexity-theoretic results that use the SAT oracle, one can use the MCSP oracle instead. For example, we show that  $ZPEXP^{MCSP} \not\subseteq P/poly$ , which should be contrasted with the previously known circuit lower bound  $ZPEXP^{NP} \not\subseteq P/poly$ . We also show that, assuming the existence of Indistinguishability Obfuscators (IO), SAT and MCSP are equivalent in the sense that one has a ZPP algorithm if and only the other one does. We interpret our results as providing some evidence that MCSP may be NP-hard under randomized polynomial-time reductions.

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

**Keywords and phrases** natural properties, Minimal Circuit Size Problem (MCSP), circuit lower bounds, hardness of MCSP, learning algorithms, obfuscation, Indistinguishability Obfuscators (IO)

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.7

**Acknowledgements** We thank Eric Allender and Scott Aaronson for answering our questions and many useful conversations. We also thank the anonymous referees for their useful comments.

## 1 Introduction

Historically, the problem of minimizing a circuit representing a given Boolean function (MCSP) was one of the first where the prohibitive computational cost of searching through a huge space of candidate solutions was noted [30, 44]. This issue would later be formalized in the theory of NP-completeness. However, the complexity of circuit minimization itself remains largely mysterious. It is an NP problem, but neither known to be NP-complete nor in any sub-class of NP thought proper. This mystery remains despite a large body of work devoted to this problem [28, 2, 4, 3, 5, 24, 38, 23].

For negative hardness results, we do know that MCSP is *not* NP-hard (even P-hard) under very restrictive reductions [38]. We also know that MCSP is not NP-hard under certain “black-box” reductions [23]. For other kinds of restricted reductions, we know that proving the NP-hardness of MCSP under such reductions would be difficult as such a proof would also yield new circuit lower bounds [28, 38, 5].



© R. Impagliazzo, V. Kabanets, and I. Volkovich;  
licensed under Creative Commons License CC-BY  
33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 7; pp. 7:1–7:20

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



On the other hand, for positive hardness result, we know that MCSP is SZK-hard under general randomized (BPP) reductions [3], and  $NC^1$ -hard under truth-table reductions computable by non-uniform  $TC^0$  circuits [40].

Looking at the negative results about NP-hardness of MCSP, one has to wonder: Are these results actually about MCSP and its relationship to other problems, or about the weakness of certain types of reductions? Given the positive results about hardness of MCSP under more powerful reductions, it seems more likely that the aforementioned negative hardness results are in fact about the weakness of certain reductions, and that it may be the case that MCSP is NP-hard under, say, general randomized polynomial-time reductions.

We seem to be very far from being able to prove the NP-hardness of MCSP. If we cannot prove that MCSP is as hard as SAT, can we find other evidence that MCSP is indeed a hard problem, or at least that it will be difficult to design an efficient algorithm for it?

One possible kind of evidence that MCSP may be “almost as hard as” SAT would be to show that many known complexity-theoretic statements that use the SAT oracle will remain true when the SAT oracle is replaced with the MCSP oracle, i.e., that *the power of the MCSP oracle is often as good as that of SAT*. This is the research direction pursued in the present paper.

## 1.1 Our results

While, for simplicity, we state our results below for MCSP, in most of our results, MCSP could be replaced with any other natural property in the sense of Razborov and Rudich [41] (having largeness and usefulness, but with oracle access replacing constructivity). Roughly, our results are of three kinds:

- circuit lower bounds for randomized complexity classes with MCSP oracle,
- relations between Indistinguishability Obfuscation (IO) and MCSP, and
- hardness results for relativized versions of MCSP under randomized reductions.

We provide a more detailed description of our results next.

### 1.1.1 Conditional collapses

Below, the notation  $SIZE[s]$  denotes the class of Boolean functions computable by size  $s$  Boolean circuits.

► **Theorem 1.** *Let  $\Gamma \in \{\oplus P, P^{\#P}, PSPACE, EXP, NEXP, EXP^{NP}\}$ . If  $\Gamma \subseteq P/poly$ , then  $\Gamma \subseteq ZPP^{MCSP}$ .*

#### 1.1.1.1 Interpretation

The results of [36], [8], [25] and [15] (building upon [31]) imply collapse theorems for the classes  $P^{\#P}$ , PSPACE and EXP, NEXP,  $EXP^{NP}$ , respectively. More specifically, they show that if any of the above classes has polynomial size Boolean circuits, then the corresponding class collapses to MA, which is known to be contained in  $ZPP^{NP}$  [7, 20]. Our Theorem 1 shows that the power of the MCSP oracle is sufficient for these conditional collapses.

As it is also known that  $MA \subseteq NP^{MCSP}$  (see, e.g., [2]), the conditional collapses to  $NP^{MCSP}$  are immediate. Our Theorem 1 strengthens these collapses to the potentially smaller class  $ZPP^{MCSP}$ .

Finally, we also interpret Theorem 1 as follows: *A proof that MCSP is not NP-hard (or even  $\#P$ -hard) under Turing ZPP-reductions would imply that  $P^{\#P} \not\subseteq P/poly$ .*

### 1.1.2 Circuit lower bounds

Given the collapse theorems above, we get fixed-polynomial and super-polynomial lower bounds for randomized polynomial and exponential time, respectively. The extra bit of advice in the case of randomized polynomial time comes to accommodate the need to keep the promise of bounded error (the same problem arises in [10, 19, 37, 42, 49]). Alternatively, we can consider the corresponding class of promise problems (i.e.,  $\text{prZPP}$ ).

► **Theorem 2.** *We have the following:*

1.  $\text{ZPP}^{\text{MCSP}}/1 \not\subseteq \text{SIZE}[n^k]$  and  $\text{prZPP}^{\text{MCSP}} \not\subseteq \text{SIZE}[n^k]$ , for all  $k \in \mathbb{N}$ .
2.  $\text{ZPEXP}^{\text{MCSP}} \not\subseteq \text{P/poly}$ .

#### 1.1.2.1 Interpretation

It is known that  $\text{MA-EXP} \not\subseteq \text{P/poly}$  [14]. By padding, we get that  $\text{MA-EXP} \subseteq \text{ZPEXP}^{\text{NP}}$  (using  $\text{MA} \subseteq \text{ZPP}^{\text{NP}}$  [7, 20]), and hence  $\text{ZPEXP}^{\text{NP}} \not\subseteq \text{P/poly}$ . Theorem 2 (item 2) shows that the MCSP oracle can replace the SAT oracle in that latter circuit lower bound.

#### 1.1.2.2 Consequences for natural properties

The above result still holds if we relax the MCSP oracle to a natural property strongly useful against  $\text{P/poly}$  (see Theorem 41 for more details). Combining this result with Lemma 25, we obtain that PAC learning algorithms imply fixed-polynomial lower bounds against  $\text{BPP}/1$  and super polynomial lower bounds against  $\text{BPEXP}$ . These bounds match the results of [49] and [18, 32], respectively (see Corollary 26 for more details). In this sense, our *unconditional* lower bounds generalize the *conditional* lower bounds of [49] and [18, 32]. Indeed, our result is obtained by extending the techniques of [18, 32, 49].

The following theorem should be contrasted with a result from [25] saying that the existence of a P-natural property (even *without* the largeness condition) that is useful against  $\text{P/poly}$  would imply that  $\text{NEXP} \not\subseteq \text{P/poly}$ . *With* the largeness condition, the circuit lower bound can be shown to hold for the potentially smaller uniform complexity class  $\text{ZPEXP}$ . This theorem is an immediate consequence of Theorem 2, item (2).

► **Theorem 3.** *Suppose there is a strongly useful ZPP-natural property. Then  $\text{ZPEXP} \not\subseteq \text{P/poly}$ .*

► **Remark.** The conclusion of Theorem 3 still holds if we assume a natural property with only weakly-exponential usefulness,  $2^{n^{\Omega(1)}}$ .

► **Corollary 4.** *If there is a ZPP-natural property that is weakly-exponentially useful against  $\text{ACC}^0$  circuits, then  $\text{ZPEXP} \not\subseteq \text{ACC}^0$ .*<sup>1</sup>

### 1.1.3 Obfuscation

We also relate the powers of MCSP and SAT to the existence of indistinguishability obfuscators (IO) [11]. Roughly speaking, an IO is an efficient randomized procedure that maps circuits to circuits, preserving the circuit input-output functionality but in an “unintelligible” manner. Indeed, applying the IO to any two functionally equivalent circuits of the same size yields two indistinguishable distributions on circuits (see Definition 27 for more details). We show the following.

<sup>1</sup> The result that P-natural properties against sub-exponential size circuits yield  $\text{ZPEXP}$  lower bounds was also obtained in independent work by Igor Oliveira and Rahul Santhanam [40].



► **Theorem 5.** *Let  $\mathcal{A}$  denote the class of randomized polynomial-time algorithms with MCSP oracle. If there exists an  $\mathcal{A}$ -indistinguishable obfuscator IO then  $\text{NP} \subseteq \text{ZPP}^{\text{MCSP}}$ .*

► **Corollary 6.** *Suppose a computational obfuscator IO exists. Then  $\text{MCSP} \in \text{ZPP}$  iff  $\text{NP} = \text{ZPP}$ .*

**Proof.** If  $\text{NP} = \text{ZPP}$  then, clearly,  $\text{MCSP} \in \text{ZPP}$ . For the other direction, if  $\text{MCSP} \in \text{ZPP}$  then IO is also an  $\mathcal{A}$ -indistinguishable obfuscator. Therefore,  $\text{NP} \subseteq \text{ZPP}^{\text{MCSP}} = \text{ZPP}^{\text{ZPP}} = \text{ZPP}$ . ◀

### 1.1.3.1 Interpretation

Corollary 6 says that, under a cryptographic assumption that computational IO exists, the computational powers of SAT and MCSP are the same in the sense that a ZPP algorithm for MCSP is as good as a ZPP algorithm for SAT.

### 1.1.4 Hardness of relativized versions of MCSP

We consider the relativized version of MCSP relative to an oracle  $A$ , denoted  $\text{MCSP}^A$ , which asks to determine the minimum circuit size for a given Boolean function (given by its truth table) where the circuit is allowed to use  $A$ -oracle gates. It is shown by [2] that every language in PSPACE is reducible to  $\text{MCSP}^{\text{PSPACE}}$  via ZPP-reductions. We use different techniques to re-prove this result, as well as obtain a few new results along the same lines. (Below  $C_k\text{P}$  is the  $k$ th level of the counting hierarchy, CH, where  $C_1\text{P} = \text{PP}$ , and  $C_{k+1}\text{P} = \text{PP}^{C_k\text{P}}$ , for all  $k \geq 1$ .)

- **Theorem 7.** 1.  $\text{PSPACE} \subseteq \text{ZPP}^{\text{MCSP}^{\text{PSPACE}}}$  [2]  
 2.  $\oplus\text{P} \subseteq \text{ZPP}^{\text{MCSP}^{\oplus\text{P}}}$   
 3.  $\text{P}^{\#\text{P}} \subseteq \text{BPP}^{\text{MCSP}^{\#\text{P}}}$   
 4.  $\text{PP} \subseteq \text{BPP}^{\text{MCSP}^{\text{PP}}}$ . Moreover, for  $k \geq 2$ :  $C_k\text{P} \subseteq C_{k-1}\text{P}^{\text{MCSP}^{\text{PP}}}$ .

#### 1.1.4.1 Interpretation

All of the inclusions of Theorem 7 become trivial if one replaces the relativized MCSP problem with the relativized SAT problem (or even just some relativized P-complete problem), since we have trivially that, e.g.,  $\text{PSPACE} \subseteq \text{P}^{\text{PSPACE}}$ . Theorem 7 says that the circuit minimization problem for circuits with  $A$ -oracle gates (for certain kinds of oracles) is at least as hard as the evaluation problem for  $A$ , under sufficiently powerful (randomized) reductions.

In [23], Hirahara and Watanabe defined the notion of oracle-independent randomized reductions and initiated a study of the set of languages that are reducible in randomized polynomial time to  $\text{MCSP}^B$  for every  $B$ . As a part of their study, they showed that  $\bigcap_B \text{BPP}^{\text{MCSP}^B[1]} \subseteq \text{AM} \cap \text{coAM}$ ; this implies that NP-hardness of MCSP cannot be established via oracle-independent reductions unless the polynomial hierarchy collapses. We show circuit lower bounds for the class  $\bigcap_B \text{BPP}^{\text{MCSP}^B}$ .

► **Theorem 8.** *We have that  $\bigcap_B \text{BPP}^{\text{MCSP}^B} / 1 \not\subseteq \text{SIZE}[n^k]$  and  $\bigcap_B \text{prBPP}^{\text{MCSP}^B} \not\subseteq \text{SIZE}(n^k)$ , for all  $k \in \mathbb{N}$ , and that  $\bigcap_B \text{BEXP}^{\text{MCSP}^B} \not\subseteq \text{P/poly}$ .*



## 1.2 Our techniques

We rely on the result of [16] showing that natural properties useful against a (sufficiently powerful) circuit class  $\mathcal{C}$  yield learning algorithms (under the uniform distribution, with membership queries) for the same circuit class. We note that this result relativizes in the following sense: if we have a natural property useful against circuits with  $L$  oracle gates (say,  $\text{MCSP}^L$ ), for some language  $L$ , then we can approximately learn  $L$ , with the hypotheses being circuits with  $\text{MCSP}^L$  oracle gates. If, in addition, this language  $L$  is both downward and random self-reducible, then we can learn  $L$  exactly, with the same type of  $\text{MCSP}^L$  oracle circuits, using the ideas of [27].

This allows us to prove, for example, that  $\text{P}^{\#\text{P}} \subseteq \text{BPP}^{\text{MCSP}^{\#\text{P}}}$ , as  $\#\text{P}$  has a complete problem (the permanent) that is well-known to be both downward and random self-reducible. We show that  $\oplus\text{P}$  also has such a complete problem (building upon [45]), getting the inclusion  $\oplus\text{P} \subseteq \text{BPP}^{\text{MCSP}^{\oplus\text{P}}}$ . To get the stronger result that  $\oplus\text{P} \subseteq \text{ZPP}^{\text{MCSP}^{\oplus\text{P}}}$ , we use Toda's Theorem [43] and hardness-randomness tradeoffs of [26] to get rid of the two-sided error of our BPP reduction (similarly to the work of [28]).

Our circuit lower bounds are proved using similar ideas. For example,  $\text{ZPEXP}^{\text{MCSP}} \not\subseteq \text{P}/\text{poly}$  is argued as follows. If  $\text{PSPACE} \not\subseteq \text{P}/\text{poly}$ , we are done (as  $\text{PSPACE} \subseteq \text{EXP}$ ). Assuming  $\text{PSPACE} \subseteq \text{P}/\text{poly}$ , we get that  $\text{PSPACE} \subseteq \text{ZPP}^{\text{MCSP}}$ , using the fact that  $\text{PSPACE}$  contains a complete problem that is both downward and random self-reducible [45], and that  $\text{MCSP}^{\text{PSPACE}} \in \text{PSPACE} \subseteq \text{P}/\text{poly}$ . The circuit lower bound then follows by a translation argument, as we get that  $\text{EXPSPACE} \subseteq \text{ZPEXP}^{\text{MCSP}}$  and  $\text{EXPSPACE}$  is known to contain languages of maximal circuit complexity (by a simple diagonalization argument).

As another consequence of the results in [16], we get the following.

► **Theorem 9.** *For any language  $B$ ,  $n \in \mathbb{N}$  and  $\delta > 0$ , there exists a  $\text{MCSP}^B$ -oracle circuit  $C$  of size  $\text{poly}(n, 1/\delta)$  that is  $1 - \delta$  close to  $B|_n$ . If, in addition,  $B$  is self-correctable then  $B$  has polynomial size  $\text{MCSP}^B$ -oracle circuits.*

► **Theorem 10.** *Let  $B$  be a language such that  $\text{PSPACE}^B$  has polynomial size  $B$ -oracle circuits. Then  $B$  has polynomial-size  $\text{MCSP}^B$ -oracle circuits.<sup>2</sup>*

For the indistinguishability related results, we combine ideas from [21, 35] with a result from [2]. Let  $\perp_s$  denote a canonical circuit of size  $s$  that outputs '0' on every input. Let  $\mathcal{A}$  denote the class of randomized polynomial-time algorithms with  $\text{MCSP}$  oracle. Given an  $\mathcal{A}$ -indistinguishable obfuscator  $\text{IO}$ , we consider the function  $f_s(r) = \text{IO}(\perp_s, r)$ , where  $r$  is a random string. Observe that for any  $s$ , the function  $f_s(r)$  is computable in time polynomial in  $|r|$ . We then apply a result of [2] that allows us to find preimages of such functions with probability  $1/\text{poly}(n)$ .

Given a circuit  $C$  of size  $s$ , we first compute an obfuscation of  $C$ ,  $\hat{C} = \text{IO}(C, r)$ , (for a random  $r$ ). Next, we (attempt to) find a preimage  $r'$  of  $\hat{C}$ . That is,  $r'$  such that  $\text{IO}(\perp_s, r') = \hat{C}$ . We accept if and only if  $r'$  is indeed a preimage. That is, if and only if  $\text{IO}(\perp_s, r') = \hat{C}$ .

We observe the following:

- If  $C = \perp_s$  then the algorithm will accept with probability  $1/\text{poly}(n)$ .

<sup>2</sup> In [2], the same outcome was achieved under a stronger assumption that  $\text{PSPACE}^B \subseteq \text{P}^B$ . We note our result is not a mere syntactical improvement, as there are numerous languages  $B$  for which  $\text{PSPACE}^B \subseteq \text{P}^B/\text{poly}$  yet  $\text{PSPACE}^B \neq \text{P}^B$ ; see Appendix C for more details. While we suspect that the consequent of the theorem holds unconditionally, we note that the precondition statement of the theorem cannot be improved further since Lemma 19 implies that, for every language  $B$ , the class  $\text{PSPACE}^B$  does not have fixed-polynomial size  $B$ -oracle circuits.

- If  $C$  is satisfiable then by the correctness requirement of IO (Requirement 2) for all  $r, r'$ :  $\text{IO}(C, r) \neq \text{IO}(\perp_s, r')$ . Therefore, the algorithm will always reject.
- Finally, if  $C$  is an *unsatisfiable* circuit of size  $s$ , then by the indistinguishability requirement (Requirement 3) the algorithm cannot distinguish between the obfuscation of  $\perp_s$  and the obfuscation of  $C$ . Hence, the algorithm will accept with probability about  $1/\text{poly}(n)$ .

Overall, we obtain that  $\overline{\text{SAT}} \in \text{RP}^{\text{MCSP}}$ .

### 1.2.0.1 Remainder of the paper

We give basic definitions and notation in Section 2. In Section 3, we prove our main results (Theorems 1 - 8) which show new collapse results as well as new circuit lower bounds for uniform complexity classes with oracle access to (relativized) MCSP. In fact, we prove somewhat stronger results (Theorems 40 and 41) which apply to the more general type of oracles: strongly useful natural properties. We prove our IO-related result, Theorem 5, in Section 3.3. Next, in Section 3.4, we prove our results about reductions to the problem  $\text{MCSP}^B$ , for various languages  $B$ . Specifically, we give such reductions for several complexity classes (Theorem 7), and also show that every language  $B$  can be approximated by “small” Boolean circuits containing  $\text{MCSP}^B$  oracle gates (Theorem 9). Finally, we show that under certain conditions, a language  $B$  can be computed exactly by “small” Boolean circuits containing  $\text{MCSP}^B$  oracle gates (rather than just approximated) (Theorem 10). We conclude with some open questions in Section 4. Some of the proofs (e.g., our proof that  $\oplus\text{P}$  has a complete problem that is both downward and random self-reducible) are given in the appendix.

## 2 Preliminaries

### 2.1 Basics

A function  $\text{negl}(n)$  is *negligible* if for any  $k \in \mathbb{N}$  there exists  $n_k \in \mathbb{N}$  such that for all  $n > n_k$ ,  $\text{negl}(n) < 1/n^k$ .

For Boolean functions  $f, g : \{0, 1\}^n \rightarrow \{0, 1\}$ , we define the *relative distance*  $\Delta(f, g)$  to be the fraction of inputs  $x \in \{0, 1\}^n$  where  $f(x) \neq g(x)$ . For  $\varepsilon \geq 0$ , we say that  $f$  is  $\varepsilon$ -close to  $g$  if  $\Delta(f, g) \leq \varepsilon$ , otherwise we say that  $f$  is  $\varepsilon$ -far from  $g$ .

Let  $L \subseteq \{0, 1\}^*$  be a language. We denote by  $L|_n$  the set of the strings of length  $n$  in  $L$ . We will associate a language  $L$  with a corresponding Boolean function in the natural way:  $L(x) = 1 \iff x \in L$ . We say that  $L$  has circuits of size  $a(n)$  and denote it by  $L \in \text{SIZE}(a(n))$  if for every  $n \in \mathbb{N}$  the function  $L|_n$  can be computed by a Boolean circuit of size  $\mathcal{O}(a(n))$ . The *circuit complexity*  $s_L(n)$  of  $L$  at length  $n$  is the smallest integer  $t$  such that there is a Boolean circuit of size  $t$  that computes  $L|_n$ . We similarly define  $s_L^B(n)$  to be the circuit complexity of  $L$  with respect to  $B$ -oracle circuits and  $\text{SIZE}^B(a(n))$ . We have the following easy observation.

► **Observation 11.** *Let  $A, B$  be two languages. Suppose that  $A \in \text{SIZE}^B(n^k)$  for some  $k \in \mathbb{N}$ . Then for every language  $L$ :  $s_L^B(n) \leq s_L^A(n)^{k+1}$ .*

A promise problem is a relaxation of a language, defined as follows.

► **Definition 12 (Promise Problems).**  $\Pi = (\Pi_{YES}, \Pi_{NO})$  is a *promise problem* if  $\Pi_{YES} \cap \Pi_{NO} = \emptyset$ . We say that a language  $L$  is *consistent* with  $\Pi$  iff  $x \in \Pi_{YES} \implies x \in L$  and  $x \in \Pi_{NO} \implies x \notin L$ . The containment of  $L$  outside of  $\Pi_{YES} \cup \Pi_{NO}$  can be arbitrary. We say that a set of languages  $\Gamma$  is consistent with a set of promise problems  $\Lambda$  iff for every  $\Pi \in \Lambda$  there is  $L \in \Gamma$  that is consistent with  $\Pi$ .

We refer the reader to [6] for the definitions of standard complexity classes such as P, ZPP, RP, BPP, NP, MA, PSPACE, etc. We say that a language  $L \in \text{BPP}/1$  if  $L$  can be decided by a BPP machine with an auxiliary *advice* bit  $b_n$  for each input of length  $n$ ; note that given the complement advice bit  $\bar{b}_n$ , the machine is not guaranteed to be a BPP machine (i.e., may not have bounded away acceptance and rejection probabilities on all inputs of length  $n$ ). We define ZPP/1 in a similar fashion.

We define a family of natural problems complete for prBPP relative to any oracle.

► **Definition 13** (Circuit Approximation). For a language  $B$ , define the following prBPP <sup>$B$</sup> -complete problem:  $\text{CA}^B \triangleq (\text{CA}_{YES}^B, \text{CA}_{NO}^B)$ , where

$$\text{CA}_{YES}^B = \{C \text{ is a } B\text{-oracle circuit} \mid \Delta(C, \bar{0}) \geq 3/4\},$$

and

$$\text{CA}_{NO}^B = \{C \text{ is a } B\text{-oracle circuit} \mid \Delta(C, \bar{0}) \leq 1/4\}.$$

To prove lower bounds against randomized classes with one bit of advice, we shall rely on the following definitions (and their extensions) from [42, 49].

► **Definition 14** (Padded Languages). Let  $L$  be a language. For  $k \in \mathbb{N}$  we define the padded version of  $L$ , denoted  $L'_k$ , to consist of the strings  $1^m x$  satisfying the following: (1)  $m$  is power of 2; (2)  $0 < r \triangleq |x| \leq m$ ; (3)  $x \in L$ ; and (4)  $s_L(r) \leq m^{2k}$ .

The main property of the padded languages is that, for every  $L$ , sufficiently small circuits for  $L'_k$  can be used to construct small circuits for  $L$ .

► **Lemma 15** ([42, 49]). Let  $k \in \mathbb{N}$ . Suppose  $L'_k \in \text{SIZE}[n^k]$ . Then  $s_L(n) = \mathcal{O}(n^{2k})$ .

The next lemma is implicit in [49]. We provide the proof for completeness.

► **Lemma 16**. Let  $\mathcal{R}$  be a strongly useful natural property and let  $L$  be a downward self-reducible and self-correctable language. Then, for all  $k \in \mathbb{N}$ , we have  $L'_k \in \text{BPP}^{\mathcal{R}}/1$ .

**Proof.** Let  $y = 1^m x$  be an input for  $L'_k$ . Conditions 1 and 2 of Definition 14 can be checked easily. As  $y$  has a unique interpretation, we use the advice bit to determine whether  $s_L(|x|) \leq m^{2k}$ . If the advice bit is 0 (i.e. “no”) we reject. Otherwise, we apply Lemma 39 with  $t = m^{2k}$  to decide if  $x \in L$ . ◀

We also need the following result that shows that a lower bound on ZPP/1 carries over to prZPP.

► **Lemma 17** ([42]). For every circuit function  $u(n)$ , if  $\text{ZPP}/1 \not\subseteq \text{SIZE}[u(n)]$ , then  $\text{prZPP} \not\subseteq \text{SIZE}[u(n)]$ .

Finally, we need the following collapse results and a simple circuit lower bound against PSPACE.

► **Lemma 18** ([8, 25, 15]). If  $\Gamma \in \{\text{EXP}, \text{NEXP}, \text{EXP}^{\text{NP}}\}$  is in P/poly, then  $\Gamma = \text{MA}$ .

► **Lemma 19** ([29]). For any language  $B$  and  $k \in \mathbb{N}$ ,  $\text{PSPACE}^B \not\subseteq \text{SIZE}^B[n^k]$ . More generally, for every function  $s(n) = \mathcal{O}(2^n)$ ,  $\text{DSPACE}^B(\text{poly}(s(n))) \not\subseteq \text{SIZE}^B[s(n)]$ .

► **Lemma 20** (Folklore). For any oracle  $A$ :  $\text{NP} \subseteq \text{BPP}^A \implies \text{NP} \subseteq \text{RP}^A$ .

## 2.2 Derandomization from hardness

We recall the celebrated hardness-randomness tradeoff.

► **Lemma 21** ([39, 9, 26, 46, 33]). *There is a polynomial-time computable oracle predicate  $M^B(x, y)$  and a constant  $\ell \in \mathbb{N}$  such that the following holds for every language  $B$  and  $s \in \mathbb{N}$ . If  $tt \in \{0, 1\}^{2^m}$  is a string that represents the truth table of an  $m$ -variate Boolean function  $f$  which requires  $B$ -oracle circuits of size  $s^\ell$ , then, for all  $s$ -size  $B$ -oracle circuits  $C$ ,  $M^B(C, tt)$  is consistent with  $CA^B$ .*

The non-relativized version of this result was used in [28] to show that  $\text{BPP} \subseteq \text{ZPP}^{\text{MCSP}}$ . We use the relativized version to show that under certain assumptions  $\text{BPP}^A = \text{ZPP}^A$ .

► **Lemma 22.** *Let  $A, B$  be any languages such that: (1)  $A \in \text{P}^B/\text{poly}$ , and (2)  $\text{MCSP}^B \in \text{ZPP}^A$ . Then  $\text{BPP}^A = \text{ZPP}^A$ .*

**Proof.** By definition,  $\text{ZPP}^A \subseteq \text{BPP}^A$ . For the second direction, let  $L \in \text{BPP}^A$ . Then for each  $n$  there exists an  $A$ -oracle circuit  $C(w, r)$  of size  $\text{poly}(n)$  such that  $x \in L \iff C(x, \cdot) \in \text{CA}^A$ . We now describe a machine that decides  $L$ : “For  $m = \mathcal{O}(\log n)$  pick a truth table  $tt \in \{0, 1\}^{2^m}$  at random. If  $tt$  has  $B$ -circuits of size less than  $2^{m/4}$  return “?” (using an oracle to  $\text{MCSP}^B$ ). Otherwise, run  $M^A(C(x, \cdot), tt)$  and answer the same (using  $M^A$  from Lemma 21).”

By counting arguments, a random function requires exponential size circuits w.h.p. Therefore, the algorithm will output “?” extremely rarely. By Observation 11  $tt$  requires  $A$ -oracle circuits of size  $2^{\Omega(m)} = n^{\Omega(1)}$ . Consequently, the correctness of the algorithm follows from Lemma 21. As described, the algorithm can be implemented in  $\text{ZPP}^{A, \text{MCSP}^B}$ . By the preconditions,  $\text{ZPP}^{A, \text{MCSP}^B} \subseteq \text{ZPP}^{A, \text{ZPP}^A} = \text{ZPP}^A$  due to the self-lowness of  $\text{ZPP}$ . ◀

## 2.3 Natural properties, PAC learning and MCSP

We first define natural properties.

► **Definition 23** (Natural Property [41]). Let  $\mathcal{C}$  be a circuit class and  $\Gamma$  be complexity classes. We say that a property  $\mathcal{R}$  is  $\Gamma$ -natural with density  $\delta_n$  and useful against  $\mathcal{C}$  if the following holds:

1. **Constructivity:** Given a binary string  $tt \in \{0, 1\}^{2^m}$ ,  $tt \in \mathcal{R}$  can be decided in  $\Gamma$ .
2. **Largeness:** For all  $n$ ,  $\mathcal{R}$  contains at least a  $\delta_n$  fraction of all  $2^n$  binary strings, representing  $n$ -variate Boolean functions.
3. **Usefulness:** For every Boolean function family  $\{f_n\}_{n \geq 0}$ , where  $f_n$  is a function on  $n$  variables, such that  $\{tt \mid tt \text{ is a truth table of some } f_n\} \subseteq \mathcal{R}$  for almost all  $n$ , we have that  $\{f_n\} \notin \Lambda$  for almost all  $n$ .

We say that  $\mathcal{R}$  is *strongly useful* if there exists  $a \in \mathbb{N}$  such that  $\mathcal{R}$  is useful against  $\text{SIZE}[2^{an}]$  and has density  $\delta_n \geq 2^{-an}$ .

Considering  $\mathcal{R}$  as an oracle allows us to “ignore” its complexity. In addition, if  $\mathcal{R}$  is a strongly useful property, then, as observed in [16, Lemma 2.7], there exists another strongly useful property  $\mathcal{R}' \in \text{P}^{\mathcal{R}}$  with density  $\delta_n \geq 1/2$ . Therefore, when considering a strongly useful property as an oracle we can assume w.l.o.g that it has density  $\delta_n \geq 1/2$ .

Observe that  $\text{MCSP}$  yields a strongly useful natural property. Often, the only requirement from an  $\text{MCSP}$  oracle is to “serve” as a strongly useful natural property. Consequently, the oracle can be relaxed. The following can be shown along the lines of the proof of Lemma 22.

► **Lemma 24.** *Let  $\mathcal{R}$  be a strongly useful natural property. Then  $\text{BPP} \subseteq \text{ZPP}^{\mathcal{R}}$ . If, in addition,  $\mathcal{R} \in \text{P/poly}$  then  $\text{BPP}^{\mathcal{R}} = \text{ZPP}^{\mathcal{R}}$ .*

Recall Valiant’s PAC learning model [48]. We have a (computationally bounded) learner that is given a set of samples of the form  $(\bar{x}, f(\bar{x}))$  from some fixed function  $f \in \mathcal{C}$ , where  $\bar{x}$  is chosen according to some unknown distribution  $D$ . Given  $\varepsilon > 0$  and  $\delta > 0$ , the learner’s goal is to output, with probability  $1 - \varepsilon$  a hypothesis  $\hat{f}$  such that  $\hat{f}$  is a  $1 - \delta$  close to  $f$  under  $D$ . We say that a function class  $\mathcal{C}$  is *PAC learnable* if there exists a learner which given any  $f \in \mathcal{C}$ ,  $\varepsilon > 0$  and  $\delta > 0$  in time polynomial in  $n, 1/\varepsilon, 1/\delta, |f|$  outputs a hypothesis as required. In a more general model, the learner is allowed membership queries (as in the exact learning model). In this case, we say that  $\mathcal{C}$  is *PAC learnable with membership queries*.

In [16] it was shown that natural properties yield efficient learning algorithms. Specifically, a BPP-natural property that is strongly useful against a circuit class  $\mathcal{C}$  implies that  $\mathcal{C}$  is PAC learnable under the uniform distribution, with membership queries (see Section 37 for more details). Here we show that the converse holds as well: if  $\mathcal{C}$  is PAC learnable under the uniform distribution, with membership queries then there is a BPP-natural property that is strongly useful against  $\mathcal{C}$ .

► **Lemma 25.** *Let  $\mathcal{C}$  be a circuit class. If  $\mathcal{C}$  is PAC learnable under the uniform distribution, with membership queries, then there exists a BPP-natural property that is strongly useful against  $\mathcal{C}$ .*

The proof goes along the lines of Theorem 3 from [49], where it is shown how to turn an efficient randomized exact learner  $\mathcal{A}$  for a circuit class  $\mathcal{C}$  into a P/poly-natural property strongly useful against  $\mathcal{C}$ . Combined with Theorem 2, we obtain a somewhat different proof for the conditional lower bounds of [49] and [18, 32].

► **Corollary 26.** *For every circuit class  $\mathcal{C}$ , if  $\mathcal{C}$  is PAC learnable under the uniform distribution, with membership queries then:*

1.  $\text{BPP}/1 \not\subseteq \mathcal{C}\text{-SIZE}[n^k]$  and  $\text{prBPP} \not\subseteq \mathcal{C}\text{-SIZE}[n^k]$ , for all  $k \in \mathbb{N}$  [49].
2.  $\text{BPEXP} \not\subseteq \mathcal{C}\text{-SIZE}[\text{poly}]$  [18, 32].

## 2.4 Obfuscation

In this section we define the notion of an Indistinguishability Obfuscator.

► **Definition 27** (Indistinguishability Obfuscator [11]). Let  $\mathcal{A}$  be a class of algorithms. We say that a procedure  $\text{IO}$  is an  $\mathcal{A}$ -Indistinguishability Obfuscator for a circuit class  $\mathcal{C}$  if the following holds:

1. **Correctness:** For every circuit  $C \in \mathcal{C}$  and for all inputs  $x$ ,  $C(x) = \text{IO}(C)(x)$ .
2. **Polynomial slowdown:** There exists  $k \in \mathbb{N}$  s.t. for every circuit  $C \in \mathcal{C}$ ,  $|\text{IO}(C)| \leq |C|^k$ .
3. **Indistinguishability:** For all pairs of circuits  $C_1, C_2 \in \mathcal{C}$  that compute the same function, if  $|C_1| = |C_2| = s$  then the distributions of  $\text{IO}(C_1)$  and  $\text{IO}(C_2)$  are indistinguishable by any algorithm  $A \in \mathcal{A}$ . More precisely, there is a negligible function  $\text{negl}(n)$  such that for any algorithm  $A \in \mathcal{A}$ :

$$|\Pr[A(\text{IO}(C_1)) = 1] - \Pr[A(\text{IO}(C_2)) = 1]| \leq \text{negl}(s).$$

In particular, when  $\mathcal{A}$  the class of **randomized polynomial-time algorithms**, we call such  $\text{IO}$  a *Computational Obfuscator*.

## 2.5 Downward self-reducible and self-correctable languages

► **Definition 28.** We say that a language  $L$  is *downward self-reducible* if there is a deterministic polynomial-time algorithm COMPUTE such that, for all  $n \geq 1$ ,  $\text{COMPUTE}^{L|_{n-1}} = L|_n$ . In other words, COMPUTE efficiently computes  $L$  on inputs of size  $n$  given oracle access to a procedure that computes  $L$  on inputs of size  $n - 1$ .

We say that a language  $L$  is *self-correctable*<sup>3</sup> if there is a probabilistic polynomial-time algorithm CORRECT such that, for any  $n \in \mathbb{N}$  and a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  it holds that if  $\Delta(f, L|_n) \leq 1/n$  then for all  $\bar{x} \in \{0, 1\}^n$ :  $\Pr[\text{CORRECT}^f(\bar{x}) \neq L|_n(\bar{x})] \leq 1/\text{poly}(n)$ .

Several complexity classes have complete problems that are both downward self-reducible and self-correctable.<sup>4</sup>

► **Lemma 29** ([47, 12, 36, 27]). *There exists a downward self-reducible and self-correctable #P-complete language  $L_{\text{perm}}$ .*

► **Lemma 30** ([45]). *There is a downward self-reducible and self-correctable PSPACE-complete language  $L_{\text{PSPACE}}$ .*

Using similar ideas as in [45], we also show the following; see Appendix A for the proof.

► **Lemma 31.** *There is a downward self-reducible and self-correctable  $\oplus\text{P}$ -complete language  $L_{\oplus\text{P}}$ .*

To handle a larger family of languages, we generalize the notion of self-correctability.

► **Definition 32.** A language  $L$  is  $(\varepsilon(n), A)$ -correctable if there are a polynomial  $r(n)$  and a randomized polynomial-time algorithm CORRECT such that, for all  $n \in \mathbb{N}$  and  $f : \{0, 1\}^{r(n)} \rightarrow \{0, 1\}$ , if  $\Delta(f, A|_{r(n)}) \leq \varepsilon(n)$ , then, for all  $\bar{x} \in \{0, 1\}^n$ ,  $\Pr[\text{CORRECT}^f(\bar{x}) \neq L|_n(\bar{x})] \leq 1/\text{poly}(n)$ .

In other words, there is a randomized polynomial-time algorithm that can decide  $L|_n$  given an oracle to a function that approximates a  $A|_{r(n)}$ . Self-correctability is special case when  $\varepsilon = 1/n$ ,  $A = L$  and  $r(n) = n$ . The following is immediate using Adleman's result [1]:

► **Lemma 33.** *Let  $L$  be a  $(\varepsilon(n), A)$ -correctable language with  $r(n)$ , and let  $B$  be a language. Suppose  $C$  is an  $r(n)$ -variate  $B$ -oracle circuit of size  $s$  such that  $\Delta(C, A|_{r(n)}) \leq \varepsilon(n)$ , for some  $n \in \mathbb{N}$ . There exists a randomized polynomial-time algorithm that given  $C$  as input, outputs an  $n$ -variate  $B$ -oracle circuit  $C'$  of size  $\text{poly}(r(n), s)$  such that  $C' \equiv L|_n$ , w.h.p.*

In particular, this result implies that such  $C'$  exists for every  $C$ .

Klivans and van Melkebeek [33] show that any language  $L$  is  $(\varepsilon(n), A)$ -correctable for  $A$  computable in PSPACE with an oracle to  $L$  (by encoding the truth table of  $L$  with an appropriate error-correcting code).

► **Theorem 34.** *For any language  $L$  and  $\varepsilon(n)$  there exist a language  $A \in \text{DSPACE}^L(n+1/\varepsilon(n))$  such that  $L$  is  $(\varepsilon(n), A)$ -correctable with  $r(n) = \text{poly}(n, 1/\varepsilon(n))$ .*

<sup>3</sup> More generally, such languages are referred to as “random self-reducible” languages.

<sup>4</sup> It is not hard to see that every downward self-reducible language is computable in PSPACE. On the other hand, the results of [17] suggest that there cannot be self-correctable languages which are complete for any level of the polynomial hierarchy, unless the hierarchy collapses.



## 2.6 Learning downward self-reducible and self-correctable languages

The following lemma essentially shows that the PAC learnability for downward self-reducible and self-correctable languages implies exact learnability; a similar lemma also appeared in [40]. See the Appendix (Section B) for the proof.

► **Lemma 35.** *Let  $B$  be a language. Suppose Boolean circuits are PAC learnable using membership and  $B$  queries with hypotheses being  $B$ -oracle circuits. Suppose  $L$  is a downward self-reducible and self-correctable language. Then there is a randomized algorithm making oracle queries to  $B$ , that, given  $x$  and  $t$ , computes  $L(x)$  with probability at least  $1 - 1/\text{poly}(|x|)$  in time  $\text{poly}(|x|, t)$ , provided that  $t \geq s_L(|x|)$ .*

### 3 The proofs

Our proofs will use the following.

► **Lemma 36** (Extension of Theorem 5.1 from [16]). *Let  $\mathcal{R}$  be a natural property with density at least  $1/5$ , that is useful against  $\text{SIZE}[u(n)]$ , for some size function  $u(n) : \mathbb{N} \rightarrow \mathbb{N}$ . Then there is a randomized algorithm that makes oracle queries to  $\mathcal{R}$  such that, given  $s \in \mathbb{N}$ , oracle access to a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  computable by a Boolean circuit of size  $s$ , and  $\delta > 0$ , it produces in time  $\text{poly}(n, 1/\delta, 2^{u^{-1}(\text{poly}(n, 1/\delta, s))})$  an  $\mathcal{R}$ -oracle circuit  $C$  where  $\Delta(C, f) \leq \delta$ .*

► **Corollary 37.** *Let  $\mathcal{R}$  be a strongly useful natural property. Then Boolean circuits are PAC learnable under the uniform distribution, using membership and  $\mathcal{R}$  queries with hypotheses being  $\mathcal{R}$ -oracle circuits.*

► **Theorem 38.** *Boolean circuits are PAC learnable under the uniform distribution, using membership and MCSP queries with hypotheses being MCSP-oracle circuits.*

Combining Lemma 35 and Corollary 37, we get the following.

► **Lemma 39.** *Let  $\mathcal{R}$  be a strongly useful natural property and let  $L$  be a downward self-reducible and self-correctable language. Then there is a randomized algorithm that makes oracle queries to  $\mathcal{R}$ , that given  $x$  and  $t$  computes  $L(x)$  with probability at least  $1 - 1/\text{poly}(|x|)$  in time  $\text{poly}(|x|, t)$  provided that  $t \geq s_L(|x|)$ .*

### 3.1 Conditional collapses

Theorem 1 follows as a corollary from the next, somewhat stronger, theorem.

► **Theorem 40.** *Let  $\mathcal{R}$  be a strongly useful natural property, and furthermore let  $\Gamma \in \{\oplus\text{P}, \text{P}^{\#\text{P}}, \text{PSPACE}, \text{EXP}, \text{NEXP}, \text{EXP}^{\text{NP}}\}$ . Then, if  $\Gamma \subseteq \text{P/poly}$ , then  $\Gamma \subseteq \text{BPP}^{\mathcal{R}}$ . If, in addition,  $\mathcal{R} \in \text{PH}$  then  $\Gamma \subseteq \text{ZPP}^{\mathcal{R}}$ .*

**Proof.** First, consider the case of  $\Gamma$  such that  $\text{PSPACE} \subseteq \Gamma$ . For  $L_{\text{PSPACE}}$  from Lemma 30, we have that  $s_{L_{\text{PSPACE}}}(n) = \mathcal{O}(n^k)$  for some  $k \in \mathbb{N}$ . By Lemma 39, given  $x$ , we can compute  $L_{\text{PSPACE}}(x)$  in randomized polynomial time given oracle to  $\mathcal{R}$ . Consequently,  $\text{PSPACE} \subseteq \text{BPP}^{\mathcal{R}}$ . By Lemma 18, we get  $\Gamma = \text{MA}$ . Hence, we have  $\Gamma \subseteq \text{MA} \subseteq \text{PSPACE} \subseteq \text{BPP}^{\mathcal{R}}$ . If, in addition,  $\mathcal{R} \in \text{PH}$  then  $\mathcal{R} \in \Gamma \subseteq \text{P/poly}$ . By Lemma 24,  $\text{BPP}^{\mathcal{R}} \subseteq \text{ZPP}^{\mathcal{R}}$ .

For  $\Gamma = \oplus\text{P}$ , we argue as before, using Lemma 31 instead of Lemma 30, to obtain that  $\oplus\text{P} \subseteq \text{BPP}^{\mathcal{R}}$ . If, in addition,  $\mathcal{R} \in \text{PH}$ , then, by Toda's Theorem [43],  $\text{PH} \subseteq \text{BPP}^{\oplus\text{P}}$ , and hence  $\mathcal{R} \in \text{BPP}^{\oplus\text{P}} \subseteq \text{P/poly}$ . The rest of the argument follows as above.

For  $\Gamma = \mathcal{P}^{\#P}$ , we argue as before using Lemma 29 instead of 30, with the additional observation that in this case the permanent has small Boolean circuits. The only difference is that in this case the function has multiple inputs. For more details we refer the reader to [27]. ◀

### 3.2 Circuit lower bounds for MCSP-oracle classes

Theorems 2 and 3 follow from the next theorem.

► **Theorem 41.** *For any strongly useful natural property  $\mathcal{R}$  we have*

1.  $\text{ZPP}^{\mathcal{R}}/1 \not\subseteq \text{SIZE}[n^k]$  and  $\text{prZPP}^{\mathcal{R}} \not\subseteq \text{SIZE}[n^k]$  for all  $k \in \mathbb{N}$ , and
2.  $\text{ZPEXP}^{\mathcal{R}} \not\subseteq \text{P/poly}$ .

**Proof.** Assume w.l.o.g that  $\mathcal{R} \in \text{P/poly}$  (otherwise there is nothing to prove). Consider  $L = L_{\text{PSPACE}}$  from Lemma 30. As  $L \in \text{PSPACE} \subseteq \text{EXP}$ , by a translation argument there exists  $d \geq 1$  such that  $L \in \text{SIZE}(2^{n^d})$ . Therefore,  $s_L(n)$  is well-defined and in particular  $s_L(n) = \mathcal{O}(2^{n^d})$ .

We first prove part (1) of the theorem. We focus on the class  $\text{ZPP}^{\mathcal{R}}/1$ ; the claim about  $\text{prZPP}$  will follow by Lemma 17. We consider two cases:

**Case 1:**  $\text{PSPACE} \subseteq \text{P/poly}$ . By Theorem 1 and Lemma 24,  $\text{PSPACE} \subseteq \text{BPP}^{\mathcal{R}} \subseteq \text{ZPP}^{\mathcal{R}}$ .

Hence, by Lemma 19, for all  $k \in \mathbb{N} : \text{ZPP}^{\mathcal{R}} \not\subseteq \text{SIZE}[n^k]$ .

**Case 2:**  $\text{PSPACE} \not\subseteq \text{P/poly}$ . As  $L$  is  $\text{PSPACE}$ -complete, we have that  $L \notin \text{P/poly}$ . Assume towards contradiction that  $\text{BPP}^{\mathcal{R}}/1 \subseteq \text{SIZE}[n^k]$ , for some  $k \in \mathbb{N}$ . By Lemma 16,  $L'_k \in \text{SIZE}[n^k]$ . And thus, by Lemma 15,  $s_L(n) = \mathcal{O}(n^{2k})$ . This contradicts the assumption that  $L \notin \text{P/poly}$ . As in Lemma 24, we obtain that, for all  $k \in \mathbb{N}$ ,  $\text{ZPP}^{\mathcal{R}}/1 \not\subseteq \text{SIZE}[n^k]$ .

Part (2) of the theorem is also shown by considering two cases:

**Case 1:**  $\text{PSPACE} \subseteq \text{P/poly}$ . As above,  $\text{PSPACE} \subseteq \text{ZPP}^{\mathcal{R}}$ . By a translation argument,  $\text{EXPSPACE} \subseteq \text{ZPEXP}^{\mathcal{R}}$ . By Lemma 19,  $\text{ZPEXP}^{\mathcal{R}} \not\subseteq \text{P/poly}$ .

**Case 2:**  $\text{PSPACE} \not\subseteq \text{P/poly}$ . Since  $\text{PSPACE} \subseteq \text{EXP} \subseteq \text{ZPEXP}^{\mathcal{R}}$ , the theorem follows. ◀

### 3.3 IO related results

We prove more general statements for strongly useful natural properties.

► **Theorem 42.** *Let  $\mathcal{R}$  be a strongly useful natural property and let  $\mathcal{A}$  denote the class of randomized polynomial-time algorithms with  $\mathcal{R}$  oracle. If there exists an  $\mathcal{A}$ -indistinguishable obfuscator  $\text{IO}$  then  $\text{NP} \subseteq \text{ZPP}^{\mathcal{R}}$ .*

Before proving the Theorem we require the following result of [2] that allows to find preimages of polynomial-time computable functions<sup>5</sup>.

► **Lemma 43** ([2]). *Let  $\mathcal{R}$  be a strongly useful natural property. Let  $f_y(x) = f(y, x)$  be a function computable uniformly in time polynomial in  $|x|$ . There exists a polynomial-time probabilistic oracle Turing machine  $M$  and  $k \in \mathbb{N}$  such that for any  $n$  and  $y$ :*

$$\Pr_{|x|=n, t} [f_y(M^{\mathcal{R}}(y, f_y(x), t)) = f_y(x)] \geq 1/n^k$$

where  $x$  is chosen uniformly at random and  $t$  denotes the randomness of  $M$ .

<sup>5</sup> The original result is formulated in a slightly different terminology.



We now present the proof of Theorem 42.

**Proof.** Consider the function  $f_C(r) = \text{IO}(C, r)$ , where  $C$  is a circuit and  $r$  is a random string. Observe that  $f_C(r)$  is computable uniformly in time polynomial in  $|r|$ . By the Lemma above there exists a polynomial-time probabilistic oracle Turing machine  $M$  and  $k \in \mathbb{N}$  such that for any circuit  $C$ :

$$\Pr_{r,t} [f_C(M^{\mathcal{R}}(C, \text{IO}(C, r), t)) = f_C(r)] \geq 1/|r|^k$$

where  $r$  is chosen uniformly at random and  $t$  denotes the randomness of  $M$ .

We now describe a polynomial-time randomized Turing machine that decides SAT. For  $s \in \mathbb{N}$ , we denote by  $\perp_s$  a canonical unsatisfiable circuit. Note that given  $s$ ,  $\perp_s$  can be computed uniformly in time polynomial in  $s$ . Given a circuit  $C$  as input:

1. Let  $s = |C|$ .
2.  $\hat{C} = \text{IO}(C, r)$  for  $r$  chosen uniformly at random.
3. Run  $M^{\mathcal{R}}(\perp_s, \hat{C}, t)$  to obtain  $r'$ .
4. Accept if and only if  $\text{IO}(\perp_s, r') = \hat{C}$ .

We observe the following:

- If  $C = \perp_s$  then the algorithm will accept with probability  $\geq 1/|r|^k$ .
  - If  $C \in \text{SAT}$  then by the correctness requirement of IO (Requirement 2) for all  $r, r'$ :  $\text{IO}(\perp_s, r') \neq \text{IO}(C, r)$ . Therefore, the algorithm will always reject.
  - Finally, if  $C \in \overline{\text{SAT}}$ , then by the indistinguishability requirement (Requirement 3), the oracle machine  $M^{\mathcal{R}}$  could not distinguish between the obfuscation of  $\perp_s$  and the obfuscation of  $C$ . Hence, the algorithm will accept with probability  $1/|r|^k - \text{negl}(|r|)$ .
- By repeating the algorithm, we obtain that  $\overline{\text{SAT}} \in \text{RP}^{\mathcal{R}}$

By Lemma 20,  $\text{SAT} \in \text{RP}^{\mathcal{R}}$  and hence  $\text{SAT} \in \text{ZPP}^{\mathcal{R}}$ . ◀

### 3.4 Hardness of relativized versions of MCSP

First we observe that, for every oracle  $B$ , there is a  $\text{P}^{\text{MCSP}^B}$ -natural property for  $B$ -oracle circuits. Combined with Lemma 36, this yields the following theorem along the lines of Theorem 38.

► **Theorem 44.** *For every oracle  $B$  the class of  $B$ -oracle circuits is PAC learnable under the uniform distribution, using membership and  $\text{MCSP}^B$  queries with hypotheses being  $\text{MCSP}^B$ -oracle circuits.*

► **Lemma 45.** *Let  $A, B$  be two oracles (languages) such that  $A \in \text{P}^B/\text{poly}$ . Then:*

1. *For every  $n \in \mathbb{N}$  and  $\delta > 0$ , there exists a  $\text{MCSP}^B$ -oracle circuit  $C$  of size  $\text{poly}(n, 1/\delta)$  such that  $\Delta(C, A|_n) \leq \delta$ .*
2. *If, in addition,  $A$  is self-correctable then  $A \in \overline{\text{P}^{\text{MCSP}^B}}/\text{poly}$ .*
3. *If, in addition to the above,  $A$  is downward self-reducible, then  $A \in \text{BPP}^{\text{MCSP}^B}$ .*

**Proof.**

1. By the assumption, for every  $n \in \mathbb{N}$  the function  $A|_n(x)$  has a  $B$ -oracle circuit of size  $\text{poly}(n)$ . Therefore, by Theorem 44, given oracle access to  $A|_n$ , the learning algorithm produces an  $\text{MCSP}^B$ -oracle circuit  $C$  of size  $\text{poly}(n, 1/\delta)$  such that  $\Delta(C, A|_n) \leq \delta$ .
2. Follows from Lemma 33.
3. Follows by combining Theorem 44 with Lemma 35. ◀

## 7:14 The Power of Natural Properties as Oracles

► **Remark.** In the lemma above, although the learning algorithm actually needs oracle access to  $A|_n$  to produce  $C$ , in parts 1 and 2 we are only interested in mere existence. In part 3, on the other hand, we actually benefit from the learning algorithm.

We are now ready to give the proofs of Theorems 7–10. For convenience, we re-state them below.

► **Theorem 46** (Theorem 7 re-stated).

1.  $\text{PSPACE} \subseteq \text{ZPP}^{\text{MCSP}^{\text{PSPACE}}}$  [2]
2.  $\oplus\text{P} \subseteq \text{ZPP}^{\text{MCSP}^{\oplus\text{P}}}$
3.  $\text{P}^{\#\text{P}} \subseteq \text{BPP}^{\text{MCSP}^{\#\text{P}}}$
4.  $\text{PP} \subseteq \text{BPP}^{\text{MCSP}^{\text{PP}}}$ . Moreover, for  $k \geq 2$ :  $C_k\text{P} \subseteq C_{k-1}\text{P}^{\text{MCSP}^{\text{PP}}}$ .

**Proof.**

1. Consider any PSPACE-complete language  $B$ . Let  $L_{\text{PSPACE}}$  be the language from Lemma 30. By Lemma 45 (3),  $L_{\text{PSPACE}} \in \text{BPP}^{\text{MCSP}^B}$ , and hence  $\text{PSPACE} \subseteq \text{BPP}^{\text{MCSP}^B}$ . Observe that

$$\text{MCSP}^B \in \text{NP}^B \subseteq \text{PSPACE}^B = \text{PSPACE},$$

and so  $\text{MCSP}^B \in \text{P}^B/\text{poly}$ . By Lemma 22, we conclude that  $\text{BPP}^{\text{MCSP}^B} = \text{ZPP}^{\text{MCSP}^B}$ .

2. Arguing as above, using Lemma 31 instead of Lemma 30, we obtain  $\oplus\text{P} \subseteq \text{BPP}^{\text{MCSP}^{\oplus\text{P}}}$ . By Toda's Theorem [43],  $\text{NP}^{\oplus\text{P}} \subseteq \text{BPP}^{\oplus\text{P}}$ . Therefore, we get

$$\text{MCSP}^{\oplus\text{P}} \in \text{NP}^{\oplus\text{P}} \subseteq \text{BPP}^{\oplus\text{P}} \subseteq \text{P}^{\oplus\text{P}}/\text{poly}.$$

The rest of the argument is the same as above.

3. Similar to the first proof, using Lemma 29 instead of Lemma 30.
4. Since  $\text{P}^{\#\text{P}} = \text{P}^{\text{PP}}$  [43], we get that  $\text{PP} \subseteq \text{P}^{\#\text{P}} \subseteq \text{BPP}^{\text{MCSP}^{\text{PP}}}$ .
5.  $\text{PP} \subseteq \text{BPP}^{\text{MCSP}^{\text{PP}}}$ . The second part of the claim follows by induction since  $C_k\text{P} = \text{PP}^{C_{k-1}\text{P}}$ . ◀

► **Theorem 47** (Theorem 8 re-stated).  $\bigcap_B \text{BPP}^{\text{MCSP}^B}/1 \not\subseteq \text{SIZE}[n^k]$  and  $\bigcap_B \text{prBPP}^{\text{MCSP}^B} \not\subseteq \text{SIZE}(n^k)$  for all  $k \in \mathbb{N}$ , and  $\bigcap_B \text{BPEXP}^{\text{MCSP}^B} \not\subseteq \text{P}/\text{poly}$ .

**Proof.** As in the proof of Theorem 41, we consider two cases:

**Case 1:**  $\text{PSPACE} \subseteq \text{P}/\text{poly}$ . Let  $L_{\text{PSPACE}}$  be the language from Lemma 30. Observe that for every language  $B$ , we have  $L_{\text{PSPACE}} \in \text{P}^B/\text{poly}$ . By Lemma 45 (3),  $\text{PSPACE} \subseteq \text{BPP}^{\text{MCSP}^B}$  for all  $B$ . By Lemma 19, the required circuit lower bound follows.

**Case 2:**  $\text{PSPACE} \not\subseteq \text{P}/\text{poly}$ . For each  $k \in \mathbb{N}$  there exists  $L'_k \notin \text{SIZE}(n^k)$  such that  $L'_k \in \text{BPP}^{\text{MCSP}^B}/1$  for all  $B$ . ◀

► **Theorem 48** (Theorem 9 re-stated). For any language  $B$ ,  $n \in \mathbb{N}$  and  $\delta > 0$ , there exists a  $\text{MCSP}^B$ -oracle circuit  $C$  of size  $\text{poly}(n, 1/\delta)$  that is  $1 - \delta$  close to  $B|_n$ . If, in addition,  $B$  is self-correctable then  $B$  has polynomial size  $\text{MCSP}^B$ -oracle circuits.

**Proof.** The proof follows from Lemma 45 for  $A = B$ , since  $B \in \text{P}^B/\text{poly}$ . ◀

► **Theorem 49** (Theorem 10 re-stated). Let  $B$  be a language such that  $\text{PSPACE}^B$  has polynomial size  $B$ -oracle circuits. Then  $B$  has polynomial-size  $\text{MCSP}^B$ -oracle circuits.

**Proof.** Let  $A \in \text{PSPACE}^B$  be a language such that  $B$  is  $(1/\text{poly}(n), A)$ -correctable with  $r(n) = \text{poly}(n)$ , as guaranteed by Theorem 34. By assumption,  $A \in \text{PSPACE}^B \subseteq \text{P}^B/\text{poly}$ . By Lemma 45 (1), for every  $n \in \mathbb{N}$ , there exists an  $\text{MCSP}^B$ -oracle circuit  $C_n$  of size  $\text{poly}(n)$  such that  $\Delta(C_n, A|_{r(n)}) \leq 1/\text{poly}(n)$ . Lemma 33 completes the proof.  $\blacktriangleleft$

## 4 Open questions

The main open question is, of course, to determine the complexity of MCSP. The results in this paper may be interpreted as giving some hope that hardness of MCSP is possible to prove under randomized Turing reductions, as we see a growing list of non-trivial computational tasks that can be solved with the help of the MCSP oracle rather than the SAT oracle. It would be interesting to see more examples of complexity results proved with the SAT oracle that remain true when SAT is replaced with MCSP. For example, is it true that if  $\text{SAT} \in \text{P}/\text{poly}$ , then SAT circuits can be found by a  $\text{ZPP}^{\text{MCSP}}$  algorithm (strengthening the  $\text{ZPP}^{\text{NP}}$  result by [13, 34])? Probably a simpler question along these lines is: Does  $\text{SAT} \in \text{P}/\text{poly}$  imply that  $\text{NP} \subseteq \text{ZPP}^{\text{MCSP}}$ ?

Some of our hardness results for the relativized MCSP (Theorem 7) are for ZPP reductions, while others for BPP reductions. Is it possible to replace the BPP reductions with ZPP reductions? We have shown it for PSPACE and  $\oplus\text{P}$ , but not for  $\#\text{P}$ .

Finally, we proved that, under some assumptions, every language  $L$  is computable by a polynomial-size circuit with  $\text{MCSP}^L$  oracle gates (Theorem 10). Is it true without any assumptions?

---

## References

- 1 L. M. Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 75–83, 1978.
- 2 Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006. doi:10.1137/050628994.
- 3 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. 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 25–32. Springer, 2014. doi:10.1007/978-3-662-44465-8\_3.
- 4 Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing disjunctive normal form formulas and  $\text{ac}^0$  circuits given a truth table. *SIAM J. Comput.*, 38(1):63–84, 2008. doi:10.1137/060664537.
- 5 Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 21–33. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.STACS.2015.21.
- 6 S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- 7 Vikraman Arvind and Johannes Köbler. On pseudorandomness and resource-bounded measure. *Theor. Comput. Sci.*, 255(1-2):205–221, 2001. doi:10.1016/S0304-3975(99)00164-4.
- 8 L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

- 9 L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.
- 10 B. Barak. A probabilistic-time hierarchy theorem for “slightly non-uniform” algorithms. In *RANDOM*, pages 194–208, 2002.
- 11 B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference*, pages 1–18, 2001.
- 12 Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In Christian Choffrut and Thomas Lengauer, editors, *STACS 90, 7th Annual Symposium on Theoretical Aspects of Computer Science, Rouen, France, February 22-24, 1990, Proceedings*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 1990. doi:10.1007/3-540-52282-4\_30.
- 13 Nader H. Bshouty, Richard Cleve, Ricard Gavaldà, Sampath Kannan, and Christino Tamon. Oracles and queries that are sufficient for exact learning. *J. Comput. Syst. Sci.*, 52(3):421–433, 1996. doi:10.1006/jcss.1996.0032.
- 14 H. Buhrman, L. Fortnow, and T. Thierauf. Nonrelativizing separations. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity (CCC)*, pages 8–12, 1998.
- 15 Harry Buhrman and Steven Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In R. K. Shyamasundar, editor, *Foundations of Software Technology and Theoretical Computer Science, 12th Conference, New Delhi, India, December 18-20, 1992, Proceedings*, volume 652 of *Lecture Notes in Computer Science*, pages 116–127. Springer, 1992. doi:10.1007/3-540-56287-7\_99.
- 16 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.10.
- 17 J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM J. on Computing*, 22(5):994–1005, 1993.
- 18 L. Fortnow and A. R. Klivans. Efficient learning algorithms yield circuit lower bounds. *J. Comput. Syst. Sci.*, 75(1):27–36, 2009.
- 19 L. Fortnow and R. Santhanam. Hierarchy theorems for probabilistic polynomial time. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 316–324, 2004.
- 20 O. Goldreich and D. Zuckerman. Another proof that  $BPP \subseteq PH$  (and more). *Studies in Complexity and Cryptography*, pages 40–53, 2011.
- 21 S. Goldwasser and G. N. Rothblum. On best-possible obfuscation. In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC*, pages 194–213, 2007.
- 22 Hans Heller. On relativized exponential and probabilistic complexity classes. *Information and Control*, 71(3):231–243, 1986. doi:10.1016/S0019-9958(86)80012-2.
- 23 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.18.
- 24 John M. Hitchcock and Aduri Pavan. On the np-completeness of the minimum circuit size problem. In Prahladh Harsha and G. Ramalingam, editors, *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*, volume 45 of *LIPICs*, pages 236–245. Schloss

- Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.FSTTCS.2015.236.
- 25 R. Impagliazzo, V. Kabanets, and A. Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *J. of Computer and System Sciences*, 65(4):672–694, 2002.
  - 26 R. Impagliazzo and A. Wigderson. P=BPP unless E has subexponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 220–229, 1997.
  - 27 R. Impagliazzo and A. Wigderson. Randomness vs. time: De-randomization under a uniform assumption. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 734–743, 1998.
  - 28 V. Kabanets and J.-Y. Cai. Circuit minimization problem. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 73–79, 2000.
  - 29 Ravi Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1-3):40–56, 1982. doi:10.1016/S0019-9958(82)90382-5.
  - 30 Richard M. Karp. Turing award lecture. In Bill Healy and Judith D. Schlesinger, editors, *Proceedings of the 1985 ACM annual conference on The range of computing: mid-80's perspective: mid-80's perspective, Denver, Colorado, USA, October 14-16, 1985*, page 193. ACM, 1985. doi:10.1145/320435.320497.
  - 31 Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. 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 302–309. ACM, 1980. doi:10.1145/800141.804678.
  - 32 A. Klivans, P. Kothari, and I. Oliveira. Constructing hard functions from learning algorithms. In *Proceedings of the 28th Annual IEEE Conference on Computational Complexity (CCC)*, pages 86–97, 2013.
  - 33 Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002. doi:10.1137/S0097539700389652.
  - 34 Johannes Köbler and Osamu Watanabe. New collapse consequences of NP having small circuits. *SIAM J. Comput.*, 28(1):311–324, 1998. doi:10.1137/S0097539795296206.
  - 35 Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 374–383. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.47.
  - 36 C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *JACM*, 39(4):859–868, 1992.
  - 37 D. van Melkebeek and K. Pervyshev. A generic time hierarchy with one bit of advice. *Computational Complexity*, 16(2):139–179, 2007.
  - 38 Cody D. Murray and Richard Ryan Williams. On the (non) np-hardness of computing circuit complexity. In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPIcs*, pages 365–380. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.CCC.2015.365.
  - 39 N. Nisan and A. Wigderson. Hardness vs. randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.
  - 40 I. C. Oliveira and R. Santhanam. Conspiracies between learning algorithms, circuit lower bounds and pseudorandomness. *CoRR*, abs/1611.01190, 2016. URL: <http://arxiv.org/abs/1611.01190>.

- 41 A. A. Razborov and S. Rudich. Natural proofs. *J. of Computer and System Sciences*, 55(1):24–35, 1997.
- 42 R. Santhanam. Circuit lower bounds for Merlin–Arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009.
- 43 S. Toda. PP is as hard as the polynomial time hierarchy. *SIAM J. on Computing*, 20(5):865–877, 1991.
- 44 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.
- 45 L. Trevisan and S. P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.
- 46 C. Umans. Pseudo-random generators for all hardnesses. *J. of Computer and System Sciences*, 67(2):419–440, 2003.
- 47 L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
- 48 L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- 49 I. Volkovich. On learning, lower bounds and (un)keeping promises. In *Proceedings of the 41st ICALP*, pages 1027–1038, 2014.

## A Self-correctable and downward-reducible $\oplus\text{P}$ -complete problem

In this section we prove Lemma 31

► **Lemma 50.** *There is a downward self-reducible and self-correctable  $\oplus\text{P}$ -complete language  $L_{\oplus\text{P}}$ .*

**Proof sketch.** The proof is very similar to the one in [45] for the case of PSPACE. We define a formula  $\Phi_n(\bar{x}, \bar{y})$  that is universal for  $n$ -variate 3-cnf formulas on the variables  $\bar{x} = (x_1, \dots, x_n)$ , where  $\bar{y} = (y_1, \dots, y_{8n^3})$  describes a particular 3-cnf formula  $\phi$  by specifying, for each possible clause on 3 variables, whether this clause is present in  $\phi$ . For example, if  $c_1, \dots, c_m$ , for  $m = 8n^3$ , is a sequence of all possible 3-clauses on  $n$  variables  $x_1, \dots, x_n$ , we can define  $\Phi$  as follows:

$$\Phi(\bar{x}, \bar{y}) = \bigwedge_{i=1}^m ((y_i \wedge c_i) \vee \neg y_i).$$

We now “arithmetize” the formula  $\Phi$ , getting a polynomial that agrees with  $\Phi$  over all Boolean inputs. We will work over the finite field  $\mathbb{F}_{2^k}$  of characteristic 2, for  $k = 5 \log n$ . Arithmetizing all clauses  $c_i$ ’s (by replacing each  $c_i$  with a degree 3 multilinear polynomial  $c'_i$ , in the same 3 variables, that agrees with  $c_i$  over Boolean all assignments), we get the following arithmetization  $\Phi'$  of  $\Phi$ :

$$\Phi'(\bar{x}, \bar{y}) = \prod_{i=1}^m (y_i \cdot c'_i + 1 + y_i).$$

For each  $0 \leq i \leq n$ , define a polynomial

$$f_{n,i}(x_1, \dots, x_i, \bar{y}) = \sum_{x_{i+1} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} \Phi'(\bar{x}, \bar{y}),$$

where the summation is over our field  $\mathbb{F}_{2^k}$  of characteristic 2. Note that  $f_{n,0}(\bar{y})$ , for a Boolean  $\bar{y}$ , is exactly  $\oplus\text{SAT}$  on the 3-cnf instance described by  $\bar{y}$ . So  $f_{n,0}$  is  $\oplus\text{P}$ -hard to compute.



We have that  $f_{n,i}$  can be expressed in terms of  $f_{n,i+1}$ , for  $i < n$ , by the formula:

$$f_{n,i}(x_1, \dots, x_i, \bar{y}) = f_{n,i+1}(x_1, \dots, x_i, 0) + f_{n,i+1}(x_1, \dots, x_i, 1).$$

So  $f_{n,i}$  can be computed in polynomial time with oracle access to  $f_{n,i+1}$ . It is also clear that  $f_{n,n}$  can be evaluated in polynomial time (directly).

Next, in the same way as in [45], we define a Boolean function family  $F = \{F_t\}_{t \geq 1}$  so that each  $f_{n,i}$  is “embedded” into some  $F_{h(n,i)}$ , for some function  $h: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ . Namely,  $h$  can be chosen so that

- $h(n,i) > h(n,i+1)$  (and so we have downward-reducibility for  $f_{n,i}$ ’s), and
- the length  $h(n,i)$  is large enough to accommodate both an input to  $f_{n,i}$  and an index  $j \in [k]$ .

We then define

$$F_{h(n,i)}(x_1, \dots, x_i, \bar{y}, j) = f_{n,i}(x_1, \dots, x_i, \bar{y})_j,$$

i.e., the  $j$ th bit of the value of  $f_{n,i}$  in the field  $\mathbb{F}_{2^k}$ , whose elements are viewed as  $k$ -bit vectors.

The downward-reducibility of  $F$  follows from the properties of  $f_{n,i}$  (and the way we arranged the lengths  $h(n,i)$ ). The self-correctability of  $F$  follows from the fact each  $f_{n,i}$  is a  $O(n^3)$ -degree polynomial over the field of size  $2^k \geq n^5$  (see [45] for more details). The  $\oplus P$ -hardness of  $F$  follows from  $\oplus P$ -hardness of  $f_{n,0}$ ’s.

It remains to show that  $F \in \oplus P$ . Note that every bit  $j$  of the value of  $f_{n,n}(\bar{y})$ , for every input  $\bar{y}$ , is computable in  $P$ , and hence also in  $\oplus P$ . For any  $0 \leq i < n$ , the  $j$ th bit of  $f_{n,i}(x_1, \dots, x_i, \bar{y})$  can be computed in  $\oplus P$  using the following nondeterministic algorithm:

“Nondeterministically guess Boolean values  $b_{i+1}, \dots, b_n$ . Compute the value

$$v = \Phi'(x_1, \dots, x_i, b_{i+1}, \dots, b_n, \bar{y}).$$

Accept if the  $j$ th bit of the computed field element  $v$  is 1, and reject otherwise.”

The parity of the number of accepting paths of the algorithm above is exactly the sum modulo 2 of the bits  $v_j$ , over all Boolean assignments to  $x_{i+1}, \dots, x_n$ . The latter is exactly the  $j$ th bit of  $f_{n,i}$  because addition in the field  $\mathbb{F}_{2^k}$  is the bit-wise XOR of the corresponding  $k$ -bit vectors. ◀

## B Proof of Lemma 35

Let  $\mathcal{A}$  be a PAC learner for  $\mathcal{C}$  and let  $n = |x|$ . First, we describe an algorithm that produces  $B$ -oracle circuits for  $L|_1, L|_2, \dots, L|_n$  w.h.p. We then use the circuit for  $L|_n$  to decide  $x$ .

- Begin with a look-up table  $\tilde{C}_1 = C_1$  for  $L|_1$ .
- For  $i \geq 2$ , invoke  $\mathcal{A}$  with  $\varepsilon = 1/i^3$  and  $\delta = 1/i$  to learn a circuit  $\tilde{C}_i$  of size  $t$  for  $L|_i$ .
- Answer the queries to  $B$  using the provided oracle.
- Given a query to  $L|_i$ , invoke COMPUTE with  $C_{i-1}(x)$  as an oracle.
- Set  $C_i \triangleq \text{CORRECT}^{\tilde{C}_i}$  (convert the algorithm into a circuit using Lemma 33).

We claim that w.h.p it holds for all  $1 \leq i \leq n$  that  $C_i$  is a  $B$ -oracle circuit of size  $\text{poly}(i, t)$  computing  $L|_i$ . The proof is by induction on  $i$ . Basis  $i = 1$  is clear. Now assume that hypothesis holds for  $i - 1$ . Observe that since  $C_{i-1}(x)$  is  $B$ -oracle circuit, it can be evaluated in polynomial time given and an oracle to  $B$ . Hence, by downward self-reducibility of  $L$

## 7:20 The Power of Natural Properties as Oracles

invoking COMPUTE with  $C_{i-1}(x)$  can be used to obtain oracle access to  $L|_i$ . As  $t \geq s_L(i)$ ,  $\mathcal{A}$  will output a circuit  $\tilde{C}_i$  of size  $\text{poly}(i, t)$ , which is  $1/i$  close to  $L|_i$ . Finally, using Lemma 33 the algorithm will produce a circuit  $C_i$  of size  $\text{poly}(i, t)$  that computes  $L|_i$ .

The above analysis is correct assuming that no errors have occurred. Note that the total number of steps is  $\text{poly}(i)$  while each step has at most  $1/\text{poly}(i)$  probability error. As the latter polynomial can be made arbitrary small, we obtain that w.h.p. for all  $i$ ,  $C_i \equiv L|_i$ .

Finally, all the listed procedures are in time  $\text{poly}(n, t)$ , given oracle access to  $B$ .

### **C** Oracles $B$ where $\text{PSPACE}^B \subseteq \text{P}^B/\text{poly}$ but $\text{PSPACE}^B \neq \text{P}^B$

► **Lemma 51.** *Let  $B$  be a language such that  $\text{EXP}^B \subseteq \text{P}^B/\text{poly}$ . Then  $\text{PSPACE}^B \neq \text{P}^B$ .*

**Proof.** Assume the contrary. By Meyer's Theorem [31]:  $\text{EXP}^B \subseteq \Sigma_2^B \subseteq \text{PSPACE}^B$ . By the assumption,  $\text{EXP}^B \subseteq \text{PSPACE}^B \subseteq \text{P}^B$  which contradicts Time Hierarchy Theorem. ◀

There are numerous examples of languages satisfying the preconditions of the Lemma; see, e.g., [22, 14].



# Linear Sketching over $\mathbb{F}_2$

**Sampath Kannan**<sup>1</sup>

University of Pennsylvania  
kannan@cis.upenn.edu

**Elchanan Mossel**<sup>2</sup>

Massachusetts Institute of Technology  
elmos@mit.edu

**Swagato Sanyal**<sup>3</sup>

Division of Mathematical Sciences, Nanyang Technological University, Singapore and Centre for Quantum Technologies, National University of Singapore, Singapore  
sanyalswagato@gmail.com

**Grigory Yaroslavl'tsev**<sup>4</sup>

Indiana University, Bloomington  
grigory@grigory.us

---

## Abstract

We initiate a systematic study of linear sketching over  $\mathbb{F}_2$ . For a given Boolean function treated as  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  a randomized  $\mathbb{F}_2$ -sketch is a distribution  $\mathcal{M}$  over  $d \times n$  matrices with elements over  $\mathbb{F}_2$  such that  $\mathcal{M}x$  suffices for computing  $f(x)$  with high probability. Such sketches for  $d \ll n$  can be used to design small-space distributed and streaming algorithms.

Motivated by these applications we study a connection between  $\mathbb{F}_2$ -sketching and a two-player one-way communication game for the corresponding XOR-function. We conjecture that  $\mathbb{F}_2$ -sketching is optimal for this communication game. Our results confirm this conjecture for multiple important classes of functions: 1) low-degree  $\mathbb{F}_2$ -polynomials, 2) functions with sparse Fourier spectrum, 3) most symmetric functions, 4) recursive majority function. These results rely on a new structural theorem that shows that  $\mathbb{F}_2$ -sketching is optimal (up to constant factors) for uniformly distributed inputs.

Furthermore, we show that (non-uniform) streaming algorithms that have to process random updates over  $\mathbb{F}_2$  can be constructed as  $\mathbb{F}_2$ -sketches for the uniform distribution. In contrast with the previous work of Li, Nguyen and Woodruff (STOC'14) who show an analogous result for linear sketches over integers in the adversarial setting our result does not require the stream length to be triply exponential in  $n$  and holds for streams of length  $\tilde{O}(n)$  constructed through uniformly random updates.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Communication complexity

**Keywords and phrases** Linear sketch, Streaming algorithms, XOR-functions, Communication complexity

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.8

---

<sup>1</sup> This work was supported by grants NSF CICI 1547360 and ONR N00014-15-1-2006.

<sup>2</sup> E.M. acknowledges the support of grant N00014-16-1-2227 from Office of Naval Research and of NSF awards CCF 1320105 and DMS-1737944 as well as support from Simons Think Tank on Geometry & Algorithms.

<sup>3</sup> S.S. acknowledges the support by the Singapore National Research Foundation under NRF RF Award No. NRF-NRFF2013-13. Part of this work was done when S. S. was a visiting research fellow at the Tata Institute of Fundamental Research, Mumbai.

<sup>4</sup> This work was supported by NSF award 1657477.



© Sampath Kannan, Elchanan Mossel,  
Swagato Sanyal, and Grigory Yaroslavl'tsev;  
licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 8; pp. 8:1–8:37



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**Related Version** This is an improved version of Kannan, Mossel, Yaroslavtsev <https://arxiv.org/pdf/1611.01879.pdf> [39] giving tight dependence on error in Theorem 4, a new result for degree- $d$   $\mathbb{F}_2$  polynomials and including several other changes. ECCC preprint is available at <https://eccc.weizmann.ac.il/report/2018/064/>.

**Acknowledgements** S.S. thanks Nikhil Mande, Rahul Jain and Anurag Anshu for helpful discussions.

## 1 Introduction

Linear sketching is the underlying technique behind many of the biggest algorithmic breakthroughs of the past two decades. It has played a key role in the development of streaming algorithms since [3] and most recently has been the key to modern randomized algorithms for numerical linear algebra (see survey [52]), graph compression (see survey [37]), dimensionality reduction, etc. Linear sketching is robust to the choice of a computational model and can be applied in settings as seemingly diverse as streaming, MapReduce as well as various other distributed models of computation including the congested clique model [19, 12, 23], allowing to save computational time, space and reduce communication in distributed settings. This remarkable versatility is based on properties of linear sketches enabled by linearity: simple and fast updates and mergeability of sketches computed on distributed data. Compatibility with fast numerical linear algebra packages makes linear sketching particularly attractive for applications.

Even more surprisingly linear sketching over the reals is known to be the best possible algorithmic approach (unconditionally) in certain settings. Most notably, under some mild conditions linear sketches are known to be almost space optimal for processing dynamic data streams [10, 31, 1]. Optimal bounds for streaming algorithms for a variety of computational problems can be derived through this connection by analyzing linear sketches rather than general algorithms. Examples include approximate matchings [5, 4], additive norm approximation [1] and frequency moments [31, 51].

In this paper we study the power of linear sketching over  $\mathbb{F}_2$ .<sup>5</sup> To the best of our knowledge no such systematic study currently exists as prior work focuses on sketching over the field of reals (or large finite fields as reals are represented as word-size bounded integers). Formally, for a random set  $\mathbf{S} \subseteq [n]$  let  $\chi_{\mathbf{S}} = \bigoplus_{i \in \mathbf{S}} x_i$ . Given a function  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  that needs to be evaluated over an input  $x = (x_1, \dots, x_n)$  we are looking for a distribution over  $k$  subsets  $\mathbf{S}_1, \dots, \mathbf{S}_k \subseteq [n]$  such that the following holds: for any input  $x$  given parities computed over these sets and denoted as  $\chi_{\mathbf{S}_1}(x), \chi_{\mathbf{S}_2}(x), \dots, \chi_{\mathbf{S}_k}(x)$ , it should be possible to compute  $f(x)$  with probability  $1 - \delta$ . While the switch from reals to  $\mathbb{F}_2$  might seem restrictive, we are unaware of any problem for which sketching over reals gives any advantage over  $\mathbb{F}_2$ . Furthermore, as shown very recently and subsequently to the early version of this work [39], almost all dynamic graph streaming algorithms<sup>6</sup> can be seen as  $\mathbb{F}_2$ -sketches [25] without losing optimality in space<sup>7</sup>.

<sup>5</sup> It is easy to see that sketching over finite fields can be significantly better than linear sketching over integers for certain computations. As an example, consider a function  $(x \bmod 2)$  (for an integer input  $x$ ) which can be trivially sketched with 1 bit over the field of two elements while any linear sketch over the integers requires word-size memory.

<sup>6</sup> With the only exception being the work of [24] on spectral graph sparsification.

<sup>7</sup> Technically [25] uses  $\mathbb{F}_3$ , but replacing  $\mathbb{F}_3$  with  $\mathbb{F}_2$  doesn't change their results.

In matrix form  $\mathbb{F}_2$ -sketching corresponds to multiplication over  $\mathbb{F}_2$  of the row vector  $x \in \mathbb{F}_2^n$  by a random  $n \times k$  matrix whose  $i$ -th column is a characteristic vector of the random parity  $\chi_{\mathbf{S}_i}$ :

$$(x_1 \ x_2 \ \dots \ x_n) \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ \chi_{\mathbf{S}_1} & \chi_{\mathbf{S}_2} & \dots & \chi_{\mathbf{S}_k} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} = (\chi_{\mathbf{S}_1}(x) \ \chi_{\mathbf{S}_2}(x) \ \dots \ \chi_{\mathbf{S}_k}(x))$$

This sketch alone should then be sufficient for computing  $f$  with high probability for any input  $x$ . This motivates us to define the *randomized linear sketch complexity* of a function  $f$  over  $\mathbb{F}_2$  as the smallest  $k$  which allows one to satisfy the above guarantee.

► **Definition 1** ( $\mathbb{F}_2$ -sketching). For a function  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  we define its *randomized linear sketch complexity*<sup>8</sup> over  $\mathbb{F}_2$  with error  $\delta$  (denoted as  $R_\delta^{lin}(f)$ ) as the smallest integer  $k$  such that there exists a distribution  $\chi_{\mathbf{S}_1}, \chi_{\mathbf{S}_2}, \dots, \chi_{\mathbf{S}_k}$  over  $k$  linear functions over  $\mathbb{F}_2$  and a postprocessing function  $g: \mathbb{F}_2^k \rightarrow \mathbb{F}_2$ <sup>9</sup> which satisfies:

$$\forall x \in \mathbb{F}_2^n: \Pr_{\mathbf{S}_1, \dots, \mathbf{S}_k} [f(x_1, x_2, \dots, x_n) = g(\chi_{\mathbf{S}_1}(x), \chi_{\mathbf{S}_2}(x), \dots, \chi_{\mathbf{S}_k}(x))] \geq 1 - \delta.$$

We note that while the above definition requires that  $f$  is computed exactly, most of our structural results including Theorem 4 can be extended to allow approximate computation of real-valued functions  $f: \mathbb{F}_2^n \rightarrow \mathbb{R}$  as shown in [54].

As we show in this paper the study of  $R_\delta^{lin}(f)$  is closely related to a certain communication problem. For  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  define the XOR-function  $f^+: \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  as  $f^+(x, y) = f(x + y)$  where  $x, y \in \mathbb{F}_2^n$ . Consider a communication game between two players Alice and Bob holding inputs  $x$  and  $y$  respectively. Given access to a shared source of random bits Alice has to send a single message to Bob so that he can compute  $f^+(x, y)$ . This is known as the one-way communication problem for XOR-functions.

► **Definition 2** (Randomized one-way communication complexity of XOR function). For a function  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  the *randomized one-way communication complexity* with error  $\delta$  (denoted as  $R_\delta^+(f^+)$ ) of its XOR-function is defined as the smallest size<sup>10</sup> (in bits) of the (randomized using public randomness) message  $M(x)$  from Alice to Bob which allows Bob to evaluate  $f^+(x, y)$  for any  $x, y \in \mathbb{F}_2^n$  with error probability at most  $\delta$ .

Communication complexity of XOR-functions has been recently studied extensively in the context of the log-rank conjecture (see e.g. [45, 55, 38, 28, 30, 47, 32, 49, 34, 18]). However, such studies either mostly focus on deterministic communication complexity or are specific to the two-way communication model. We discuss implications of this line of work for our  $\mathbb{F}_2$ -sketching model in our discussion of prior work.

<sup>8</sup> In the language of decision trees this can be interpreted as randomized non-adaptive parity decision tree complexity. We are unaware of any systematic study of this quantity either. Since heavy decision tree terminology seems excessive for our applications (in particular, sketching is done in one shot so there isn't a decision tree involved) we prefer to use a shorter and more descriptive name.

<sup>9</sup> Technically  $g$  can also depend on the sampled sets  $\mathbf{S}_1, \dots, \mathbf{S}_k$ , but all sketches used in this paper are oblivious to the choice of these sets.

<sup>10</sup> Formally the minimum here is taken over all possible protocols where for each protocol the size of the message  $M(x)$  refers to the largest size (in bits) of such message taken over all inputs  $x \in \mathbb{F}_2^n$ . See [27] for a formal definition.

It is easy to see that  $R_\delta^\rightarrow(f^+) \leq R_\delta^{lin}(f)$  as using shared randomness for sampling  $\mathbf{S}_1, \dots, \mathbf{S}_k$  Alice can just send  $k$  bits  $\chi_{\mathbf{S}_1}(x), \chi_{\mathbf{S}_2}(x), \dots, \chi_{\mathbf{S}_k}(x)$  to Bob who can for each  $i \in [k]$  compute  $\chi_{\mathbf{S}_i}(x+y) = \chi_{\mathbf{S}_i}(x) + \chi_{\mathbf{S}_i}(y)$ . This gives Bob an  $\mathbb{F}_2$ -sketch of  $f$  on  $x+y$  and hence suffices for computing  $f^+(x, y)$  with probability  $1 - \delta$ . The main open question raised in our work is whether the reverse inequality holds (at least approximately), thus implying the equivalence of the two notions.

► **Conjecture 3.** *Is it true that  $R_\delta^\rightarrow(f^+) = \tilde{\Theta}(R_\delta^{lin}(f))$  for every  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  and  $0 < \delta < 1/2$ ?*

In fact all known one-way protocols for XOR-functions can be seen as  $\mathbb{F}_2$ -sketches so it is natural to ask whether this is always true. In this paper we further motivate this conjecture through a number of examples of classes of functions for which it holds. One important such example from the previous work is a function  $Ham_{\geq k}$  which evaluates to 1 if and only if the Hamming weight of the input string is at least  $k$ . The corresponding XOR-function  $Ham_{\geq k}^+$  can be seen to have one-way communication complexity of  $\Theta(k \log k)$  via the small set disjointness lower bound of [9] and a basic upper bound based on random parities [20]. Conjecture 3 would imply that in order to prove a one-way disjointness lower bound it suffices to only consider  $\mathbb{F}_2$ -sketches.

A deterministic analog of Definition 1 requires that  $f(x) = g(\chi_{\alpha_1}(x), \chi_{\alpha_2}(x), \dots, \chi_{\alpha_k}(x))$  for a fixed choice of  $\alpha_1, \dots, \alpha_k \in \mathbb{F}_2^n$ . The smallest value of  $k$  which satisfies this definition is known to be equal to the Fourier dimension of  $f$  denoted as  $dim(f)$ . It corresponds to the smallest dimension of a linear subspace of  $\mathbb{F}_2^n$  that contains the entire spectrum of  $f$  (see Section 2.2 for a formal definition). In order to keep the notation uniform we also denote it as  $D^{lin}(f)$ . Most importantly, as shown in [38] an analog of Conjecture 3 holds without any loss in the deterministic case, i.e.  $D^\rightarrow(f^+) = dim(f) = D^{lin}(f)$ , where  $D^\rightarrow$  denotes the deterministic one-way communication complexity. This striking fact is one of the reasons why we suggest Conjecture 3 as an open problem.

## Previous work and our results

In the discussion below using Yao's principle we switch to the equivalent notion of distributional complexity of the above problems denoted as  $\mathcal{D}_\delta^\rightarrow$  and  $\mathcal{D}_\delta^{lin}$  respectively. For the formal definitions we refer to the reader to Section 2.1 and a standard textbook on communication complexity [27]. Equivalence between randomized and distributional complexities allows us to restate Conjecture 3 as  $\mathcal{D}_\delta^\rightarrow = \tilde{\Theta}(\mathcal{D}_\delta^{lin})$ .

For a fixed distribution  $\mu$  over  $\mathbb{F}_2^n$  we define  $\mathcal{D}_\delta^{lin, \mu}(f)$  to be the smallest dimension of an  $\mathbb{F}_2$ -sketch that correctly outputs  $f$  with probability  $1 - \delta$  over  $\mu$ . Similarly for a distribution  $\mu$  over  $(x, y) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$  we denote distributional one-way communication complexity of  $f$  with error  $\delta$  as  $\mathcal{D}_\delta^{\rightarrow, \mu}(f^+)$  (See Section 2 for a formal definition). Our first main result is an analog of Conjecture 3 for the uniform distribution  $U$  over  $(x, y)$  that matches the statement of the conjecture up to constant factors:

► **Theorem 4.** *For any  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  it holds that  $\mathcal{D}_{1/9}^{\rightarrow, U}(f^+) \geq \frac{1}{6} \cdot \mathcal{D}_{1/3}^{lin, U}(f)$ .*

In order to prove Theorem 4 we introduce the notion of an *approximate Fourier dimension* (Definition 13) that extends the definition of exact Fourier dimension to allow that only  $1 - \epsilon$  fraction of the total “energy” in  $f$ 's spectrum should be contained in the linear subspace. The key ingredient in the proof is a structural theorem, Theorem 14, that characterizes both  $\mathcal{D}_\delta^{lin, U}(f)$  and  $\mathcal{D}_\delta^{\rightarrow, U}(f^+)$  in terms of  $f$ 's approximate Fourier dimension.

Using Theorem 14 we confirm Conjecture 3 for several well-studied classes of functions in Section 4. It is important to note that while we could have stated these results for randomized one-way communication it is critical that all lower bounds in this section hold for uniform distribution in order to derive our results for random streams in Section 5.

### Low-degree $\mathbb{F}_2$ polynomials

Low-degree  $\mathbb{F}_2$  polynomials have been extensively studied in theoretical computer science in various contexts: learning theory (Mossel, O’Donnell and Servedio [40]), property testing (Rubinfeld and Sudan [42], Bhattacharyya *et al.* [6], Alon *et al.* [2]), pseudorandomness (Bogdanov and Viola [8], Lovett [33], Viola [50]), communication complexity (Tsang *et al.* [49]), etc.

Tsang *et al.* [49] studied deterministic two-way communication protocols for XOR-functions with low  $\mathbb{F}_2$ -degree. They gave an upper bound on deterministic communication complexity of  $f^+$  in terms of the spectral norm and the  $\mathbb{F}_2$ -degree of  $f$ . Their result was obtained by observing that the communication complexity of  $f^+$  is bounded above by the parity decision tree complexity of  $f$ , and then bounding the latter. In this work, we prove a lower bound on the randomized one-way communication complexity of  $f^+$  in terms of the Fourier dimension of  $f$  and the  $\mathbb{F}_2$ -degree of  $f$ , denoted as  $d$ . We prove the following result:

$$D^{lin}(f) = O\left(R_{1/3}^{\rightarrow}(f^+) \cdot d\right).$$

In the regime  $d = O(1)$ , the above result implies that use of randomness does not enable us to design a better linear-sketching or a one-way communication protocol. Furthermore, since  $R_{1/3}^{lin}(f) \leq D^{lin}(f)$ , the above result implies Conjecture 3 for constant degree  $\mathbb{F}_2$ -polynomials. For  $\mathbb{F}_2$  polynomials with bounded spectral norm this implies a new bound on Fourier dimension shown in Corollary 23:  $D^{lin}(f) = \dim(f) = O(d\|\hat{f}\|_1^2)$  improving a result of Tsang *et al.* for  $d = \omega\left(\log^{1/3}\|\hat{f}\|_1\right)$ .

### Address function and Fourier sparsity

The number  $s$  of non-zero Fourier coefficients of  $f$  (known as Fourier sparsity) is one of the key quantities in the analysis of Boolean functions. It also plays an important role in the recent work on log-rank conjecture for XOR-functions [49, 46]. A recent result by Sanyal [44] shows that for Boolean functions  $\dim(f) = O(\sqrt{s} \log s)$ , namely all non-zero Fourier coefficients are contained in a subspace of a polynomially smaller dimension. This bound is almost tight as the *address function* (see Section 4.2 for a definition) exhibits a quadratic gap. A direct implication of Sanyal’s result is a deterministic  $\mathbb{F}_2$ -sketching upper bound of  $O(\sqrt{s} \log s)$  for any  $f$  with Fourier sparsity  $s$ . As we show in Section 4.2 this dependence on sparsity can’t be improved even if randomization is allowed.

### Symmetric functions

A function  $f$  is symmetric if it only depends on the Hamming weight of its input. In Section 4.3 we show that Conjecture 3 holds for all symmetric functions which are not too close to a constant function or the parity function  $\sum_i x_i$ , where the sum is taken over  $\mathbb{F}_2$ .

### Composition theorem for recursive majority

As an example of a composition theorem we give such a theorem for recursive majority. For an odd integer  $n$  the majority function  $Maj_n$  is defined to be 1 if and only if the

Hamming weight of the input is greater than  $n/2$ . Of particular interest is the recursive majority function  $Maj_3^{\circ k}$  that corresponds to  $k$ -fold composition of  $Maj_3$  for  $k = \log_3 n$ . This function was introduced by Boppana [43] and serves as an important example of various properties of Boolean functions, most importantly in randomized decision tree complexity ([43, 22, 36, 29, 35]), deterministic parity decision tree complexity [7] and communication complexity [22, 13].

In Section 4.4 we use Theorem 14 to obtain the following result:

► **Theorem 5.** *For any  $\epsilon \in [0, \frac{1}{2}]$ ,  $\xi > 4\epsilon^2$  and  $k = \log_3 n$  it holds that:*

$$\mathcal{D}_{\frac{1-\xi}{6}}^{\rightarrow, U}(Maj_3^{\circ k+}) = \Omega(\epsilon^2 n).$$

### Applications to streaming and distributed computing

In the turnstile streaming model of computation a vector  $x$  of dimension  $n$  is updated through a sequence of additive updates applied to its coordinates and the goal of the algorithm is to be able to output  $f(x)$  at any point during the stream while using space that is sublinear in  $n$ . In the real-valued case we have either  $x \in [0, m]^n$  or  $x \in [-m, m]^n$  for some universal upper bound  $m$  and updates can be increments or decrements to  $x$ 's coordinates of arbitrary magnitude.

For  $x \in \mathbb{F}_2^n$  additive updates have a particularly simple form as they always flip the corresponding coordinate of  $x$ . In the streaming literature this model is referred to as the XOR update model (see e.g. [48]) Note that XOR updates can't be handled using standard turnstile streaming algorithms as only the coordinate but not the sign of the update is given. As we show in Section 5.2 it is easy to see based on the recent work of [10, 31, 1] that in the adversarial streaming setting the space complexity of turnstile streaming algorithms over  $\mathbb{F}_2$  is determined by the  $\mathbb{F}_2$ -sketch complexity of the function of interest. However, this proof technique only works for very long streams which are unrealistic in practice – the length of the adversarial stream has to be triply exponential in  $n$  in order to enforce linear behavior. Large stream length requirement is inherent in the proof structure in this line of work and while one might expect to improve triply exponential dependence on  $n$  at least an exponential dependence appears necessary, which is a major limitation of this approach.

As we show in Section 5.1 it follows directly from our Theorem 4 that turnstile streaming algorithms that achieve low error probability under random  $\mathbb{F}_2$  updates might as well be  $\mathbb{F}_2$ -sketches. For two natural choices of the random update model short streams of length either  $O(n)$  or  $O(n \log n)$  suffice for our reduction. We stress that our lower bounds are also stronger than the worst-case adversarial lower bounds as they hold under an average-case scenario. Furthermore, our Conjecture 3 would imply that space optimal turnstile streaming algorithms over  $\mathbb{F}_2$  have to be linear sketches for adversarial streams of length only  $2n$ . We believe that such result will also help show an analogous statement for real-valued linear sketches thus removing the triply exponential in  $n$  stream length assumption of [31, 1].

By linearity all  $\mathbb{F}_2$ -sketching upper bounds are also applicable in the distributed setting where two parties Alice and Bob need to send messages to the coordinator who is required to output  $f^+$ . This is also known as the Simultaneous Message Passing (SMP) model and all our one-way lower bounds hold in this model as well.

### Other previous work

Closely related to ours is work on communication protocols for XOR-functions [45, 38, 49, 18]. In particular [38] presents two basic one-way communication protocols based on random

parities. The first one, stated as Fact 60 generalizes the classic communication protocol for equality. The second one uses the result of Grolmusz [17] and implies that  $\ell_1$ -sampling of Fourier characters gives a randomized  $\mathbb{F}_2$ -sketch of size  $O(\|\hat{f}\|_1^2)$  (for constant error).

In [18] structural results about deterministic two-way communication protocols for XOR-functions have been obtained. In particular, they show that the parity decision tree complexity of  $f$  is  $O(D(f^+)^6)$ . The key difference between our work and [18] lies in our focus on randomized protocols. In [18] it is left as the main open problem whether randomized parity decision tree complexity can be bounded by  $\text{poly}(R(f^+))$ . Our results can be seen as a step towards resolving this open problem in one-way communication setting. Full resolution of Conjecture 3 would show that the conjecture of [18] holds even without polynomial loss for one-way communication as we show for all the classes considered in Section 4.

Another line of work that is closely related to ours is the study of the two-player simultaneous message passing model (SMP). This model can also allow to prove lower bounds on  $\mathbb{F}_2$ -sketching complexity. Since our results hold for one-way communication they also hold in the SMP model. Moreover, in the context of our work there is no substantial difference as for product distributions the two models are essentially equivalent. Recent results in the SMP model include [38, 30, 32].

While decision tree literature is not directly relevant to us since our model doesn't allow adaptivity we remark that there has been interest recently in the study of (adaptive) deterministic parity decision trees [7] and non-adaptive deterministic parity decision trees [46, 44]. As mentioned above, our model can be interpreted as non-adaptive randomized parity decision trees and to the best of our knowledge it hasn't been studied explicitly before. Another related model is that of *parity kill numbers*. In this model a composition theorem has recently been shown by [41] but the key difference is again adaptivity.

Finally recent developments in the line of work on lifting theorems such as [15, 14] might suggest that such results might be applied in our context. However for our purposes we would need a lifting theorem for the XOR gadget and to the best of our knowledge no such result is known for randomized one-way communication.

## Organization

The rest of this paper is organized as follows. In Section 2 we introduce the required background from communication complexity and Fourier analysis of Boolean functions. In Section 3 we prove Theorem 4. In Section 4 we give applications of this theorem for recursive majority (Theorem 5), address function, low-degree  $\mathbb{F}_2$  polynomials and symmetric functions. In Section 5 we describe applications to streaming.

In Appendix B we give some basic results about deterministic  $\mathbb{F}_2$ -sketching (or Fourier dimension) of composition and convolution of functions. We also present a basic lower bound argument based on affine dispersers. In Appendix C we give some basic results about randomized  $\mathbb{F}_2$ -sketching including a lower bound based on extractors and a classic protocol based on random parities which we use as a building block in our sketch for LTFs. We also present evidence for why an analog of Theorem 14 doesn't hold for arbitrary distributions. In Appendix D we show a lower bound for one-bit protocols making progress towards resolving Conjecture 3.

## 2 Preliminaries

For an integer  $n$  we use notation  $[n] = \{1, \dots, n\}$ . For integers  $n \leq m$  we use notation  $[n, m] = \{n, \dots, m\}$ . For an arbitrary domain  $\mathcal{D}$  we denote the uniform distribution over

this domain as  $U(\mathcal{D})$ . We use the notation  $x, x' \sim U(\mathcal{D})$  to denote that  $x$  and  $x'$  are sampled uniformly at random and independently from  $\mathcal{D}$ . The variance of a random variable  $X$  is denoted by  $\text{Var}[X]$ . For a vector  $x$  and  $p \geq 1$  we denote the  $p$ -norm of  $x$  as  $\|x\|_p$  and reserve the notation  $\|x\|_0$  for the Hamming weight.

## 2.1 Communication complexity

Consider a function  $f: \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  and a distribution  $\mu$  over  $\mathbb{F}_2^n \times \mathbb{F}_2^n$ . The *one-way distributional complexity* of  $f$  with respect to  $\mu$ , denoted as  $\mathcal{D}_\delta^{\rightarrow, \mu}(f)$  is the smallest communication cost of a one-way deterministic protocol that outputs  $f(x, y)$  with probability at least  $1 - \delta$  over the inputs  $(x, y)$  drawn from the distribution  $\mu$ . The *one-way distributional complexity* of  $f$  denoted as  $\mathcal{D}_\delta^{\rightarrow}(f)$  is defined as  $\mathcal{D}_\delta^{\rightarrow}(f) = \sup_\mu \mathcal{D}_\delta^{\rightarrow, \mu}(f)$ . By Yao's minimax theorem [53] it follows that  $R_\delta^{\rightarrow}(f) = \mathcal{D}_\delta^{\rightarrow}(f)$ . *One-way communication complexity over product distributions* is defined as  $\mathcal{D}_\delta^{\rightarrow, \times}(f) = \sup_{\mu = \mu_x \times \mu_y} \mathcal{D}_\delta^{\rightarrow, \mu}(f)$  where  $\mu_x$  and  $\mu_y$  are distributions over  $\mathbb{F}_2^n$ .

With every two-party function  $f: \mathbb{F}_2^n \times \mathbb{F}_2^n$  we associate a *communication matrix*  $M^f \in \mathbb{F}_2^{2^n \times 2^n}$  with entries  $M_{x,y}^f = f(x, y)$ . We say that a deterministic protocol  $M(x)$  with length  $t$  of the message that Alice sends to Bob partitions the rows of this matrix into  $2^t$  *combinatorial rectangles* where each rectangle contains all rows of  $M^f$  corresponding to the same fixed message  $y \in \{0, 1\}^t$ .

## 2.2 Fourier analysis

We consider functions<sup>11</sup> from  $\mathbb{F}_2^n$  to  $\mathbb{R}$ . For any fixed  $n \geq 1$ , the space of these functions forms an inner product space with the inner product  $\langle f, g \rangle = \mathbb{E}_{x \in \mathbb{F}_2^n} [f(x)g(x)] = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)g(x)$ . The  $\ell_2$  norm of  $f: \mathbb{F}_2^n \rightarrow \mathbb{R}$  is  $\|f\|_2 = \sqrt{\langle f, f \rangle} = \sqrt{\mathbb{E}_x [f(x)^2]}$  and the  $\ell_2$  distance between two functions  $f, g: \mathbb{F}_2^n \rightarrow \mathbb{R}$  is the  $\ell_2$  norm of the function  $f - g$ . In other words,  $\|f - g\|_2 = \sqrt{\langle f - g, f - g \rangle} = \sqrt{\frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (f(x) - g(x))^2}$ .

For  $\alpha \in \mathbb{F}_2^n$ , the *character*  $\chi_\alpha: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  is the function defined by  $\chi_\alpha(x) = (-1)^{\alpha \cdot x}$ . Characters form an orthonormal basis as  $\langle \chi_\alpha, \chi_\beta \rangle = \delta_{\alpha\beta}$  where  $\delta$  is the Kronecker symbol. The *Fourier coefficient* of  $f: \mathbb{F}_2^n \rightarrow \mathbb{R}$  corresponding to  $\alpha$  is  $\hat{f}(\alpha) = \mathbb{E}_x [f(x)\chi_\alpha(x)]$ . The *Fourier transform* of  $f$  is the function  $\hat{f}: \mathbb{F}_2^n \rightarrow \mathbb{R}$  that returns the value of each Fourier coefficient of  $f$ . We use notation  $\text{Spec}(f) = \{\alpha \in \mathbb{F}_2^n : \hat{f}(\alpha) \neq 0\}$  to denote the set of all non-zero Fourier coefficients of  $f$ . The Fourier  $\ell_1$  norm, or the *spectral norm* of  $f$ , is defined as  $\|\hat{f}\|_1 := \sum_{\alpha \in \mathbb{F}_2^n} |\hat{f}(\alpha)|$ .

► **Fact 6** (Parseval's identity). *For any  $f: \mathbb{F}_2^n \rightarrow \mathbb{R}$  it holds that*

$$\|f\|_2 = \|\hat{f}\|_2 = \sqrt{\sum_{\alpha \in \mathbb{F}_2^n} \hat{f}(\alpha)^2}.$$

Moreover, if  $f: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  then  $\|f\|_2 = \|\hat{f}\|_2 = 1$ .

We use notation  $A \leq \mathbb{F}_2^n$  to denote the fact that  $A$  is a linear subspace of  $\mathbb{F}_2^n$ .

<sup>11</sup> In all Fourier-analytic arguments Boolean functions are treated as functions of the form  $f: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  where 0 is mapped to 1 and 1 is mapped to -1. Otherwise we use these two notations interchangeably.



► **Definition 7** (Fourier dimension). The *Fourier dimension* of  $f: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  denoted as  $\dim(f)$  is the smallest integer  $k$  such that there exists  $A \leq \mathbb{F}_2^n$  of dimension  $k$  for which  $\text{Spec}(f) \subseteq A$ .

We say that  $A \leq \mathbb{F}_2^n$  is a *standard subspace* if it has a basis  $v_1, \dots, v_d$  where each  $v_i$  has Hamming weight equal to 1. An *orthogonal subspace*  $A^\perp$  is defined as:

$$A^\perp = \{\gamma \in \mathbb{F}_2^n : \forall x \in A \quad \gamma \cdot x = 0\}.$$

An *affine subspace* (or coset) of  $\mathbb{F}_2^n$  of the form  $A = H + a$  for some  $H \leq \mathbb{F}_2^n$  and  $a \in \mathbb{F}_2^n$  is defined as:

$$A = \{\gamma \in \mathbb{F}_2^n : \forall x \in H^\perp \quad \gamma \cdot x = a \cdot x\}.$$

We now introduce notation for restrictions of functions to affine subspaces.

► **Definition 8.** Let  $f: \mathbb{F}_2^n \rightarrow \mathbb{R}$  and  $z \in \mathbb{F}_2^n$ . We define  $f^{+z}: \mathbb{F}_2^n \rightarrow \mathbb{R}$  as  $f^{+z}(x) = f(x + z)$ .

► **Fact 9.** The Fourier coefficients of  $f^{+z}$  are  $\widehat{f^{+z}}(\gamma) = (-1)^{\gamma \cdot z} \widehat{f}(\gamma)$  and hence:

$$f^{+z} = \sum_{S \in \mathbb{F}_2^n} \widehat{f}(S) \chi_S(z) \chi_S.$$

► **Definition 10** (Coset restriction). For  $f: \mathbb{F}_2^n \rightarrow \mathbb{R}$ ,  $z \in \mathbb{F}_2^n$  and  $H \leq \mathbb{F}_2^n$  we write  $f_H^{+z}: H \rightarrow \mathbb{R}$  for the restriction of  $f$  to  $H + z$ .

► **Definition 11** (Convolution). For two functions  $f, g: \mathbb{F}_2^n \rightarrow \mathbb{R}$  their convolution  $(f * g): \mathbb{F}_2^n \rightarrow \mathbb{R}$  is defined as  $(f * g)(x) = \mathbb{E}_{y \sim U(\mathbb{F}_2^n)} [f(y)g(x + y)]$ .

For  $S \in \mathbb{F}_2^n$  the corresponding Fourier coefficient of convolution is given as  $\widehat{f * g}(S) = \widehat{f}(S)\widehat{g}(S)$ .

### 3 $\mathbb{F}_2$ -sketching over the uniform distribution

We use the following definition of Fourier concentration that plays an important role in learning theory [26]. As mentioned above in all Fourier-analytic arguments we replace the range of the functions with  $\{+1, -1\}$ .

► **Definition 12** (Fourier concentration). The spectrum of a function  $f: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  is  $\epsilon$ -concentrated on a collection of Fourier coefficients  $Z \subseteq \mathbb{F}_2^n$  if  $\sum_{\alpha \in Z} \widehat{f}^2(\alpha) \geq \epsilon$ .

We now introduce the notion of *approximate Fourier dimension* of a Boolean function.

► **Definition 13** (Approximate Fourier dimension). Let  $\mathcal{A}_k$  be the set of all linear subspaces of  $\mathbb{F}_2^n$  of dimension  $k$ . For  $f: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  and  $\epsilon \in (0, 1]$  the  $\epsilon$ -approximate Fourier dimension  $\dim_\epsilon(f)$  is defined as:

$$\dim_\epsilon(f) = \min \left\{ k: \exists A \in \mathcal{A}_k: \sum_{\alpha \in A} \widehat{f}^2(\alpha) \geq \epsilon \right\}.$$

The following theorem shows that for uniformly distributed inputs, both the one-way communication complexity of  $f^+$  and the linear sketch complexity of  $f$  are characterized by the approximate Fourier dimension of  $f$ . An immediate corollary is that, up to some slack in the dependence on the probability of error, the one-way communication complexity under the uniform distribution matches the linear sketch complexity. We note that the lower bounds given by this theorem are stronger than the basic extractor lower bound given in Appendix C.1. See Remark C.1 for further discussion.

## 8:10 Linear Sketching over $\mathbb{F}_2$

► **Theorem 14.** Let  $f: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  be a Boolean function. Let  $\xi \in [0, 1]$  and  $\gamma < \frac{1-\sqrt{\xi}}{2}$ . Let  $d = \dim_\xi(f)$ . Then,

$$1. \mathcal{D}_{(1-\xi)/2}^{\rightarrow, U}(f^+) \leq \mathcal{D}_{(1-\xi)/2}^{lin, U}(f) \leq d, \quad 2. \mathcal{D}_\gamma^{lin, U}(f) \geq d, \quad 3. \mathcal{D}_{(1-\xi)/6}^{\rightarrow, U} \geq \frac{d}{6}.$$

**Proof.**

**Part 1.** <sup>12</sup> Since  $d = \dim_\xi(f)$ , there exists a subspace  $A \leq \mathbb{F}_2^n$  of dimension at most  $d$  which satisfies  $\sum_{\alpha \in A} \hat{f}^2(\alpha) \geq \xi$ . Let  $g: \mathbb{F}_2^n \rightarrow \mathbb{R}$  be a function defined by its Fourier transform as follows:

$$\hat{g}(\alpha) = \begin{cases} \hat{f}(\alpha), & \text{if } \alpha \in A \\ 0, & \text{otherwise.} \end{cases}$$

Consider drawing a random variable  $\theta$  from the distribution with p.d.f  $1 - |\theta|$  over  $[-1, 1]$ .

► **Proposition 15.** For all  $t$  such that  $-1 \leq t \leq 1$  and  $z \in \{+1, -1\}$  random variable  $\theta$  satisfies:

$$\Pr_\theta[\text{sgn}(t - \theta) \neq z] \leq \frac{1}{2}(z - t)^2.$$

**Proof.** W.l.o.g we can assume  $z = 1$  as the case  $z = -1$  is symmetric. Then we have:

$$\Pr_\theta[\text{sgn}(t - \theta) \neq 1] = \int_t^1 (1 - |\gamma|) d\gamma \leq \int_t^1 (1 - \gamma) d\gamma = \frac{1}{2}(1 - t)^2. \quad \blacktriangleleft$$

Define a family of functions  $g_\theta: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  as  $g_\theta(x) = \text{sgn}(g(x) - \theta)$ . Then we have:

$$\begin{aligned} \mathbb{E}_\theta \left[ \Pr_{x \sim \mathbb{F}_2^n} [g_\theta(x) \neq f(x)] \right] &= \mathbb{E}_{x \sim \mathbb{F}_2^n} \left[ \Pr_\theta [g_\theta(x) \neq f(x)] \right] \\ &= \mathbb{E}_{x \sim \mathbb{F}_2^n} \left[ \Pr_\theta [\text{sgn}(g(x) - \theta) \neq f(x)] \right] \\ &\leq \mathbb{E}_{x \sim \mathbb{F}_2^n} \left[ \frac{1}{2}(f(x) - g(x))^2 \right] \text{ (by Proposition 15)} \\ &= \frac{1}{2} \|f - g\|_2^2. \end{aligned}$$

Using the definition of  $g$  and Parseval we have:

$$\frac{1}{2} \|f - g\|_2^2 = \frac{1}{2} \|\widehat{f - g}\|_2^2 = \frac{1}{2} \|\hat{f} - \hat{g}\|_2^2 = \frac{1}{2} \sum_{\alpha \notin A} \hat{f}^2(\alpha) \leq \frac{1 - \xi}{2}.$$

Thus, there exists a choice of  $\theta$  such that  $g_\theta$  achieves error at most  $\frac{1-\xi}{2}$ . Clearly  $g_\theta$  can be computed based on the  $d$  parities forming a basis for  $A$  and hence  $\mathcal{D}_{(1-\xi)/2}^{lin, U}(f) \leq d$ .

<sup>12</sup>This argument is a refinement of the standard “sign trick” from learning theory which approximates a Boolean function by taking a sign of its real-valued approximation under  $\ell_2$ .

**Part 2.** Fix any deterministic sketch that uses  $d - 1$  parities  $\chi_{\alpha_1}, \dots, \chi_{\alpha_{d-1}}$  and let  $S = (\alpha_1, \dots, \alpha_{d-1})$ . For fixed values of these sketches  $b = (b_1, \dots, b_{d-1})$  where  $b_i = \chi_{\alpha_i}(x)$  we denote the resulting affine restriction of  $f$  as  $f|_{(S,b)}$ . Using the standard expression for the Fourier coefficients of an affine restriction the constant Fourier coefficient of the restricted function is given as:

$$\widehat{f|_{(S,b)}}(\emptyset) = \sum_{Z \subseteq [d-1]} (-1)^{\sum_{i \in Z} b_i} \hat{f}\left(\sum_{i \in Z} \alpha_i\right).$$

Thus, we have:

$$\widehat{f|_{(S,b)}}^2(\emptyset) = \sum_{Z \subseteq [d-1]} \hat{f}^2\left(\sum_{i \in Z} \alpha_i\right) + \sum_{Z_1 \neq Z_2 \subseteq [d-1]} (-1)^{\sum_{i \in Z_1 \Delta Z_2} b_i} \hat{f}\left(\sum_{i \in Z_1} \alpha_i\right) \hat{f}\left(\sum_{i \in Z_2} \alpha_i\right).$$

Taking expectation over a uniformly random  $b \sim U(\mathbb{F}_2^d)$  we have:

$$\begin{aligned} & \mathbb{E}_{b \sim U(\mathbb{F}_2^d)} \left[ \widehat{f|_{(S,b)}}^2(\emptyset) \right] \\ &= \mathbb{E}_{b \sim U(\mathbb{F}_2^d)} \left[ \sum_{Z \subseteq [d-1]} \hat{f}^2\left(\sum_{i \in Z} \alpha_i\right) + \sum_{Z_1 \neq Z_2 \subseteq [d-1]} (-1)^{\sum_{i \in Z_1 \Delta Z_2} b_i} \hat{f}\left(\sum_{i \in Z_1} \alpha_i\right) \hat{f}\left(\sum_{i \in Z_2} \alpha_i\right) \right] \\ &= \sum_{Z \subseteq [d-1]} \hat{f}^2\left(\sum_{i \in Z} \alpha_i\right). \end{aligned}$$

The latter sum is the sum of squared Fourier coefficients over a linear subspace of dimension  $d - 1 < \dim_{\xi}(f)$ , and hence is strictly less than  $\xi$ . Using Jensen's inequality:

$$\mathbb{E}_{b \sim U(\mathbb{F}_2^d)} \left[ |\widehat{f|_{(S,b)}}(\emptyset)| \right] \leq \sqrt{\mathbb{E}_{b \sim U(\mathbb{F}_2^d)} \left[ \widehat{f|_{(S,b)}}^2(\emptyset) \right]} < \sqrt{\xi}.$$

For a fixed restriction  $(S, b)$  if  $|\widehat{f|_{(S,b)}}(\emptyset)| < \alpha$  then  $|\Pr[f|_{(S,b)} = 1] - \Pr[f|_{(S,b)} = -1]| < \alpha$  and hence no algorithm can predict the value of the restricted function on this coset with probability at least  $\frac{1+\alpha}{2}$ . Thus no algorithm can predict  $f|_{(\alpha_1, b_1), \dots, (\alpha_{d-1}, b_{d-1})}$  for a uniformly random choice of  $(b_1, \dots, b_{d-1})$ , and hence also on a uniformly at random chosen  $x$ , with probability at least  $\frac{1+\sqrt{\xi}}{2}$ .

**Part 3.** We will need the following fact about entropy of a binary random variable. The proof is given in the appendix (Section A.1).

► **Fact 16.** For any random variable  $X$  supported on  $\{1, -1\}$ ,  $H(X) \leq 1 - \frac{1}{2}(\mathbb{E}X)^2$ .

We will need the following proposition that states that random variables taking value in  $\{1, -1\}$  that are highly biased have low variance. The proof of Proposition 17 can be found in the appendix (Section E.1).

► **Proposition 17.** Let  $X$  be a random variable taking values in  $\{1, -1\}$ . Define  $p := \min_{b \in \{1, -1\}} \Pr[X = b]$ . Then  $\text{Var}[X] \in [2p, 4p]$ .

## 8:12 Linear Sketching over $\mathbb{F}_2$

In the next two lemmas, we look into the structure of a one-way communication protocol for  $f^+$ , and analyze its performance when the inputs are uniformly distributed. We give a lower bound on the number of bits of information that any correct randomized one-way protocol reveals about Alice's input, in terms of the linear sketching complexity of  $f$  for uniform distribution<sup>13</sup>.

The next lemma bounds the probability of error of a one-way protocol from below in terms of the Fourier coefficients of  $f$ , and the conditional distributions of different parities of Alice's input conditioned on Alice's random message.

► **Lemma 18.** *Let  $\epsilon \in [0, \frac{1}{2})$ . Let  $\Pi$  be a deterministic one-way protocol for  $f^+$  such that  $\Pr_{x,y \sim U(\mathbb{F}_2^n)}[\Pi(x,y) \neq f^+(x,y)] \leq \epsilon$ . Let  $M$  denote the distribution of the random message sent by Alice to Bob in  $\Pi$ . For any fixed message  $m$  sent by Alice, let  $D_m$  denote the distribution of Alice's input  $x$  conditioned on the event that  $M = m$ . Then,*

$$4\epsilon \geq \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}^2(\alpha) \cdot \left( 1 - \mathbb{E}_{m \sim M} \left( \mathbb{E}_{x \sim D_m} [\chi_\alpha(x)] \right)^2 \right).$$

**Proof.** For any fixed input  $y$  of Bob, define  $\epsilon_m^{(y)} := \Pr_{x \sim D_m}[\Pi(x,y) \neq f^+(x,y)]$ . Thus,

$$\epsilon \geq \mathbb{E}_{m \sim M} \mathbb{E}_{y \sim U(\mathbb{F}_2^n)} [\epsilon_m^{(y)}]. \quad (1)$$

Note that the output of the protocol is determined by Alice's message and  $y$ . Hence for a fixed message and Bob's input, if the restricted function is largely unbiased, then any protocol is forced to commit an error with high probability. Formally,

$$\epsilon_m^{(y)} \geq \min_{b \in \{1, -1\}} \Pr_{x \sim D_m} [f^+(x,y) = b] \geq \frac{\text{Var}_{x \sim D_m} [f^+(x,y)]}{4}. \quad (2)$$

Since  $f^+(\cdot, \cdot)$  takes values in  $\{+1, -1\}$ , the second inequality follows from Proposition 17. Now,

$$\begin{aligned} \text{Var}_{x \sim D_m} [f^+(x,y)] &= 1 - \left( \mathbb{E}_{x \sim D_m} [f^+(x,y)] \right)^2 \quad (\text{since } f^+(x,y) \in \{1, -1\}) \\ &= 1 - \left( \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha) \chi_\alpha(y) \mathbb{E}_{x \sim D_m} [\chi_\alpha(x)] \right)^2 \quad (\text{by Fact 9 and linearity of expectation}) \\ &= 1 - \left( \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}^2(\alpha) \left( \mathbb{E}_{x \sim D_m} [\chi_\alpha(x)] \right)^2 \right. \\ &\quad \left. + \sum_{(\alpha_1, \alpha_2) \in \mathbb{F}_2^n \times \mathbb{F}_2^n : \alpha_1 \neq \alpha_2} \widehat{f}(\alpha_1) \widehat{f}(\alpha_2) \chi_{\alpha_1 + \alpha_2}(y) \mathbb{E}_{x \sim D_m} [\chi_{\alpha_1}(x)] \mathbb{E}_{x \sim D_m} [\chi_{\alpha_2}(x)] \right). \end{aligned}$$

Taking expectation over  $y$  we have:

$$\mathbb{E}_{y \sim U(\mathbb{F}_2^n)} [\text{Var}_{x \sim D_m} [f^+(x,y)]] = 1 - \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}^2(\alpha) \left( \mathbb{E}_{x \sim D_m} [\chi_\alpha(x)] \right)^2. \quad (3)$$

<sup>13</sup>We thus prove an *information complexity* lower bound. See, for example, [21] for an introduction to information complexity.

Taking expectation over messages it follows from (1), (2) and (3) that,

$$\begin{aligned} 4\epsilon &\geq 1 - \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}^2(\alpha) \cdot \mathbb{E}_{m \sim M} \left( \mathbb{E}_{x \sim D_m} [\chi_\alpha(x)] \right)^2 \\ &= \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}^2(\alpha) \cdot \left( 1 - \mathbb{E}_{m \sim M} \left( \mathbb{E}_{x \sim D_m} [\chi_\alpha(x)] \right)^2 \right). \end{aligned} \quad (4)$$

The second equality above follows from the Parseval's identity (Fact 6). The lemma follows.  $\blacktriangleleft$

Let  $\epsilon := \frac{1-\xi}{6}$ . Let  $\Pi$  be a deterministic protocol such that  $\Pr_{x,y \sim U(\mathbb{F}_2^n)}[\Pi(x,y) \neq f^+(x,y)] \leq \epsilon$ , with optimal cost  $c_\Pi := \mathcal{D}_\epsilon^{\rightarrow,U}(f^+) = \mathcal{D}_{\frac{1-\xi}{6}}^{\rightarrow,U}(f^+)$ . Let  $M$  denote the distribution of the random message sent by Alice to Bob in  $\Pi$ . For any fixed message  $m$  sent by Alice, let  $D_m$  denote the distribution of Alice's input  $x$  conditioned on the event that  $M = m$ . To prove Part 3 of Theorem 14 we use the protocol  $\Pi$  to come up with a subspace of  $\mathbb{F}_2^n$ . Next, in Lemma 19 (a) we prove, using Lemma 18, that  $f$  is  $\xi$ -concentrated on that subspace. In Lemma 19 (b) we upper bound the dimension of that subspace in terms of  $c_\Pi$ .

**► Lemma 19.** *Let  $\mathcal{A} := \{\alpha \in \mathbb{F}_2^n : \mathbb{E}_{m \sim M} (\mathbb{E}_{x \sim D_m} \chi_\alpha(x))^2 \geq \frac{1}{3}\} \subseteq \mathbb{F}_2^n$ . Let  $\ell = \dim(\text{span}(\mathcal{A}))$ . Then,*

(a)  $\ell \geq d$ .

(b)  $\ell \leq 6c_\Pi$ .

**Proof.** (a) We prove part (a) by showing that  $f$  is  $\xi$ -concentrated on  $\text{span}(\mathcal{A})$ . By Lemma 18 we have that

$$\begin{aligned} 4\epsilon &\geq \sum_{\alpha \in \text{span}(\mathcal{A})} \widehat{f}^2(\alpha) \cdot \left( 1 - \mathbb{E}_{m \sim M} \left( \mathbb{E}_{x \sim D_m} \chi_\alpha(x) \right)^2 \right) + \\ &\quad \sum_{\alpha \notin \text{span}(\mathcal{A})} \widehat{f}^2(\alpha) \cdot \left( 1 - \mathbb{E}_{m \sim M} \left( \mathbb{E}_{x \sim D_m} \chi_\alpha(x) \right)^2 \right) \\ &> \frac{2}{3} \cdot \sum_{\alpha \notin \text{span}(\mathcal{A})} \widehat{f}^2(\alpha). \end{aligned}$$

Thus  $\sum_{\alpha \notin \text{span}(\mathcal{A})} \widehat{f}^2(\alpha) < 6\epsilon$ . Hence,  $\sum_{\alpha \in \text{span}(\mathcal{A})} \widehat{f}^2(\alpha) \geq 1 - 6\epsilon = \xi$ . Hence we have  $\ell = \dim(\text{span}(\mathcal{A})) \geq \dim_\xi(f) = d$ .

(b) Notice that  $\chi_\alpha(x)$  is a unbiased random variable taking values in  $\{1, -1\}$ . For each  $\alpha$  in the set  $\mathcal{A}$  in Proposition 19, the value of  $\mathbb{E}_{m \sim M} (\mathbb{E}_{x \sim D_m} \chi_\alpha(x))^2$  is bounded away from 0. This suggests that for a typical message  $m$  drawn from  $M$ , the distribution of  $\chi_\alpha(x)$  conditioned on the event  $M = m$  is significantly biased. Fact 16 enables us to conclude that Alice's message reveals  $\Omega(1)$  bit of information about  $\chi_\alpha(x)$ . However, since the total information content of Alice's message is at most  $c_\Pi$ , there can be at most  $O(c_\Pi)$  independent vectors in  $\mathcal{A}$ . Now we formalize this intuition.

## 8:14 Linear Sketching over $\mathbb{F}_2$

Let  $\mathcal{T} = \{\alpha_1, \dots, \alpha_\ell\}$  be a basis of  $\text{span}(\mathcal{A})$ . Then,

$$\begin{aligned}
 c_{\Pi} &\geq H(M) && \text{(by the third inequality of Fact 45 (1))} \\
 &\geq I(M; \chi_{\alpha_1}(x), \dots, \chi_{\alpha_\ell}(x)) && \text{(by observation 47)} \\
 &= H(\chi_{\alpha_1}(x), \dots, \chi_{\alpha_\ell}(x)) - H(\chi_{\alpha_1}(x), \dots, \chi_{\alpha_\ell}(x) \mid M) \\
 &= \ell - H(\chi_{\alpha_1}(x), \dots, \chi_{\alpha_\ell}(x) \mid M) \\
 &\quad \text{(by Fact 45 (3) as } \chi_{\alpha_i}(x)\text{'s are independent as random variables)} \\
 &\geq \ell - \sum_{i=1}^{\ell} H(\chi_{\alpha_i}(x) \mid M) && \text{(by Fact 45 (2))} \\
 &\geq \ell - \ell \left(1 - \frac{1}{2} \cdot \frac{1}{3}\right) && \text{(by Fact 16)} \\
 &= \frac{\ell}{6}.
 \end{aligned}$$

Recall that  $c_{\Pi} = \mathcal{D}_{\frac{1-\xi}{6}}^{\rightarrow, U}(f^+)$ . Part 3 of Theorem 14 follows easily from Lemma 19:

$$\begin{aligned}
 \mathcal{D}_{\frac{1-\xi}{6}}^{\rightarrow, U}(f^+) &= c_{\Pi} \\
 &\geq \frac{\ell}{6} && \text{(by Lemma 19 (b))} \\
 &\geq \frac{d}{6}. && \text{(by Lemma 19 (a))}
 \end{aligned}$$

The proof of Theorem 4 now follows directly from Part 1 and Part 3 of Theorem 14 by setting  $\xi = 1/3$ .

## 4 Applications

In this section using Theorem 14 we confirm Conjecture 3 for several function classes: low-degree  $\mathbb{F}_2$  polynomials, functions with sparse Fourier spectrum and symmetric functions (which are not too imbalanced). We also give an example of a composition theorem using recursive majority function as an example.

### 4.1 Low-degree $\mathbb{F}_2$ polynomials

In this section we show that for Boolean functions with low  $\mathbb{F}_2$ -degree randomness does not help in the design of linear sketches or one-way communication protocols. We briefly review some basic definitions, facts and results below.

► **Fact 20.** *For every Boolean function  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  there is a unique  $n$ -variate polynomial  $p \in \mathbb{F}_2[x_1, \dots, x_n]$  such that for every  $(x_1, \dots, x_n) \in \mathbb{F}_2^n$ ,  $f(x_1, \dots, x_n) = p(x_1, \dots, x_n)$ .*

The uniqueness of this representation in particular implies that the only  $\mathbb{F}_2$  polynomial representing the constant 0 function is the polynomial 0. Taking the contrapositive, we have that for every non-constant  $\mathbb{F}_2$  polynomial there is an assignment to its input variables on which the polynomial evaluates to 1.

The degree of  $p$  is referred to as the  $\mathbb{F}_2$ -degree of  $f$ . We will need the following standard result which states that a function with low  $\mathbb{F}_2$ -degree cannot vanish on too many points in its domain. For the sake of completion, we add a proof of it in the appendix (Section E.2).

► **Lemma 21.** *Let  $f$  be a Boolean function different than the constant 0 function with  $\mathbb{F}_2$  degree  $d$ . Then,*

$$\Pr_{x \sim U(\mathbb{F}_2^n)} [f(x) = 1] \geq \frac{1}{2^d}.$$

In this section we prove the following theorem.

► **Theorem 22.** *Let  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  be a Boolean function, and let the  $\mathbb{F}_2$ -degree of  $f$  be  $d$ . Then,*

$$D^{\text{lin}}(f) = \dim(f) = O\left(R_{1/3}^{\rightarrow}(f^+) \cdot d\right).$$

**Proof.** Let  $\ell = \mathcal{D}_{\frac{1}{4 \cdot 2^d}}^{\text{lin}, U}(f)$ . This implies that there is a set  $\mathcal{P} = \{P_1, \dots, P_\ell\}$  of at most  $\ell$  parities and a Boolean function  $g$  such that  $\Pr_{x \sim U(\mathbb{F}_2^n)} [f(x) \neq g(P_1(x), \dots, P_\ell(x))] \leq \frac{1}{4 \cdot 2^d}$ . We now prove that  $D^{\text{lin}}(f)$  (or equivalently Fourier dimension) of  $f$  is at most  $\ell$ . That will prove the theorem as:

$$\begin{aligned} \mathcal{D}_{\frac{1}{4 \cdot 2^d}}^{\text{lin}, U}(f) &= O\left(\mathcal{D}_{\frac{1}{12 \cdot 2^d}}^{\rightarrow, U}(f^+)\right), \\ \mathcal{D}_{\frac{1}{12 \cdot 2^d}}^{\rightarrow, U}(f^+) &= O\left(R_{\frac{1}{12 \cdot 2^d}}^{\rightarrow}(f^+)\right), \\ R_{\frac{1}{12 \cdot 2^d}}^{\rightarrow}(f^+) &= O\left(R_{1/3}^{\rightarrow}(f^+) \cdot d\right). \end{aligned}$$

where the first relation follows by invoking parts 1 and 3 of Theorem 14 with  $\xi = 1 - \frac{1}{2^{d+1}}$ , the second relation holds by fixing the randomness of a randomized one-way protocol appropriately, and the third relation is true because the error of a randomized one-way protocol can be reduced from  $1/3$  to  $\frac{1}{12 \cdot 2^d}$  by taking the majority of  $O(d)$  independent parallel repetitions.

It is left to prove that  $D^{\text{lin}}(f) \leq \ell$ . We prove it by showing that evaluations of all the parities in the set  $\mathcal{P}$  determine the value of  $f$ . For each  $b = (b_1, \dots, b_\ell) \in \mathbb{F}_2^\ell$ , let  $V_b$  denote the affine subspace  $\{x \in \mathbb{F}_2^n : P_1(x) = b_1, \dots, P_\ell(x) = b_\ell\}$  and define:

$$p_b := \Pr_{x \sim U(V_b)} [f(x) \neq g(P_1(x), \dots, P_\ell(x))] = \Pr_{x \sim U(V_b)} [f(x) \neq g(b_1, \dots, b_\ell)].$$

Note that:

$$p_b \geq \min\left\{\Pr_{x \sim U(V_b)} [f(x) = 0], \Pr_{x \sim U(V_b)} [f(x) = 1]\right\} \geq \frac{1}{2} \Pr_{x, x' \sim U(V_b)} [f(x) \neq f(x')]. \quad (5)$$

Given this observation, define  $F : \mathbb{F}_2^n \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  as follows. For  $x, x' \in \mathbb{F}_2^n$  let:

$$F(x, x') := \mathbf{1}_{f(x) \neq f(x')} = f(x) + f(x') \pmod{2}.$$

Note that  $\mathbb{F}_2$ -degree of  $F$  is at most  $d$ . Now,

$$\begin{aligned} &\Pr_{x \sim U(\mathbb{F}_2^n)} [f(x) \neq g(P_1(x), \dots, P_\ell(x))] \leq \frac{1}{4 \cdot 2^d} \\ \Rightarrow &\mathbb{E}_{b \sim U(\mathbb{F}_2^\ell)} \left[ \Pr_{x \sim U(V_b)} [f(x) \neq g(b_1, \dots, b_\ell)] \right] \leq \frac{1}{4 \cdot 2^d} \\ \Rightarrow &\mathbb{E}_{b \sim U(\mathbb{F}_2^\ell)} [p_b] \leq \frac{1}{4 \cdot 2^d} \\ \Rightarrow &\mathbb{E}_{b \sim U(\mathbb{F}_2^\ell)} \left[ \Pr_{x, x' \sim U(V_b)} [f(x) \neq f(x')] \right] \leq \frac{1}{2 \cdot 2^d} \quad (\text{From equation (5)}) \\ \Rightarrow &\mathbb{E}_{b \sim U(\mathbb{F}_2^\ell)} \left[ \Pr_{x, x' \sim U(V_b)} [F(x, x') = 1] \right] \leq \frac{1}{2 \cdot 2^d} \quad (6) \end{aligned}$$

## 8:16 Linear Sketching over $\mathbb{F}_2$

Let  $V$  denote the subspace  $\{(x, x') \in \mathbb{F}_2^n \times \mathbb{F}_2^n : P_1(x) = P_1(x'), \dots, P_\ell(x) = P_\ell(x')\}$  of  $\mathbb{F}_2^n \times \mathbb{F}_2^n$ . From 6 we have that

$$\Pr_{(x, x') \sim U(V)} [F(x, x') = 1] \leq \frac{1}{2 \cdot 2^d} < \frac{1}{2^d}. \quad (7)$$

Since  $\mathbb{F}_2$ -degree of  $F$  is at most  $d$ , restriction of  $F$  to  $V$  also has  $\mathbb{F}_2$  degree at most  $d$ . Equation 7 and Fact 21 imply that  $F$  is the constant 0 function on  $V$ . Thus for each  $x, x'$  such that  $P_1(x) = P_1(x'), \dots, P_\ell(x) = P_\ell(x')$ ,  $f(x) = f(x')$ . Thus  $f(x)$  is a function of  $P_1(x), \dots, P_\ell(x)$ . Hence, Fourier dimension of  $f$  is at most  $\ell$ .  $\blacktriangleleft$

For low-degree polynomials with bounded spectral norm we obtain the following corollary.

► **Corollary 23.** *Let  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  be a Boolean function of  $\mathbb{F}_2$ -degree  $d$ . Then*

$$D^{lin}(f) = \dim(f) = O\left(d \cdot \|\hat{f}\|_1^2\right).$$

**Proof.** The proof follows from the result of Grolmusz [17, 38] that shows that  $R_{1/3}^{\rightarrow}(f^+) = O(\|\hat{f}\|_1^2)$  and Theorem 22.  $\blacktriangleleft$

This result should be compared with Corollary 6 in Tsang et al. [49] who show that  $D^{lin}(f) = O(2^{d^3/2} \log^{d^2} \|\hat{f}\|_1)$ . Corollary 23 gives a stronger bound for  $d = \omega\left(\log^{1/3} \|\hat{f}\|_1\right)$ .

### 4.2 Address function and Fourier sparsity

Consider the *addressing function*  $Add_n : \{0, 1\}^{\log n + n} \rightarrow \{0, 1\}$  defined as follows<sup>14</sup>:

$$Add_n(x, y_1, \dots, y_n) = y_x, \text{ where } x \in \{0, 1\}^{\log n}, y_i \in \{0, 1\},$$

i.e. the value of  $Add_n$  on an input  $(x, y)$  is given by the  $x$ -th bit of the vector  $y$  where  $x$  is treated as a binary representation of an integer number in between 1 and  $n$ . Here  $x$  is commonly referred to as the *address block* and  $y$  as the *addressee block*. Addressing function has only  $n^2$  non-zero Fourier coefficients. In fact, as shown by Sanyal [44] the Fourier dimension, and hence by Fact 48 also the deterministic sketch complexity, of any Boolean function with Fourier sparsity  $s$  is  $O(\sqrt{s} \log s)$ .

Below using the addressing function we show that this relationship is tight (up to a logarithmic factor) even if randomization is allowed, i.e. even for a function with Fourier sparsity  $s$  an  $\mathbb{F}_2$  sketch of size  $\Omega(\sqrt{s})$  might be required.

► **Theorem 24.** *For the addressing function  $Add_n$  and values  $1 \leq d \leq n$  and  $\xi > d/n$  it holds that:*

$$\mathcal{D}_{\frac{1-\sqrt{\xi}}{2}}^{lin, U}(Add_n^+) > d, \quad \mathcal{D}_{\frac{1-\xi}{6}}^{\rightarrow, U}(Add_n) > \frac{d}{6}.$$

**Proof.** If we apply the standard Fourier notation switch where we replace 0 with 1 and 1 with  $-1$  in the domain and the range of the function then the addressing function  $Add_n(x, y)$  can be expressed as the following multilinear polynomial:

$$Add_n(x, y) = \sum_{i \in \{0, 1\}^{\log n}} y_i \prod_{j: i_j=1} \left(\frac{1-x_j}{2}\right) \prod_{j: i_j=0} \left(\frac{1+x_j}{2}\right),$$

<sup>14</sup>In this section it will be more convenient to represent both domain and range of the function using  $\{0, 1\}$  rather than  $\mathbb{F}_2$ .



which makes it clear that the only non-zero Fourier coefficients correspond to the sets that contain a single variable from the addressee block and an arbitrary subset of variables from the address block. This expansion also shows that the absolute value of each Fourier coefficient is equal to  $\frac{1}{n}$ .

Fix any  $d$ -dimensional subspace  $\mathcal{A}_d$  and consider the matrix  $M \in \mathbb{F}_2^{d \times (\log n + n)}$  composed of the basis vectors as rows. We add to  $M$  extra  $\log n$  rows which contain an identity matrix in the first  $\log n$  coordinates and zeros everywhere else. This gives us a new matrix  $M' \in \mathbb{F}_2^{(d + \log n) \times (\log n + n)}$ . Applying Gaussian elimination to  $M'$  we can assume that it is of the following form:

$$M' = \begin{pmatrix} I_{\log n} & 0 & 0 \\ 0 & I_{d'} & M'' \\ 0 & 0 & 0 \end{pmatrix},$$

where  $d' \leq d$ . Thus, the total number of non-zero Fourier coefficients spanned by the rows of  $M'$  equals  $nd'$ . Hence, the total sum of squared Fourier coefficients in  $\mathcal{A}_d$  is at most  $\frac{d'}{n} \leq \frac{d}{n}$ , i.e.  $\dim_{\mathcal{E}}(\text{Add}_n) > d$ . By Part 2 and Part 3 of Theorem 14 the statement of the theorem follows.  $\blacktriangleleft$

### 4.3 Symmetric functions

A function  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  is symmetric if it can be expressed as  $g(\|x\|_0)$  for some function  $g : [0, n] \rightarrow \mathbb{F}_2$ . We give the following lower bound for symmetric functions:

► **Theorem 25** (Lower bound for symmetric functions). *For any symmetric function  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  that isn't  $(1 - \epsilon)$ -concentrated on  $\{\emptyset, \{1, \dots, n\}\}$ :*

$$\mathcal{D}_{\epsilon/8}^{\text{lin}, U}(f) \geq \frac{n}{2e}, \quad \mathcal{D}_{\epsilon/12}^{\rightarrow, U}(f^+) \geq \frac{n}{2e}.$$

**Proof.** First we prove an auxiliary lemma. Let  $W_k$  be the set of all vectors in  $\mathbb{F}_2^n$  of Hamming weight  $k$ .

► **Lemma 26.** *For any  $d \in [n/2]$ ,  $k \in [n - 1]$  and any  $d$ -dimensional subspace  $\mathcal{A}_d \leq \mathbb{F}_2^n$ :*

$$\frac{|W_k \cap \mathcal{A}_d|}{|W_k|} \leq \binom{ed}{n}^{\min(k, n-k, d)} \leq \frac{ed}{n}.$$

**Proof.** Fix any basis in  $\mathcal{A}_d$  and consider the matrix  $M \in \mathbb{F}_2^{d \times n}$  composed of the basis vectors as rows. W.l.o.g we can assume that this matrix is diagonalized and is in the standard form  $(I_d, M')$  where  $I_d$  is a  $d \times d$  identity matrix and  $M'$  is a  $d \times (n - d)$ -matrix. Clearly, any linear combination of more than  $k$  rows of  $M$  has Hamming weight greater than  $k$  just from the contribution of the first  $d$  coordinates. Thus, we have  $|W_k \cap \mathcal{A}_d| \leq \sum_{i=0}^k \binom{d}{i}$ .

For any  $k \leq d$  it is a standard fact about binomials that  $\sum_{i=0}^k \binom{d}{i} \leq \left(\frac{ed}{k}\right)^k$ . On the other hand, we have  $|W_k| = \binom{n}{k} \geq (n/k)^k$ . Thus, we have  $\frac{|W_k \cap \mathcal{A}_d|}{|W_k|} \leq \left(\frac{ed}{n}\right)^k$  and hence for  $1 \leq k \leq d$  the desired inequality holds.

If  $d < k$  then consider two cases. Since  $d \leq n/2$  the case  $n - d \leq k \leq n - 1$  is symmetric to  $1 \leq k \leq d$ . If  $d < k < n - d$  then we have  $|W_k| > |W_d| \geq (n/d)^d$  and  $|W_k \cap \mathcal{A}_d| \leq 2^d$  so that the desired inequality follows.  $\blacktriangleleft$

Any symmetric function has its spectrum distributed uniformly over Fourier coefficients of any fixed weight. Let  $w_i = \sum_{S \in W_i} \hat{f}^2(S)$ . By the assumption of the theorem we have

## 8:18 Linear Sketching over $\mathbb{F}_2$

$\sum_{i=1}^{n-1} w_i \geq \epsilon$ . Thus, by Lemma 26 any linear subspace  $\mathcal{A}_d$  of dimension at most  $d \leq n/2$  satisfies that:

$$\begin{aligned} \sum_{S \in \mathcal{A}_d} f^2(S) &\leq \hat{f}^2(\emptyset) + \hat{f}^2(\{1, \dots, n\}) + \sum_{i=1}^{n-1} w_i \frac{|W_i \cap \mathcal{A}_d|}{|W_i|} \\ &\leq \hat{f}^2(\emptyset) + \hat{f}^2(\{1, \dots, n\}) + \sum_{i=1}^{n-1} w_i \frac{ed}{n} \\ &\leq (1 - \epsilon) + \epsilon \frac{ed}{n}. \end{aligned}$$

Thus,  $f$  isn't  $1 - \epsilon(1 - \frac{ed}{n})$ -concentrated on any  $d$ -dimensional linear subspace, i.e.  $\dim_\xi(f) > d$  for  $\xi = 1 - \epsilon(1 - \frac{ed}{n})$ . By Part 2 of Theorem 14 this implies that  $f$  doesn't have randomized sketches of dimension at most  $d$  which err with probability less than:

$$\frac{1}{2} - \frac{\sqrt{1 - \epsilon(1 - \frac{ed}{n})}}{2} \geq \frac{\epsilon}{4} \left(1 - \frac{ed}{n}\right) \geq \frac{\epsilon}{8}$$

where the last inequality follows by the assumption that  $d \leq \frac{n}{2e}$ . The communication complexity lower bound follows by Part 3 of Theorem 14 by setting  $d = \frac{n}{2e}$ .  $\blacktriangleleft$

### 4.4 Composition theorem for majority

In this section using Theorem 14 we give a composition theorem for  $\mathbb{F}_2$ -sketching of the composed  $Maj_3$  function. Unlike in the deterministic case for which the composition theorem is easy to show (see Lemma 53) in the randomized case composition results require more work.

► **Definition 27** (Composition). For  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  and  $g: \mathbb{F}_2^m \rightarrow \mathbb{F}_2$  their composition  $f \circ g: \mathbb{F}_2^{mn} \rightarrow \mathbb{F}_2$  is defined as:

$$(f \circ g)(x) = f(g(x_1, \dots, x_m), g(x_{m+1}, \dots, x_{2m}), \dots, g(x_{m(n-1)+1}, \dots, x_{mn})).$$

Consider the recursive majority function  $Maj_3^{\circ k} \equiv Maj_3 \circ Maj_3 \circ \dots \circ Maj_3$  where the composition is taken  $k$  times.

► **Theorem 28.** For any  $d \leq n$ ,  $k = \log_3 n$  and  $\xi > \frac{4d}{n}$  it holds that  $\dim_\xi(Maj_3^{\circ k}) > d$ .

First, we show a slightly stronger result for standard subspaces and then extend this result to arbitrary subspaces with a loss of a constant factor. Fix any set  $S \subseteq [n]$  of variables. We associate this set with a collection of standard unit vectors corresponding to these variables. Hence in this notation  $\emptyset$  corresponds to the all-zero vector.

► **Lemma 29.** For any standard subspace whose basis consists of singletons from the set  $S \subseteq [n]$  it holds that:

$$\sum_{Z \in \text{span}(S)} \left(\widehat{Maj_3^{\circ k}}(Z)\right)^2 \leq \frac{|S|}{n}$$

**Proof.** The Fourier expansion of  $Maj_3$  is given as

$$Maj_3(x_1, x_2, x_3) = \frac{1}{2}(x_1 + x_2 + x_3 - x_1x_2x_3).$$

For  $i \in \{1, 2, 3\}$  let  $N_i = \{(i-1)n/3 + 1, \dots, in/3\}$ . Let  $S_i = S \cap N_i$ . Let  $\alpha_i$  be defined as:

$$\alpha_i = \sum_{Z \in \text{span}(S_i)} \left( \widehat{Maj}_3^{\circ k-1}(Z) \right)^2.$$

Then we have:

$$\begin{aligned} \sum_{Z \in \text{span}(S)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2 &= \sum_{i=1}^3 \sum_{Z \in \text{span}(S_i)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2 + \\ &\quad \sum_{Z \in \text{span}(S) - \cup_{i=1}^3 \text{span}(S_i)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2. \end{aligned}$$

For each  $S_i$  we have

$$\sum_{Z \in \text{span}(S_i)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2 = \frac{1}{4} \sum_{Z \in \text{span}(S_i)} \left( \widehat{Maj}_3^{\circ k-1}(Z) \right)^2 = \frac{\alpha_i}{4}.$$

Moreover, for each  $Z \in \text{span}(S) - \cup_{i=1}^3 \text{span}(S_i)$  we have:

$$\widehat{Maj}_3^{\circ k}(Z) = \begin{cases} -\frac{1}{2} \widehat{Maj}_3^{\circ k-1}(Z_1) \widehat{Maj}_3^{\circ k-1}(Z_2) \widehat{Maj}_3^{\circ k-1}(Z_3) & \text{if } Z \in \times_{i=1}^3 (\text{span}(S_i) \setminus \emptyset) \\ 0 & \text{otherwise.} \end{cases}$$

Thus, we have:

$$\begin{aligned} &\sum_{Z \in (\text{span}(S_1) \setminus \emptyset) \times (\text{span}(S_2) \setminus \emptyset) \times (\text{span}(S_3) \setminus \emptyset)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2 \\ &= \sum_{Z \in (\text{span}(S_1) \setminus \emptyset) \times (\text{span}(S_2) \setminus \emptyset) \times (\text{span}(S_3) \setminus \emptyset)} \frac{1}{4} \left( \widehat{Maj}_3^{\circ k-1}(Z_1) \right)^2 \cdot \left( \widehat{Maj}_3^{\circ k-1}(Z_2) \right)^2 \cdot \left( \widehat{Maj}_3^{\circ k-1}(Z_3) \right)^2 \\ &= \frac{1}{4} \left( \sum_{Z \in (\text{span}(S_1) \setminus \emptyset)} \left( \widehat{Maj}_3^{\circ k-1}(Z_1) \right)^2 \right) \cdot \left( \sum_{Z \in (\text{span}(S_2) \setminus \emptyset)} \left( \widehat{Maj}_3^{\circ k-1}(Z_2) \right)^2 \right) \cdot \left( \sum_{Z \in (\text{span}(S_3) \setminus \emptyset)} \left( \widehat{Maj}_3^{\circ k-1}(Z_3) \right)^2 \right) \\ &= \frac{1}{4} \alpha_1 \alpha_2 \alpha_3. \end{aligned}$$

where the last equality holds since  $\widehat{Maj}_3^{\circ k-1}(\emptyset) = 0$ . Putting this together we have:

$$\begin{aligned} \sum_{Z \in \text{span}(S)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2 &= \frac{1}{4} (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_1 \alpha_2 \alpha_3) \\ &\leq \frac{1}{4} \left( \alpha_1 + \alpha_2 + \alpha_3 + \frac{1}{3} (\alpha_1 + \alpha_2 + \alpha_3) \right) = \frac{1}{3} (\alpha_1 + \alpha_2 + \alpha_3). \end{aligned}$$

Applying this argument recursively to each  $\alpha_i$  for  $k-1$  times we have:

$$\sum_{Z \in \text{span}(S)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2 \leq \frac{1}{3^k} \sum_{i=1}^3 \gamma_i,$$

where  $\gamma_i = 1$  if  $i \in S$  and 0 otherwise. Thus,  $\sum_{Z \in \text{span}(S)} \left( \widehat{Maj}_3^{\circ k}(Z) \right)^2 \leq \frac{|S|}{n}$ . ◀

To extend the argument to arbitrary linear subspaces we show that any such subspace has less Fourier weight than a collection of three carefully chosen standard subspaces. First we show how to construct such subspaces in Lemma 30.

For a linear subspace  $L \leq \mathbb{F}_2^n$  we denote the set of all vectors in  $L$  of odd Hamming weight as  $\mathcal{O}(L)$  and refer to it as the *odd set* of  $L$ . For two vectors  $v_1, v_2 \in \mathbb{F}_2^n$  we say that  $v_1$  *dominates*  $v_2$  if the set of non-zero coordinates of  $v_1$  is a (not necessarily proper) subset of the set of non-zero coordinates of  $v_2$ . For two sets of vectors  $S_1, S_2 \subseteq \mathbb{F}_2^n$  we say that  $S_1$  *dominates*  $S_2$  (denoted as  $S_1 \prec S_2$ ) if there is a matching  $M$  between  $S_1$  and  $S_2$  of size  $|S_2|$  such that for each  $(v_1 \in S_1, v_2 \in S_2) \in M$  the vector  $v_1$  dominates  $v_2$ .

► **Lemma 30** (Standard subspace domination lemma). *For any linear subspace  $L \leq \mathbb{F}_2^n$  of dimension  $d$  there exist three standard linear subspaces  $S_1, S_2, S_3 \leq \mathbb{F}_2^n$  such that:*

$$\mathcal{O}(L) \prec \mathcal{O}(S_1) \cup \mathcal{O}(S_2) \cup \mathcal{O}(S_3),$$

and  $\dim(S_1) = d - 1, \dim(S_2) = d, \dim(S_3) = 2d$ .

**Proof.** Let  $A \in \mathbb{F}_2^{d \times n}$  be the matrix with rows corresponding to the basis in  $L$ . We will assume that  $A$  is normalized in a way described below. First, we apply Gaussian elimination to ensure that  $A = (I, M)$  where  $I$  is a  $d \times d$  identity matrix. If all rows of  $A$  have even Hamming weight then the lemma holds trivially since  $\mathcal{O}(L) = \emptyset$ . By reordering rows and columns of  $A$  we can always assume that for some  $k \geq 1$  the first  $k$  rows of  $A$  have odd Hamming weight and the last  $d - k$  have even Hamming weight. Finally, we add the first column to each of the last  $d - k$  rows, which makes all rows have odd Hamming weight. This results in  $A$  of the following form:

$$A = \left( \begin{array}{c|cc|c} 1 & 0 \cdots 0 & 0 \cdots 0 & a \\ 0 & \vdots & 0 & \vdots \\ \vdots & I_{k-1} & 0 & M_1 \\ 0 & \vdots & \vdots & \vdots \\ \hline 1 & 0 & I_{d-k} & M_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \vdots & \vdots & \vdots \end{array} \right)$$

We use the following notation for submatrices:  $A[i_1, j_1; i_2, j_2]$  refers to the submatrix of  $A$  with rows between  $i_1$  and  $j_1$  and columns between  $i_2$  and  $j_2$  inclusive. We denote to the first row by  $v$ , the submatrix  $A[2, k; 1, n]$  as  $\mathcal{A}$  and the submatrix  $A[k + 1, d; 1, n]$  as  $\mathcal{B}$ . Each  $x \in \mathcal{O}(L)$  can be represented as  $\sum_{i \in S} A_i$  where the set  $S$  is of odd size and the sum is over  $\mathbb{F}_2^n$ . We consider the following three cases corresponding to different types of the set  $S$ .

**Case 1.**  $S \subseteq \text{rows}(\mathcal{A}) \cup \text{rows}(\mathcal{B})$ . This corresponds to all odd size linear combinations of the rows of  $A$  that don't include the first row. Clearly, the set of such vectors is dominated by  $\mathcal{O}(S_1)$  where  $S_1$  is the standard subspace corresponding to the span of the rows of the submatrix  $A[2, d; 2, d]$ .

**Case 2.**  $S$  contains the first row,  $|S \cap \text{rows}(\mathcal{A})|$  and  $|S \cap \text{rows}(\mathcal{B})|$  are even. All such linear combinations have their first coordinate equal 1. Hence, they are dominated by a standard subspace corresponding to span of the rows the  $d \times d$  identity matrix, which we refer to as  $S_2$ .

**Case 3.**  $S$  contains the first row,  $|S \cap \text{rows}(\mathcal{A})|$  and  $|S \cap \text{rows}(\mathcal{B})|$  are odd. All such linear combinations have their first coordinate equal 0. This implies that the Hamming weight of the first  $d$  coordinates of such linear combinations is even and hence the other coordinates cannot be all equal to 0. Consider the submatrix  $M = A[1, d; d + 1, n]$  corresponding to the last  $n - d$  columns of  $A$ . Since the rank of this matrix is at most  $d$  by running Gaussian elimination on  $M$  we can construct a matrix  $M'$  containing as rows the basis for the row space of  $M$  of the following form:

$$M' = \begin{pmatrix} I_t & M_1 \\ 0 & 0 \end{pmatrix}$$

where  $t = \text{rank}(M)$ . This implies that any non-trivial linear combination of the rows of  $M$  contains 1 in one of the first  $t$  coordinates. We can reorder the columns of  $A$  in such a way that these  $t$  coordinates have indices from  $d + 1$  to  $d + t$ . Note that now the set of vectors spanned by the rows of the  $(d + t) \times (d + t)$  identity matrix  $I_{d+t}$  dominates the set of linear combinations we are interested in. Indeed, each such linear combination has even Hamming weight in the first  $d$  coordinates and has at least one coordinate equal to 1 in the set  $\{d + 1, \dots, d + t\}$ . This gives a vector of odd Hamming weight that dominates such linear combination. Since this mapping is injective we have a matching. We denote the standard linear subspace constructed this way by  $S_3$  and clearly  $\dim(S_3) \leq 2d$ . ◀

The following proposition shows that the spectrum of the  $\text{Maj}_3^{\circ k}$  is monotone decreasing under inclusion if restricted to odd size sets only:

▶ **Proposition 31.** *For any two sets  $Z_1 \subseteq Z_2$  of odd size it holds that:*

$$\left| \widehat{\text{Maj}_3^{\circ k}}(Z_1) \right| \geq \left| \widehat{\text{Maj}_3^{\circ k}}(Z_2) \right|.$$

**Proof.** The proof is by induction on  $k$ . Consider the Fourier expansion of  $\text{Maj}_3(x_1, x_2, x_3) = \frac{1}{2}(x_1 + x_2 + x_3 - x_1x_2x_3)$ . The case  $k = 1$  holds since all Fourier coefficients have absolute value  $1/2$ . Since  $\text{Maj}_3^{\circ k} = \text{Maj}_3 \circ (\text{Maj}_3^{\circ k-1})$  all Fourier coefficients of  $\text{Maj}_3^{\circ k}$  result from substituting either a linear or a cubic term in the Fourier expansion by the multilinear expansions of  $\text{Maj}_3^{\circ k-1}$ . This leads to four cases.

**Case 1.**  $Z_1$  and  $Z_2$  both arise from linear terms. In this case if  $Z_1$  and  $Z_2$  aren't disjoint then they arise from the same linear term and thus satisfy the statement by the inductive hypothesis.

**Case 2.** If  $Z_1$  arises from a cubic term and  $Z_2$  from the linear term then it can't be the case that  $Z_1 \subseteq Z_2$  since  $Z_2$  contains some variables not present in  $Z_1$ .

**Case 3.** If  $Z_1$  and  $Z_2$  both arise from the cubic term then we have  $(Z_1 \cap N_i) \subseteq (Z_2 \cap N_i)$  for each  $i$ . By the inductive hypothesis we then have  $\left| \widehat{\text{Maj}_3^{\circ k-1}}(Z_1 \cap N_i) \right| \geq \left| \widehat{\text{Maj}_3^{\circ k-1}}(Z_2 \cap N_i) \right|$ .

Since for  $j = 1, 2$  we have  $\widehat{\text{Maj}_3^{\circ k}}(Z_j) = -\frac{1}{2} \prod_i \widehat{\text{Maj}_3^{\circ k-1}}(Z_j \cap N_i)$  the desired inequality follows.

**Case 4.** If  $Z_1$  arises from the linear term and  $Z_2$  from the cubic term then w.l.o.g. assume that  $Z_1$  arises from the  $x_1$  term. Note that  $Z_1 \subseteq (Z_2 \cap N_1)$  since  $Z_1 \cap (N_2 \cup N_3) = \emptyset$ . By the inductive hypothesis applied to  $Z_1$  and  $Z_2 \cap N_1$  the desired inequality holds. ◀

We can now complete the proof of Theorem 28

**Proof of Theorem 28.** By combining Proposition 31 and Lemma 29 we have that any set  $\mathcal{T}$  of vectors that is dominated by  $\mathcal{O}(\mathcal{S})$  for some standard subspace  $\mathcal{S}$  satisfies  $\sum_{S \in \mathcal{T}} \widehat{Maj_3^{\circ k}}(S)^2 \leq \frac{\dim(\mathcal{S})}{n}$ . By the standard subspace domination lemma (Lemma 30) any subspace  $L \leq \mathbb{F}_2^n$  of dimension  $d$  has  $\mathcal{O}(L)$  dominated by a union of three standard subspaces of dimension  $2d$ ,  $d$  and  $d - 1$  respectively. Thus, we have  $\sum_{S \in \mathcal{O}(L)} \widehat{Maj_3^{\circ k}}(S)^2 \leq \frac{2d}{n} + \frac{d}{n} + \frac{d-1}{n} \leq \frac{4d}{n}$ . ◀

We have the following corollary of Theorem 28 that proves Theorem 5.

► **Corollary 32.** For any  $\epsilon \in [0, \frac{1}{2}]$ ,  $\xi > 4\epsilon^2$  and  $k = \log_3 n$  it holds that:

$$\mathcal{D}_{\frac{1-\sqrt{\xi}}{2}}^{\text{lin}, U}(Maj_3^{\circ k}) > \epsilon^2 n, \quad \mathcal{D}_{\frac{1-\xi}{6}}^{\rightarrow, U}(Maj_3^{\circ k+}) > \frac{\epsilon^2 n}{6}.$$

**Proof.** Fix  $d = \epsilon^2 n$ . For this choice of  $d$  Theorem 28 implies that for  $\xi > 4\epsilon^2$  it holds that  $\dim_{\xi}(Maj_3^{\circ k}) > d$ . The first part follows from Part 2 of Theorem 14. The second part is by Part 3 of Theorem 14. ◀

## 5 Streaming algorithms over $\mathbb{F}_2$

Let  $e_i$  be the standard unit vector in  $\mathbb{F}_2^n$ . In the turnstile streaming model the input  $x \in \mathbb{F}_2^n$  is represented as a stream  $\sigma = (\sigma_1, \sigma_2, \dots)$  where  $\sigma_i \in \{e_1, \dots, e_n\}$ . For a stream  $\sigma$  the resulting vector  $x$  corresponds to its frequency vector  $\text{freq } \sigma \equiv \sum_i \sigma_i$ . Concatenation of two streams  $\sigma$  and  $\tau$  is denoted as  $\sigma \circ \tau$ .

### 5.1 Random streams

In this section we show how to translate our results in Section 3 and 4 into lower bounds for streaming algorithms. We consider the following two natural models of random streams over  $\mathbb{F}_2$ :

**Model 1.** In the first model we start with  $x \in \mathbb{F}_2^n$  that is drawn from the uniform distribution over  $\mathbb{F}_2^n$  and then apply a uniformly random update  $y \sim U(\mathbb{F}_2^n)$  obtaining  $x + y$ . In the streaming language this corresponds to a stream  $\sigma = \sigma_1 \circ \sigma_2$  where  $\text{freq } \sigma_1 \sim U(\mathbb{F}_2^n)$  and  $\text{freq } \sigma_2 \sim U(\mathbb{F}_2^n)$ . A specific example of such stream would be one where for both  $\sigma_1$  and  $\sigma_2$  we flip an unbiased coin to decide whether or not to include a vector  $e_i$  in the stream for each value of  $i$ . The expected length of the stream in this case is  $n$ .

**Model 2.** In the second model we consider a stream  $\sigma$  which consists of uniformly random updates. Let  $\sigma_i = e_{r(i)}$  where  $r(i) \sim U([n])$ . This corresponds to each update being a flip in a coordinate of  $x$  chosen uniformly at random. This model is equivalent to the previous model but requires longer streams to mix. Using coupon collector's argument such streams of length  $\Theta(n \log n)$  can be divided into two substreams  $\sigma_1$  and  $\sigma_2$  such that with high probability both  $\text{freq } \sigma_1$  and  $\text{freq } \sigma_2$  are uniformly distributed over  $\mathbb{F}_2^n$  and  $\sigma = \sigma_1 \circ \sigma_2$ .

► **Theorem 33.** Let  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  be an arbitrary function. In the two random streaming models for generating  $\sigma$  described above any algorithm that computes  $f(\text{freq } \sigma)$  with probability at least  $8/9$  in the end of the stream has to use space that is at least  $\mathcal{D}_{1/3}^{\text{lin}, U}(f)$ .

**Proof.** The proof follows directly from Theorem 4 as in both models we can partition the stream into  $\sigma_1$  and  $\sigma_2$  such that  $\text{freq } \sigma_1$  and  $\text{freq } \sigma_2$  are both distributed uniformly over  $\mathbb{F}_2^n$ . We treat these two frequency vectors as inputs of Alice and Bob in the communication game. Since communication  $\mathcal{D}_{1/9}^{\rightarrow, U}(f^+) \geq \mathcal{D}_{1/3}^{\text{lin}, U}(f)$  is required no streaming algorithm with less space exists as otherwise Alice would transfer its state to Bob with less communication. ◀

Using the same proof as in Theorem 33 it follows that all the lower bounds in Section 4 hold for both random streaming models described above.

## 5.2 Adversarial streams

We now show that any randomized turnstile streaming algorithm for computing  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  with error probability  $\delta$  has to use space that is at least  $R_{6\delta}^{\text{lin}}(f) - O(\log n + \log(1/\delta))$  under adversarial sequences of updates. The proof is based on the recent line of work that shows that this relationship holds for real-valued sketches [10, 31, 1]. The proof framework developed by [10, 31, 1] for real-valued sketches consists of two steps. First, a turnstile streaming algorithm is converted into a path-independent stream automaton (Definition 35). Second, using the theory of modules and their representations it is shown that such automata can always be represented as linear sketches. We observe that the first step of this framework can be left unchanged under  $\mathbb{F}_2$ . However, as we show the second step can be significantly simplified as path-independent automata over  $\mathbb{F}_2$  can be directly seen as linear sketches without using module theory. Furthermore, since we are working over  $\mathbb{F}_2$  we also avoid the  $O(\log m)$  factor loss in the reduction between path independent automata and linear sketches that is present in [10].

We use the following abstraction of a *stream automaton* from [10, 31, 1] adapted to our context to represent general turnstile streaming algorithms over  $\mathbb{F}_2$ .

► **Definition 34** (Deterministic Stream Automaton). A *deterministic stream automaton*  $\mathcal{A}$  is a Turing machine that uses two tapes, an unidirectional read-only input tape and a bidirectional work tape. The input tape contains the input stream  $\sigma$ . After processing the input, the automaton writes an output, denoted as  $\phi_{\mathcal{A}}(\sigma)$ , on the work tape. A configuration (or state) of  $\mathcal{A}$  is determined by the state of its finite control, head position, and contents of the work tape. The computation of  $\mathcal{A}$  can be described by a transition function  $\oplus_{\mathcal{A}} : C \times \mathbb{F}_2 \rightarrow C$ , where  $C$  is the set of all possible configurations. For a configuration  $c \in C$  and a stream  $\sigma$ , we denote by  $c \oplus_{\mathcal{A}} \sigma$  the configuration of  $\mathcal{A}$  after processing  $\sigma$  starting from the initial configuration  $c$ . The set of all configurations of  $\mathcal{A}$  that are reachable via processing some input stream  $\sigma$  is denoted as  $C(\mathcal{A})$ . The space of  $\mathcal{A}$  is defined as  $\mathcal{S}(\mathcal{A}) = \log |C(\mathcal{A})|$ .

We say that a deterministic stream automaton computes a function  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  over a distribution  $\Pi$  if  $\Pr_{\sigma \sim \Pi}[\phi_{\mathcal{A}}(\sigma) = f(\text{freq } \sigma)] \geq 1 - \delta$ .

► **Definition 35** (Path-independent automaton). An automaton  $\mathcal{A}$  is said to be *path-independent* if for any configuration  $c$  and any input stream  $\sigma$ ,  $c \oplus_{\mathcal{A}} \sigma$  depends only on  $\text{freq } \sigma$  and  $c$ .

► **Definition 36** (Randomized Stream Automaton). A *randomized stream automaton*  $\mathcal{A}$  is a deterministic automaton with an additional tape for the random bits. This random tape is initialized with a random bit string  $R$  before the automaton is executed. During the execution of the automaton this bit string is used in a bidirectional read-only manner while the rest of the execution is the same as in the deterministic case. A randomized automaton  $\mathcal{A}$  is said to be path-independent if for each possible fixing of its randomness  $R$  the deterministic automaton  $\mathcal{A}_R$  is path-independent. The space complexity of  $\mathcal{A}$  is defined as  $\mathcal{S}(\mathcal{A}) = \max_R(|R| + \mathcal{S}(\mathcal{A}_R))$ .

Theorems 5 and 9 of [31] combined with the observation in Appendix A of [1] that guarantees path independence yields the following:

► **Theorem 37** (Theorems 5 and 9 in [31] + [1]). *Suppose that a randomized stream automaton  $\mathcal{A}$  computes  $f$  on any stream with probability at least  $1 - \delta$ . For an arbitrary distribution  $\Pi$  over streams there exists a deterministic<sup>15</sup> path independent stream automaton  $\mathcal{B}$  that computes  $f$  with probability  $1 - 6\delta$  over  $\Pi$  such that  $\mathcal{S}(\mathcal{B}) \leq \mathcal{S}(\mathcal{A}) + O(\log n + \log(1/\delta))$ .*

The rest of the argument below is based on the work of Ganguly [10] adopted for our needs. Since we are working over a finite field we also avoid the  $O(\log m)$  factor loss in the reduction between path independent automata and linear sketches that is present in Ganguly’s work.

Let  $A_n$  be a path-independent stream automaton over  $\mathbb{F}_2$  and let  $\oplus$  abbreviate  $\oplus_{A_n}$ . Define the function  $*$  :  $\mathbb{F}_2^n \times C(A_n) \rightarrow C(A_n)$  as:  $x*a = a \oplus \sigma$ , where  $\text{freq}(\sigma) = x$ . Let  $o$  be the initial configuration of  $A_n$ . The kernel  $M_{A_n}$  of  $A_n$  is defined as  $M_{A_n} = \{x \in \mathbb{F}_2^n : x * o = 0^n * o\}$ .

► **Proposition 38.** *The kernel  $M_{A_n}$  of a path-independent automaton  $A_n$  is a linear subspace of  $\mathbb{F}_2^n$ .*

**Proof.** For  $x, y \in M_{A_n}$  by path independence  $(x+y)*o = x*(y*o) = 0^n*o$  so  $x+y \in M_{A_n}$ . ◀

Since  $M_{A_n} \leq \mathbb{F}_2^n$  the kernel partitions  $\mathbb{F}_2^n$  into cosets of the form  $x + M_{A_n}$ . Next we show that there is a one to one mapping between these cosets and the states of  $A_n$ .

► **Proposition 39.** *For  $x, y \in \mathbb{F}_2^n$  and a path independent automaton  $A_n$  with a kernel  $M_{A_n}$  it holds that  $x * o = y * o$  if and only if  $x$  and  $y$  lie in the same coset of  $M_{A_n}$ .*

**Proof.** By path independence  $x * o = y * o$  iff  $x * (x * o) = x * (y * o)$  or equivalently  $0^n * o = (x + y) * o$ . The latter condition holds iff  $x + y \in M_{A_n}$  which is equivalent to  $x$  and  $y$  lying in the same coset of  $M_{A_n}$ . ◀

The same argument implies that the transition function of a path-independent automaton has to be linear since  $(x + y) * o = x * (y * o)$ . Combining these facts together we conclude that a path-independent automaton has at least as many states as the best deterministic  $\mathbb{F}_2$ -sketch for  $f$  that succeeds with probability at least  $1 - 6\delta$  over  $\Pi$  (and hence the best randomized sketch as well). Putting things together we get:

► **Theorem 40.** *Any randomized streaming algorithm that computes  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  under arbitrary updates over  $\mathbb{F}_2$  with error probability at least  $1 - \delta$  has space complexity at least  $R_{6\delta}^{\text{lin}}(f) - O(\log n + \log(1/\delta))$ .*

---

## References

- 1 Yuqing Ai, Wei Hu, Yi Li, and David P. Woodruff. New characterizations in turnstile streams with applications. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 20:1–20:22. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.20.
- 2 Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Trans. Information Theory*, 51(11):4032–4039, 2005.

---

<sup>15</sup>We note that [31] construct  $\mathcal{B}$  as a randomized automaton in their Theorem 9 but it can always be made deterministic by fixing the randomness that achieves the smallest error.



- 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. doi:10.1006/jcss.1997.1545.
- 4 Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1723–1742, 2017.
- 5 Sepehr Assadi, Sanjeev Khanna, Yang Li, and Grigory Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364, 2016.
- 6 Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 488–497, 2010.
- 7 Eric Blais, Li-Yang Tan, and Andrew Wan. An inequality for the fourier spectrum of parity decision trees. *CoRR*, abs/1506.01055, 2015. arXiv:1506.01055.
- 8 Andrej Bogdanov and Emanuele Viola. Pseudorandom bits for polynomials. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 41–51, 2007.
- 9 Anirban Dasgupta, Ravi Kumar, and D. Sivakumar. Sparse and lopsided set disjointness via information theory. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 517–528, 2012.
- 10 Sumit Ganguly. Lower bounds on frequency estimation of data streams (extended abstract). In *Computer Science - Theory and Applications, Third International Computer Science Symposium in Russia, CSR 2008, Moscow, Russia, June 7-12, 2008, Proceedings*, pages 204–215, 2008.
- 11 Dmitry Gavinsky, Julia Kempe, and Ronald de Wolf. Quantum communication cannot simulate a public coin. *CoRR*, quant-ph/0411051, 2004. URL: <http://arxiv.org/abs/quant-ph/0411051>.
- 12 Mohsen Ghaffari and Merav Parter. MST in log-star rounds of congested clique. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 19–28, 2016.
- 13 Mika Göös and T. S. Jayram. A composition theorem for conical juntas. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 5:1–5:16, 2016.
- 14 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 257–266, 2015.
- 15 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1077–1088, 2015.
- 16 Parikshit Gopalan, Ryan O’Donnell, Rocco A. Servedio, Amir Shpilka, and Karl Wimmer. Testing fourier dimensionality and sparsity. *SIAM J. Comput.*, 40(4):1075–1100, 2011. doi:10.1137/100785429.
- 17 Vince Grolmusz. On the power of circuits with gates of low  $l_1$  norms. *Theor. Comput. Sci.*, 188(1-2):117–128, 1997. doi:10.1016/S0304-3975(96)00290-3.

- 18 Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for XOR functions. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 282–288, 2016.
- 19 James W. Hegeman, Gopal Pandurangan, Sriram V. Pemmaraju, Vivek B. Sardeshmukh, and Michele Scquizzato. Toward optimal bounds in the congested clique: Graph connectivity and MST. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 91–100, 2015.
- 20 Wei Huang, Yaoyun Shi, Shengyu Zhang, and Yufan Zhu. The communication complexity of the hamming distance problem. *Inf. Process. Lett.*, 99(4):149–153, 2006. doi:10.1016/j.ipl.2006.01.014.
- 21 T. S. Jayram. Information complexity: a tutorial. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, USA*, pages 159–168, 2010.
- 22 T. S. Jayram, Ravi Kumar, and D. Sivakumar. Two applications of information complexity. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 673–682, 2003.
- 23 Tomasz Jurdzinski and Krzysztof Nowicki. MST in  $O(1)$  rounds of congested clique. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2620–2632, 2018.
- 24 Michael Kapralov, Yin Tat Lee, Cameron Musco, Christopher Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 561–570, 2014.
- 25 Michael Kapralov, Jelani Nelson, Jakub Pachocki, Zhengyu Wang, David P. Woodruff, and Mobin Yahyazadeh. Optimal lower bounds for universal relation, and for samplers and finding duplicates in streams. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 475–486, 2017.
- 26 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993. doi:10.1137/0222080.
- 27 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.
- 28 Troy Lee and Shengyu Zhang. Composition theorems in communication complexity. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, pages 475–489, 2010.
- 29 Nikos Leonardos. An improved lower bound for the randomized decision tree complexity of recursive majority,. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 696–708, 2013.
- 30 Ming Lam Leung, Yang Li, and Shengyu Zhang. Tight bounds on the randomized communication complexity of symmetric XOR functions in one-way and SMP models. *CoRR*, abs/1101.4555, 2011. arXiv:1101.4555.
- 31 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.
- 32 Yang Liu and Shengyu Zhang. Quantum and randomized communication complexity of XOR functions in the SMP model. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:10, 2013. URL: <http://eccc.hpi-web.de/report/2013/010>.

- 33 Shachar Lovett. Unconditional pseudorandom generators for low degree polynomials. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 557–562, 2008.
- 34 Shachar Lovett. Recent advances on the log-rank conjecture in communication complexity. *Bulletin of the EATCS*, 112, 2014. URL: <http://eatcs.org/beatcs/index.php/beatcs/article/view/260>.
- 35 Frédéric Magniez, Ashwin Nayak, Miklos Santha, Jonah Sherman, Gábor Tardos, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. *CoRR*, abs/1309.7565, 2013. arXiv:1309.7565.
- 36 Frédéric Magniez, Ashwin Nayak, Miklos Santha, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, pages 317–329, 2011.
- 37 Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014. doi:10.1145/2627692.2627694.
- 38 Ashley Montanaro and Tobias Osborne. On the communication complexity of XOR functions. *CoRR*, abs/0909.3392, 2009. arXiv:0909.3392.
- 39 Elchanan Mossel, Sampath Kannan, and Grigory Yaroslavtsev. Linear sketching over  $\mathbb{F}_2$ . *Electronic Colloquium on Computational Complexity (ECCC)*, 23:174, 2016. URL: <http://eccc.hpi-web.de/report/2016/174>.
- 40 Elchanan Mossel, Ryan O’Donnell, and Rocco A. Servedio. Learning juntas. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 206–212, 2003.
- 41 Ryan O’Donnell, John Wright, Yu Zhao, Xiaorui Sun, and Li-Yang Tan. A composition theorem for parity kill number. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 144–154, 2014.
- 42 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.
- 43 Michael E. Saks and Avi Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 29–38, 1986.
- 44 Swagato Sanyal. Near-optimal upper bound on fourier dimension of boolean functions in terms of fourier sparsity. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 1035–1045, 2015.
- 45 Yaoyun Shi and Zhiqiang Zhang. Communication complexities of symmetric xor functions. *Quantum Inf. Comput.*, pages 0808–1762, 2008.
- 46 Amir Shpilka, Avishay Tal, and Ben lee Volk. On the structure of boolean functions with small spectral norm. In *Innovations in Theoretical Computer Science, ITCS’14, Princeton, NJ, USA, January 12-14, 2014*, pages 37–48, 2014.
- 47 Xiaoming Sun and Chengu Wang. Randomized communication complexity for linear algebra problems over finite fields. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, pages 477–488, 2012.
- 48 Justin Thaler. Semi-streaming algorithms for annotated graph streams. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 59:1–59:14, 2016.
- 49 Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *54th Annual IEEE Symposium on Foundations of*

*Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 658–667, 2013.

- 50 Emanuele Viola. The sum of  $d$  small-bias generators fools polynomials of degree  $d$ . In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA*, pages 124–127, 2008.
- 51 Omri Weinstein and David P. Woodruff. The simultaneous communication of disjointness with applications to data streams. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 1082–1093, 2015.
- 52 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1-2):1–157, 2014. doi:10.1561/04000000060.
- 53 Andrew Chi-Chih Yao. Lower bounds by probabilistic arguments (extended abstract). In *24th Annual Symposium on Foundations of Computer Science, Tucson, Arizona, USA, 7-9 November 1983*, pages 420–428, 1983.
- 54 Grigory Yaroslavtsev. Approximate linear sketching over  $\mathbb{F}_2$ , 2017.
- 55 Zhiqiang Zhang and Yaoyun Shi. On the parity complexity measures of boolean functions. *Theor. Comput. Sci.*, 411(26-28):2612–2618, 2010. doi:10.1016/j.tcs.2010.03.027.

## A Information theory

Let  $X$  be a random variable supported on a finite set  $\{x_1, \dots, x_s\}$ . Let  $\mathcal{E}$  be any event in the same probability space. Let  $\mathbb{P}[\cdot]$  denote the probability of any event. The *conditional entropy*  $H(X | \mathcal{E})$  of  $X$  conditioned on  $\mathcal{E}$  is defined as follows.

► **Definition 41** (Conditional entropy).

$$H(X | \mathcal{E}) := \sum_{i=1}^s \mathbb{P}[X = x_i | \mathcal{E}] \log_2 \frac{1}{\mathbb{P}[X = x_i | \mathcal{E}]}$$

An important special case is when  $\mathcal{E}$  is the entire sample space. In that case the above conditional entropy is referred to as the *Shannon entropy*  $H(X)$  of  $X$ .

► **Definition 42** (Entropy).

$$H(X) := \sum_{i=1}^s \mathbb{P}[X = x_i] \log_2 \frac{1}{\mathbb{P}[X = x_i]}$$

Let  $Y$  be another random variable in the same probability space as  $X$ , taking values from a finite set  $\{y_1, \dots, y_t\}$ . Then the conditional entropy of  $X$  conditioned on  $Y$ ,  $H(X | Y)$ , is defined as follows.

► **Definition 43.**

$$H(X | Y) = \sum_{i=1}^t \mathbb{P}[Y = y_i] \cdot H(X | Y = y_i)$$

We next define the binary entropy function  $H_b(\cdot)$ .

► **Definition 44** (Binary entropy). For  $p \in (0, 1)$ , the binary entropy of  $p$ ,  $H_b(p)$ , is defined to be the Shannon entropy of a random variable taking two distinct values with probabilities  $p$  and  $1 - p$ .

$$H_b(p) := p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p}.$$

The following properties of entropy and conditional entropy will be useful.

► **Fact 45.**

- (1) Let  $X$  be a random variable supported on a finite set  $\mathcal{A}$ , and let  $Y$  be another random variable in the same probability space. Then  $0 \leq H(X | Y) \leq H(X) \leq \log_2 |\mathcal{A}|$ .
- (2) (Sub-additivity of conditional entropy). Let  $X_1, \dots, X_n$  be  $n$  jointly distributed random variables in some probability space, and let  $Y$  be another random variable in the same probability space, all taking values in finite domains. Then,

$$H(X_1, \dots, X_n | Y) \leq \sum_{i=1}^n H(X_i | Y).$$

- (3) Let  $X_1, \dots, X_n$  are independent random variables taking values in finite domains. Then,

$$H(X_1, \dots, X_n) = \sum_{i=1}^n H(X_i).$$

- (4) (Taylor expansion of binary entropy in the neighbourhood of  $\frac{1}{2}$ ).

$$H_b(p) = 1 - \frac{1}{2 \log_e 2} \sum_{n=1}^{\infty} \frac{(1-2p)^{2n}}{n(2n-1)}$$

► **Definition 46** (Mutual information). Let  $X$  and  $Y$  be two random variables in the same probability space, taking values from finite sets. The mutual information between  $X$  and  $Y$ ,  $I(X; Y)$ , is defined as follows.

$$I(X; Y) := H(X) - H(X | Y).$$

It can be shown that  $I(X; Y)$  is symmetric in  $X$  and  $Y$ , i.e.  $I(X; Y) = I(Y; X) = H(Y) - H(Y | X)$ .

The following observation follows immediately from the first inequality of Fact 45 (1).

► **Observation 47.** For any two random variables  $X$  and  $Y$ ,  $I(X; Y) \leq H(X)$ .

## A.1 Proof of Fact 16

Let  $\mathbb{E}X = \delta$ . Then,

$$H(X) = \begin{cases} 1 & \text{with probability } \frac{1}{2} + \frac{\delta}{2} \\ -1 & \text{with probability } \frac{1}{2} - \frac{\delta}{2} \end{cases}$$

So,

$$\begin{aligned} H(X) &= H_b\left(\frac{1}{2} + \frac{\delta}{2}\right) \\ &= 1 - \frac{1}{2 \log_e 2} \sum_{n=1}^{\infty} \frac{\delta^{2n}}{n(2n-1)} \quad (\text{From Fact 45 (4)}) \\ &\leq 1 - \frac{\delta^2}{2}. \end{aligned}$$

**B Deterministic  $\mathbb{F}_2$ -sketching**

In the deterministic case it will be convenient to represent  $\mathbb{F}_2$ -sketch of a function  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  as a  $d \times n$  matrix  $M_f \in \mathbb{F}_2^{d \times n}$  that we call the *sketch matrix*. The  $d$  rows of  $M_f$  correspond to vectors  $\alpha_1, \dots, \alpha_d$  used in the deterministic sketch so that the sketch can be computed as  $M_f x$ . W.l.o.g below we will assume that the sketch matrix  $M_f$  has linearly independent rows and that the number of rows in it is the smallest possible among all sketch matrices (ties in the choice of the sketch matrix are broken arbitrarily).

The following fact is standard (see e.g. [38, 16]):

► **Fact 48.** *For any function  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  it holds that  $D^{lin}(f) = \dim(f) = \text{rank}(M_f)$ . Moreover, set of rows of  $M_f$  forms a basis for a subspace  $A \leq \mathbb{F}_2^n$  containing all non-zero coefficients of  $f$ .*

**B.1 Disperser argument**

We show that the following basic relationship holds between deterministic linear sketching complexity and the property of being an affine disperser. For randomized  $\mathbb{F}_2$ -sketching an analogous statement holds for affine extractors as shown in Lemma 56.

► **Definition 49** (Affine disperser). A function  $f$  is an affine disperser of dimension at least  $d$  if for any affine subspace of  $\mathbb{F}_2^n$  of dimension at least  $d$  the restriction of  $f$  on it is a non-constant function.

► **Lemma 50.** *Any function  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  which is an affine disperser of dimension at least  $d$  has deterministic linear sketching complexity at least  $n - d + 1$ .*

**Proof.** Assume for the sake of contradiction that there exists a linear sketch matrix  $M_f$  with  $k \leq n - d$  rows and a deterministic function  $g$  such that  $g(M_f x) = f(x)$  for every  $x \in \mathbb{F}_2^n$ . For any vector  $b \in \mathbb{F}_2^k$ , which is in the span of the columns of  $M_f$ , the set of vectors  $x$  which satisfy  $M_f x = b$  forms an affine subspace of dimension at least  $n - k \geq d$ . Since  $f$  is an affine disperser for dimension at least  $d$  the restriction of  $f$  on this subspace is non-constant. However, the function  $g(M_f x) = g(b)$  is constant on this subspace and thus there exists  $x$  such that  $g(M_f x) \neq f(x)$ , a contradiction. ◀

**B.2 Composition and convolution**

In order to prove a composition theorem for  $D^{lin}$  we introduce the following operation on matrices which for a lack of a better term we call matrix super-slam<sup>16</sup>.

► **Definition 51** (Matrix super-slam). For two matrices  $A \in \mathbb{F}_2^{a \times n}$  and  $B \in \mathbb{F}_2^{b \times m}$  their *super-slam*  $A \dagger B \in \mathbb{F}_2^{ab^n \times nm}$  is a block matrix consisting of  $a$  blocks  $(A \dagger B)_i$ . The  $i$ -th block  $(A \dagger B)_i \in \mathbb{F}_2^{b^n \times nm}$  is constructed as follows: for every vector  $j \in \{1, \dots, b\}^n$  the corresponding row of  $(A \dagger B)_i$  is defined as  $(A_{i,1}B_{j_1}, A_{i,2}B_{j_2}, \dots, A_{i,n}B_{j_n})$ , where  $B_k$  denotes the  $k^{th}$  row of  $B$ .

► **Proposition 52.**  $\text{rank}(A \dagger B) \geq \text{rank}(A)\text{rank}(B)$ .

<sup>16</sup>This name was suggested by Chris Ramsey.

**Proof.** Consider the matrix  $C$  which is a subset of rows of  $A \dagger B$  where from each block  $(A \dagger B)_i$  we select only  $b$  rows corresponding to the vectors  $j$  of the form  $\alpha^n$  for all  $\alpha \in \{1, \dots, b\}$ . Note that  $C \in \mathbb{F}_2^{ab \times mn}$  and  $C_{(i,k),(j,l)} = A_{i,j}B_{k,l}$ . Hence,  $C$  is a Kronecker product of  $A$  and  $B$  and we have:

$$\text{rank}(A \dagger B) \geq \text{rank}(C) = \text{rank}(A)\text{rank}(B). \quad \blacktriangleleft$$

The following composition theorem for  $D^{\text{lin}}$  holds as long as the inner function is balanced:

► **Lemma 53.** For  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  and  $g: \mathbb{F}_2^m \rightarrow \mathbb{F}_2$  if  $g$  is a balanced function then:

$$D^{\text{lin}}(f \circ g) \geq D^{\text{lin}}(f)D^{\text{lin}}(g)$$

**Proof.** The multilinear expansions of  $f$  and  $g$  are given as  $f(y) = \sum_{S \in \mathbb{F}_2^n} \hat{f}(S)\chi_S(y)$  and  $g(y) = \sum_{S \in \mathbb{F}_2^m} \hat{g}(S)\chi_S(y)$ . The multilinear expansion of  $f \circ g$  can be obtained as follows. For each monomial  $\hat{f}(S)\chi_S(y)$  in the multilinear expansion of  $f$  and each variable  $y_i$  substitute  $y_i$  by the multilinear expansion of  $g$  on a set of variables  $x_{m(i-1)+1, \dots, mi}$ . Multiplying all these multilinear expansions corresponding to the term  $\hat{f}(S)\chi_S$  gives a polynomial which is a sum of at most  $b^n$  monomials where  $b$  is the number of non-zero Fourier coefficients of  $g$ . Each such monomial is obtained by picking one monomial from the multilinear expansions corresponding to different variables in  $\chi_S$  and multiplying them. Note that there are no cancellations between the monomials corresponding to a fixed  $\chi_S$ . Moreover, since  $g$  is balanced and thus  $\hat{g}(\emptyset) = 0$  all monomials corresponding to different characters  $\chi_S$  and  $\chi_{S'}$  are unique since  $S$  and  $S'$  differ on some variable and substitution of  $g$  into that variable doesn't have a constant term but introduces new variables. Thus, the characteristic vectors of non-zero Fourier coefficients of  $f \circ g$  are the same as the set of rows of the super-slam of the sketch matrices  $M_f$  and  $M_g$  (note, that in the super-slam some rows can be repeated multiple times but after removing duplicates the set of rows of the super-slam and the set of characteristic vectors of non-zero Fourier coefficients of  $f \circ g$  are exactly the same). Using Proposition 52 and Fact 48 we have:

$$D^{\text{lin}}(f \circ g) = \text{rank}(M_{f \circ g}) = \text{rank}(M_f \dagger M_g) \geq \text{rank}(M_f)\text{rank}(M_g) = D^{\text{lin}}(f)D^{\text{lin}}(g). \quad \blacktriangleleft$$

Deterministic  $\mathbb{F}_2$ -sketch complexity of convolution satisfies the following property:

► **Proposition 54.**  $D^{\text{lin}}(f * g) \leq \min(D^{\text{lin}}(f), D^{\text{lin}}(g))$ .

**Proof.** The Fourier spectrum of convolution is given as  $\widehat{f * g}(S) = \hat{f}(S)\hat{g}(S)$ . Hence, the set of non-zero Fourier coefficients of  $f * g$  is the intersection of the sets of non-zero coefficients of  $f$  and  $g$ . Thus by Fact 48 we have  $D^{\text{lin}}(f * g) \leq \min(\text{rank}(M_f, M_g)) = \min(D^{\text{lin}}(f), D^{\text{lin}}(g))$ . ◀

## C Randomized $\mathbb{F}_2$ -sketching

We represent randomized  $\mathbb{F}_2$ -sketches as distributions over  $d \times n$  matrices over  $\mathbb{F}_2$ . For a fixed such distribution  $\mathcal{M}_f$  the randomized sketch is computed as  $\mathcal{M}_f x$ . If the set of rows of  $\mathcal{M}_f$  satisfies Definition 1 for some reconstruction function  $g$  then we call it a *randomized sketch matrix* for  $f$ .

### C.1 Extractor argument

We now establish a connection between randomized  $\mathbb{F}_2$ -sketching and affine extractors which will be used to show that the converse of Part 1 of Theorem 14 doesn't hold for arbitrary distributions.

► **Definition 55** (Affine extractor). A function  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  is an affine  $\delta$ -extractor if for any affine subspace  $A$  of  $\mathbb{F}_2^n$  of dimension at least  $d$  it satisfies:

$$\min_{z \in \{0,1\}} \Pr_{x \sim U(A)} [f(x) = z] > \delta.$$

► **Lemma 56.** For any  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  which is an affine  $\delta$ -extractor of dimension at least  $d$  it holds that:

$$R_\delta^{lin}(f) \geq n - d + 1.$$

**Proof.** For the sake of contradiction assume that there exists a randomized linear sketch with a reconstruction function  $g : \mathbb{F}_2^k \rightarrow \mathbb{F}_2$  and a randomized sketch matrix  $\mathcal{M}_f$  which is a distribution over matrices with  $k \leq n - d$  rows. First, we show that:

$$\Pr_{x \sim U(\mathbb{F}_2^n), M \sim \mathcal{M}_f} [g(Mx) \neq f(x)] > \delta.$$

Indeed, fix any matrix  $M \in \text{supp}(\mathcal{M}_f)$ . For any affine subspace  $\mathcal{S}$  of the form  $\mathcal{S} = \{x \in \mathbb{F}_2^n \mid Mx = b\}$  of dimension at least  $n - k \geq d$  we have that  $\min_{z \in \{0,1\}} \Pr_{x \sim U(\mathcal{S})} [f(x) = z] > \delta$ . This implies that  $\Pr_{x \sim U(\mathcal{S})} [f(x) \neq g(Mx)] > \delta$ . Summing over all subspaces corresponding to the fixed  $M$  and all possible choices of  $b$  we have that  $\Pr_{x \sim U(\mathbb{F}_2^n)} [f(x) \neq g(Mx)] > \delta$ . Since this holds for any fixed  $M$  the bound follows.

Using the above observation it follows by averaging over  $x \in \{0,1\}^n$  that there exists  $x^* \in \{0,1\}^n$  such that:

$$\Pr_{M \sim \mathcal{M}_f} [g(Mx^*) \neq f(x^*)] > \delta.$$

This contradicts the assumption that  $\mathcal{M}_f$  and  $g$  form a randomized linear sketch of dimension  $k \leq n - d$ . ◀

► **Fact 57.** The inner product function  $IP(x_1, \dots, x_n) = \sum_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}$  is an  $(1/2 - \epsilon)$ -extractor for affine subspaces of dimension  $\geq (1/2 + \alpha)n$  where  $\epsilon = \exp(-\alpha n)$ .

► **Corollary 58.** Randomized linear sketching complexity of the inner product function is at least  $n/2 - O(1)$ .

► **Remark.** We note that the extractor argument of Lemma 56 is often much weaker than the arguments we give in Part 2 and Part 3 Theorem 14 and wouldn't suffice for our applications in Section 4. In fact, the extractor argument is too weak even for the majority function  $Maj_n$ . If the first  $100\sqrt{n}$  variables of  $Maj_n$  are fixed to 0 then the resulting restriction has value 0 with probability  $1 - e^{-\Omega(n)}$ . Hence for constant error  $Maj_n$  isn't an extractor for dimension greater than  $100\sqrt{n}$ . However, as shown in Section 4.3 for constant error  $\mathbb{F}_2$ -sketch complexity of  $Maj_n$  is linear.



## C.2 Existential lower bound for arbitrary distributions

Now we are ready to show that an analog of Part 1 of Theorem 14 doesn't hold for arbitrary distributions, i.e. concentration on a low-dimensional linear subspace doesn't imply existence of randomized linear sketches of small dimension.

► **Lemma 59.** *For any fixed constant  $\epsilon > 0$  there exists a function  $f: \mathbb{F}_2^n \rightarrow \{+1, -1\}$  such that  $R_{\epsilon/8}^{\text{lin}}(f) \geq n - 3 \log n$  such that  $f$  is  $(1 - 2\epsilon)$ -concentrated on the 0-dimensional linear subspace.*

**Proof.** The proof is based on probabilistic method. Consider a distribution over functions from  $\mathbb{F}_2^n$  to  $\{+1, -1\}$  which independently assigns to each  $x$  value 1 with probability  $1 - \epsilon/4$  and value  $-1$  with probability  $\epsilon/4$ . By a Chernoff bound with probability  $e^{-\Omega(\epsilon 2^n)}$  a random function  $f$  drawn from this distribution has at least an  $\epsilon/2$ -fraction of  $-1$  values and hence  $\hat{f}(\emptyset) = \frac{1}{2^n} \sum_{\alpha \in \mathbb{F}_2^n} f(x) \geq 1 - \epsilon$ . This implies that  $\hat{f}(\emptyset)^2 \geq (1 - \epsilon)^2 \geq 1 - 2\epsilon$  so  $f$  is  $(1 - 2\epsilon)$ -concentrated on a linear subspace of dimension 0. However, as we show below the randomized sketching complexity of some functions in the support of this distribution is large.

The total number of affine subspaces of codimension  $d$  is at most  $(2 \cdot 2^n)^d = 2^{(n+1)d}$  since each such subspace can be specified by  $d$  vectors in  $\mathbb{F}_2^n$  and a vector in  $\mathbb{F}_2^d$ . The number of vectors in each such affine subspace is  $2^{n-d}$ . The probability that less than  $\epsilon/8$  fraction of inputs in a fixed subspace have value  $-1$  is by a Chernoff bound at most  $e^{-\Omega(\epsilon 2^{n-d})}$ . By a union bound the probability that a random function takes value  $-1$  on less than  $\epsilon/8$  fraction of the inputs in any affine subspace of codimension  $d$  is at most  $e^{-\Omega(\epsilon 2^{n-d})} 2^{(n+1)d}$ . For  $d \leq n - 3 \log n$  this probability is less than  $e^{-\Omega(\epsilon n)}$ . By a union bound, the probability that a random function is either not an  $\epsilon/8$ -extractor or isn't  $(1 - 2\epsilon)$ -concentrated on  $\hat{f}(\emptyset)$  is at most  $e^{-\Omega(\epsilon n)} + e^{-\Omega(\epsilon 2^n)} \ll 1$ . Thus, there exists a function  $f$  in the support of our distribution which is an  $\epsilon/8$ -extractor for any affine subspace of dimension at least  $3 \log n$  while at the same time is  $(1 - 2\epsilon)$ -concentrated on a linear subspace of dimension 0. By Lemma 56 there is no randomized linear sketch of dimension less than  $n - 3 \log n$  for  $f$  which errs with probability less than  $\epsilon/8$ . ◀

## C.3 Random $\mathbb{F}_2$ -sketching

The following result is folklore as it corresponds to multiple instances of the communication protocol for the equality function [27, 11] and can be found e.g. in [38] (Proposition 11). We give a proof for completeness.

► **Fact 60.** *A function  $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  such that  $\min_{z \in \{0,1\}} \Pr_x[f(x) = z] \leq \epsilon$  satisfies*

$$R_\delta^{\text{lin}}(f) \leq \log \frac{\epsilon 2^{n+1}}{\delta}.$$

**Proof.** We assume that  $\operatorname{argmin}_{z \in \{0,1\}} \Pr_x[f(x) = z] = 1$  as the other case is symmetric. Let  $T = \{x \in \mathbb{F}_2^n \mid f(x) = 1\}$ . For every two inputs  $x \neq x' \in T$  a random  $\mathbb{F}_2$ -sketch  $\chi_\alpha$  for  $\alpha \sim U(\mathbb{F}_2^n)$  satisfies  $\Pr[\chi_\alpha(x) \neq \chi_\alpha(x')] = 1/2$ . If we draw  $t$  such sketches  $\chi_{\alpha_1}, \dots, \chi_{\alpha_t}$  then  $\Pr[\chi_{\alpha_i}(x) = \chi_{\alpha_i}(x'), \forall i \in [t]] = 1/2^t$ . For any fixed  $x \in T$  we have:

$$\Pr[\exists x' \neq x \in T \forall i \in [t] : \chi_{\alpha_i}(x) = \chi_{\alpha_i}(x')] \leq \frac{|T| - 1}{2^t} \leq \frac{\epsilon 2^n}{2^t} \leq \frac{\delta}{2}.$$

Conditioned on the negation of the event above for a fixed  $x \in T$  the domain of  $f$  is partitioned by the linear sketches into affine subspaces such that  $x$  is the only element of  $T$  in the subspace that contains it. We only need to ensure that we can sketch  $f$  on this subspace

which we denote as  $\mathcal{A}$ . On this subspace  $f$  is isomorphic to an OR function (up to taking negations of some of the variables) and hence can be sketched using  $O(\log 1/\delta)$  uniformly random sketches with probability  $1 - \delta/2$ . For the OR-function existence of the desired protocol is clear since we just need to verify whether there exists at least one coordinate of the input that is set to 1. In case it does exist a random sketch contains this coordinate with probability  $1/2$  and hence evaluates to 1 with probability at least  $1/4$ . Repeating  $O(\log 1/\delta)$  times the desired guarantee follows.  $\blacktriangleleft$

## D Towards the proof of Conjecture 3

We call a function  $f : \mathbb{F}_2^n \rightarrow \{+1, -1\}$  *non-linear* if for all  $S \in \mathbb{F}_2^n$  there exists  $x \in \mathbb{F}_2^n$  such that  $f(x) \neq \chi_S(x)$ . Furthermore, we say that  $f$  is  $\epsilon$ -far from being linear if:

$$\max_{S \in \mathbb{F}_2^n} \left[ \Pr_{x \sim U(\mathbb{F}_2^n)} [\chi_S(x) = f(x)] \right] = 1 - \epsilon.$$

The following theorem is our first step towards resolving Conjecture 3. Since non-linear functions don't admit 1-bit linear sketches we show that the same is also true for the corresponding communication complexity problem, namely no 1-bit communication protocol for such functions can succeed with a small constant error probability.

► **Theorem 61.** *For any non-linear function  $f$  that is at most  $1/10$ -far from linear  $\mathcal{D}_{1/200}^{\rightarrow}(f^+) > 1$ .*

**Proof.** Let  $S = \arg \max_T [\Pr_{x \in \mathbb{F}_2^n} [\chi_T(x) = f(x)]]$ . Pick  $z \in \mathbb{F}_2^n$  such that  $f(z) \neq \chi_S(z)$ . Let the distribution over the inputs  $(x, y)$  be as follows:  $y \sim U(\mathbb{F}_2^n)$  and  $x \sim \mathcal{D}_y$  where  $\mathcal{D}_y$  is defined as:

$$\mathcal{D}_y = \begin{cases} y + z & \text{with probability } 1/2, \\ U(\mathbb{F}_2^n) & \text{with probability } 1/2. \end{cases}$$

Fix any deterministic Boolean function  $M(x)$  that is used by Alice to send a one-bit message based on her input. For a fixed Bob's input  $y$  he outputs  $g_y(M(x))$  for some function  $g_y$  that can depend on  $y$ . Thus, the error that Bob makes at predicting  $f$  for fixed  $y$  is at least:

$$\frac{1 - |\mathbb{E}_{x \sim \mathcal{D}_y} [g_y(M(x))f(x + y)]|}{2}.$$

The key observation is that since Bob only receives a single bit message there are only four possible functions  $g_y$  to consider for each  $y$ : constants  $-1/1$  and  $\pm M(x)$ .

### Bounding error for constant estimators

For both constant functions we introduce notation  $B_y^c = |\mathbb{E}_{x \sim \mathcal{D}_y} [g_y(M(x))f(x + y)]|$  and have:

$$B_y^c = |\mathbb{E}_{x \sim \mathcal{D}_y} [g_y(M(x))f(x + y)]| = |\mathbb{E}_{x \sim \mathcal{D}_y} [f(x + y)]| = \left| \frac{1}{2}f(z) + \frac{1}{2}\mathbb{E}_{w \sim U(\mathbb{F}_2^n)} [f(w)] \right|$$

If  $\chi_S$  is not constant then  $|\mathbb{E}_{w \sim U(\mathbb{F}_2^n)} [f(w)]| \leq 2\epsilon$  we have:

$$\left| \frac{1}{2}f(z) + \frac{1}{2}\mathbb{E}_{w \sim U(\mathbb{F}_2^n)} [f(w)] \right| \leq \frac{1}{2} (|f(z)| + |\mathbb{E}_{w \sim U(\mathbb{F}_2^n)} [f(w)]|) \leq 1/2 + \epsilon.$$

If  $\chi_S$  is a constant then w.l.o.g  $\chi_S = 1$  and  $f(z) = -1$ . Also  $\mathbb{E}_{w \sim U(\mathbb{F}_2^n)}[f(w)] \geq 1 - 2\epsilon$ . Hence we have:

$$\left| \frac{1}{2}f(z) + \frac{1}{2}\mathbb{E}_{w \sim U(\mathbb{F}_2^n)}[f(w)] \right| = \frac{1}{2}|-1 + \mathbb{E}_{w \sim U(\mathbb{F}_2^n)}[f(w)]| \leq \epsilon.$$

Since  $\epsilon \leq 1/10$  in both cases  $B_y^c \leq \frac{1}{2} + \epsilon$  which is the bound we will use below.

### Bounding error for message-based estimators

For functions  $\pm M(x)$  we need to bound  $|\mathbb{E}_{x \sim D_y}[M(x)f(x+y)]|$ . We denote this expression as  $B_y^M$ . Proposition 62 shows that  $\mathbb{E}_y[B_y^M] \leq \frac{\sqrt{2}}{2}(1 + \epsilon)$ .

► **Proposition 62.**  $\mathbb{E}_{y \sim U(\mathbb{F}_2^n)}[|\mathbb{E}_{x \sim D_y}[M(x)f(x+y)]|] \leq \frac{\sqrt{2}}{2}(1 + \epsilon)$ .

We have:

$$\begin{aligned} & \mathbb{E}_y [|\mathbb{E}_{x \sim D_y}[M(x)f(x+y)]|] \\ &= \mathbb{E}_y \left[ \left| \frac{1}{2}(M(y+z)f(z) + \mathbb{E}_{x \sim D_y}[M(x)f(x+y)]) \right| \right] \\ &= \frac{1}{2} \mathbb{E}_y [|(M(y+z)f(z) + (M * f)(y))|] \\ &\leq \frac{1}{2} \left( \mathbb{E}_y [((M(y+z)f(z) + (M * f)(y))^2)] \right)^{1/2} \\ &= \frac{1}{2} \left( \mathbb{E}_y [((M(y+z)f(z))^2 + ((M * f)(y))^2 + 2M(y+z)f(z)(M * f)(y))] \right)^{1/2} \\ &= \frac{1}{2} \left( \mathbb{E}_y [((M(y+z)f(z))^2) + \mathbb{E}_y [((M * f)(y))^2] + \right. \\ &\quad \left. 2\mathbb{E}_y [M(y+z)f(z)(M * f)(y))] \right)^{1/2} \end{aligned}$$

We have  $(M(y+z)f(z))^2 = 1$  and also by Parseval, expression for the Fourier spectrum of convolution and Cauchy-Schwarz:

$$\mathbb{E}_y [((M * f)(y))^2] = \sum_{S \in \mathbb{F}_2^n} \widehat{M * f}(S)^2 = \sum_{S \in \mathbb{F}_2^n} \widehat{M}(S)^2 \widehat{f}(S)^2 \leq \|M\|_2 \|f\|_2 = 1$$

Thus, it suffices to give a bound on  $\mathbb{E}[M(y+z)f(z)(M * f)(y)]$ . First we give a bound on  $(M * f)(y)$ :

$$(M * f)(y) = \mathbb{E}_x[M(x)f(x+y)] \leq \mathbb{E}_x[M(x)\chi_S(x+y)] + 2\epsilon$$

Plugging this in we have:

$$\begin{aligned} & \mathbb{E}_y[M(y+z)f(z)(M * f)(y)] \\ &= -\chi_S(z)\mathbb{E}_y[M(y+z)(M * f)(y)] \\ &\leq -\chi_S(z)\mathbb{E}_y[M(y+z)(M * \chi_S)(y)] + 2\epsilon \\ &= -\chi_S(z)(M * (M * \chi_S))(z) + 2\epsilon \\ &= -\chi_S(z)^2 \widehat{M}(S)^2 + 2\epsilon \\ &\leq 2\epsilon. \end{aligned}$$

where we used the fact that the Fourier spectrum of  $(M * (M * \chi_S))$  is supported on  $S$  only and  $M * \widehat{(M * \chi_S)}(S) = \widehat{M}^2(S)$  and thus  $(M * (M * \chi_S))(z) = \widehat{M}^2(S)\chi_S(z)$ .

Thus, overall, we have:

$$\mathbb{E}_y [|\mathbb{E}_{x \sim D_y}[M(x)f(x+y)]|] \leq \frac{1}{2}\sqrt{2+4\epsilon} \leq \frac{\sqrt{2}}{2}(1 + \epsilon). \quad \blacktriangleleft$$

**Putting things together**

We have that the error that Bob makes is at least:

$$\mathbb{E}_y \left[ \frac{1 - \max(B_y^c, B_y^M)}{2} \right] = \frac{1 - \mathbb{E}_y[\max(B_y^c, B_y^M)]}{2}$$

Below we now bound  $\mathbb{E}_y[\max(B_y^c, B_y^M)]$  from above by 99/100 which shows that the error is at least 1/200.

$$\begin{aligned} & \mathbb{E}_y[\max(B_y^c, B_y^M)] \\ &= \Pr[B_y^M \geq 1/2 + \epsilon] \mathbb{E}[B_y^M | B_y^M \geq 1/2 + \epsilon] + \Pr[B_y^M < 1/2 + \epsilon] \left( \frac{1}{2} + \epsilon \right) \\ &= \mathbb{E}_y[B_y^M] + \Pr[B_y^M < 1/2 + \epsilon] \left( \frac{1}{2} + \epsilon - \mathbb{E}[B_y^M | B_y^M < 1/2 + \epsilon] \right) \end{aligned}$$

Let  $\delta = \Pr[B_y^M < 1/2 + \epsilon]$ . Then the first of the expressions above gives the following bound:

$$\mathbb{E}_y[\max(B_y^c, B_y^M)] \leq (1 - \delta) + \delta \left( \frac{1}{2} + \epsilon \right) = 1 - \frac{\delta}{2} + \epsilon\delta \leq 1 - \frac{\delta}{2} + \epsilon$$

The second expression gives the following bound:

$$\mathbb{E}_y[\max(B_y^c, B_y^M)] \leq \frac{\sqrt{2}}{2} (1 + \epsilon) + \delta \left( \frac{1}{2} + \epsilon \right) \leq \frac{\sqrt{2}}{2} + \frac{\delta}{2} + \frac{\sqrt{2}}{2} \epsilon + \epsilon.$$

These two bounds are equal for  $\delta = 1 - \frac{\sqrt{2}}{2} (1 + \epsilon)$  and hence the best of the two bounds is always at most  $(\frac{\sqrt{2}}{4} + \frac{1}{2}) + \epsilon \left( \frac{\sqrt{2}}{4} + 1 \right) \leq \frac{99}{100}$  where the last inequality uses the fact that  $\epsilon \leq \frac{1}{10}$ .

**E Auxiliary Proofs****E.1 Proof of Proposition 17**

Without loss of generality assume that  $p = \Pr[X = 1]$

$$\begin{aligned} \text{Var}[X] &= \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \\ &= 1 - (\mathbb{E}[X])^2 \quad (X^2 = 1 \text{ as } X \text{ is supported on } \{1, -1\}) \\ &= 1 - (p \cdot 1 + (1 - p)(-1))^2 \\ &= 1 - (2p - 1)^2 \\ &= 4p(1 - p) \end{aligned}$$

Since  $p \leq \frac{1}{2}$ ,  $4(1 - p) \in [2, 4]$  and the proposition follows.

**E.2 Proof of Lemma 21**

Let  $p \in \mathbb{F}_2[x_1, \dots, x_n]$  be the  $\mathbb{F}_2$ -polynomial corresponding to  $f$ . Fix one monomial  $\mathcal{M} = \prod_{i \in S} x_i$  of the largest degree. Thus  $|S| = d$ . We will show that for each assignment  $a_{\bar{S}}$  to the variables outside of  $S$ , there is an assignment  $a_S$  to the variables in  $S$  such that  $p(a_S, a_{\bar{S}}) = 1$ . This will prove that there are at least  $2^{n-d}$  assignments on which  $p$  evaluates to 1, and will thus imply the lemma.

To this end, fix an assignment  $a_{\bar{S}}$  to the variables in  $\bar{S}$ . Let  $p|_{\bar{S} \leftarrow a_{\bar{S}}}$  be the polynomial obtained from  $p$  by setting the variables in  $\bar{S}$  according to  $a_{\bar{S}}$ . Notice that since  $\mathcal{M}$  was a monomial of largest degree in  $p$ ,  $\mathcal{M}$  continues to be a monomial in  $p|_{\bar{S} \leftarrow a_{\bar{S}}}$ . Thus  $p|_{\bar{S} \leftarrow a_{\bar{S}}}$  is a non-constant polynomial in the variables  $\{x_i \mid i \in S\}$ . In particular, this implies that there exists an assignment  $a_S$  to the variables in  $S$ , such that  $p|_{\bar{S} \leftarrow a_{\bar{S}}}(a_S) = 1$  (see the discussion in the paragraph after fact 20). This in turn implies that  $p(a_S, a_{\bar{S}}) = 1$ .



# Communication Complexity with Small Advantage

Thomas Watson<sup>1</sup>

Department of Computer Science, University of Memphis  
Memphis, TN, USA  
Thomas.Watson@memphis.edu

---

## Abstract

We study problems in randomized communication complexity when the protocol is only required to attain some small advantage over purely random guessing, i.e., it produces the correct output with probability at least  $\epsilon$  greater than one over the codomain size of the function. Previously, Braverman and Moitra (STOC 2013) showed that the set-intersection function requires  $\Theta(\epsilon n)$  communication to achieve advantage  $\epsilon$ . Building on this, we prove the same bound for several variants of set-intersection: (1) the classic “tribes” function obtained by composing with AND (provided  $1/\epsilon$  is at most the width of the AND), and (2) the variant where the sets are uniquely intersecting and the goal is to determine partial information about (say, certain bits of the index of) the intersecting coordinate.

**2012 ACM Subject Classification** Theory of computation → Communication complexity

**Keywords and phrases** Communication, complexity, small, advantage

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.9

**Related Version** <https://eccc.weizmann.ac.il/report/2016/148/>

**Funding** Supported by NSF grant CCF-1657377.

## 1 Introduction

In randomized communication complexity, protocols are commonly required to succeed with probability at least some constant less than 1, such as  $3/4$ . Achieving success probability one over the codomain size of the function is trivial by outputting a uniformly random guess. There is a spectrum of complexities between these extremes, where we require a protocol to achieve success probability  $\epsilon$  greater than one over the codomain size, i.e., *advantage*  $\epsilon$ . We study the fine-grained question “How does the communication complexity of achieving advantage  $\epsilon$  depend on  $\epsilon$ ?”

Formally, for a two-party function  $F$ , let  $R_p(F)$  denote the minimum worst-case communication cost of any randomized protocol (with both public and private coins) that is  $p$ -correct in the sense that for each input  $(X, Y)$  in the domain of  $F$ , it outputs  $F(X, Y)$  with probability at least  $p$ .

First let us consider functions with codomain size 2. One observation is that running an advantage- $\epsilon$  protocol  $O(1/\epsilon^2)$  times independently and taking the majority outcome yields an advantage- $1/4$  protocol (we call this “majority-amplification”); i.e.,  $R_{1/2+\epsilon}(F) \geq \Omega(\epsilon^2 R_{3/4}(F))$ . However, this does not tell the whole story; achieving advantage  $\epsilon$  may be harder than this bound suggests, depending on the function. For example, consider the well-studied functions INNER-PROD (inner product mod 2), SET-INTER (set-intersection, where 1-inputs are intersecting), and GAP-HAMMING (determining whether the Hamming distance

---

<sup>1</sup> Supported by NSF grant CCF-1657377.



is  $\geq n/2 + \sqrt{n}$  or  $\leq n/2 - \sqrt{n}$ ). Each of these three functions  $F$  satisfies  $R_{3/4}(F) = \Theta(n)$ , and yet

- $R_{1/2+\epsilon}(\text{INNER-PROD}) = \Theta(n)$  provided  $\epsilon \geq 2^{-o(n)}$  [19, §3.5–3.6 and references therein];
- $R_{1/2+\epsilon}(\text{SET-INTER}) = \Theta(\epsilon n)$  provided  $\epsilon n \geq 1$  [3, 12];
- $R_{1/2+\epsilon}(\text{GAP-HAMMING}) = \Theta(\epsilon^2 n)$  provided  $\epsilon^2 n \geq 1$  [7, 23, 22].

(We provide a proof of the GAP-HAMMING upper bound in the full version.)

Hence it is naturally interesting to study the dependence of the complexity on  $\epsilon$  for different important functions, in order to build a more complete understanding of randomized communication. For functions with codomain size greater than 2, small-advantage protocols are not even amenable to amplification, so no lower bounds for them follow a priori from lower bounds for higher-advantage protocols.

The functions we study are defined using composition. Letting  $g: \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$  be a two-party total function (usually called a *gadget*), and  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a (possibly partial) function, the two-party composed (possibly partial) function  $f \circ g^n: \mathcal{X}^n \times \mathcal{Y}^n \rightarrow \{0, 1\}$  is defined by  $(f \circ g^n)(X, Y) := f(g(X_1, Y_1), \dots, g(X_n, Y_n))$  where  $X = (X_1, \dots, X_n)$  and  $Y = (Y_1, \dots, Y_n)$  with  $X_i \in \mathcal{X}$  and  $Y_i \in \mathcal{Y}$  for each  $i$ . Sometimes, the outer function  $f$  itself will be defined using standard function composition.

In the functions  $\text{AND}_m$  and  $\text{OR}_m$ , the subscript indicates the number of input bits.

## 1.1 Tribes

Just as SET-INTER is the canonical NP-complete communication problem, so-called TRIBES is the canonical  $\Pi_2\text{P}$ -complete communication problem. A linear randomized lower bound for TRIBES (with constant advantage) was shown in [17] using information complexity (thereby giving a nearly optimal (quadratic) separation between the  $(\text{NP} \cap \text{coNP})$ -type and BPP-type communication complexity measures for a total function). This spawned a line of research on the communication complexity of read-once formulas [16, 20, 15, 9]. An alternative proof of the lower bound for TRIBES was given in [13] using the smooth rectangle bound technique introduced by [14, 7]. A multi-party version of TRIBES has been studied in the message-passing model [8].

Analogously to  $\text{SET-INTER}_m := \text{OR}_m \circ \text{AND}_2^m$ , we have the definition

$$\text{TRIBES}_{\ell, m} := \text{AND}_{\ell} \circ \text{OR}_m^{\ell} \circ \text{AND}_2^{\ell \times m} = \text{AND}_{\ell} \circ \text{SET-INTER}_m^{\ell}.$$

We always assume  $m \geq 2$  (since if  $m = 1$  then  $\text{TRIBES}_{\ell, m}$  is trivially computable with constant communication). Note that the outer function  $\text{AND}_{\ell} \circ \text{OR}_m^{\ell}$  takes a boolean  $\ell \times m$  matrix and indicates whether every row has at least one 1. For  $\text{TRIBES}_{\ell, m}$ , Alice and Bob each get such a matrix, and the above function is applied to the bitwise AND of the two matrices.

► **Theorem 1.**  $R_{1/2+\epsilon}(\text{TRIBES}_{\ell, m}) = \Theta(\epsilon \ell m)$  provided  $\epsilon \ell \geq 1$ .

The upper bound is shown as follows. Let  $M$  denote the boolean  $\ell \times m$  matrix that is fed into  $\text{AND}_{\ell} \circ \text{OR}_m^{\ell}$ . Consider the protocol in which Alice and Bob publicly sample a uniformly random set of  $4\epsilon\ell$  rows, evaluate all the bits of  $M$  in those rows (using  $O(\epsilon\ell m)$  communication), and accept iff each of those rows of  $M$  contains at least one 1. For a 1-input, this rejects with probability 0, and for a 0-input it finds an all-0 row (and hence rejects) with probability at least  $4\epsilon$ . Now if we modify the above protocol so it rejects automatically with probability  $1/2 - \epsilon$  and otherwise proceeds as before, then it rejects 1-inputs with probability  $1/2 - \epsilon$  and 0-inputs with probability at least  $(1/2 - \epsilon) + (1/2 + \epsilon) \cdot 4\epsilon \geq 1/2 + \epsilon$ . The



provision  $\epsilon\ell \geq 1$  was stated cleanly to ensure that we can round  $4\epsilon\ell$  up to an integer without affecting the asymptotic complexity. (If  $\epsilon\ell \leq o(1)$  then just evaluating a single row of  $M$  takes  $\omega(\epsilon\ell m)$  communication.) The lower bound, which we prove in Section 2, does not require this provision.

Let us describe why the  $\Omega(\epsilon\ell m)$  lower bound does not follow straightforwardly from known results. First of all, applying standard majority-amplification to the known  $\Omega(\ell m)$  lower bound for constant advantage only yields an  $\Omega(\epsilon^2\ell m)$  lower bound. What about the technique used by [12] to give a simplified proof of the tight  $\epsilon$ -advantage lower bound for SET-INTER? Let us summarize this technique (known as “and-amplification”) as applied to the complement function SET-DISJ: Running an  $\epsilon$ -advantage protocol  $O(1/\epsilon)$  times, and accepting iff all runs accept, yields a so-called SBP-type protocol, for which the complexity is characterized by the corruption bound. Hence the  $\epsilon$ -advantage complexity is always at least  $\Omega(\epsilon)$  times the corruption bound (which is  $\Omega(n)$  for SET-DISJ $_n$  by [21]). Applied to TRIBES $_{\ell,m}$  (or its complement), the and-amplification technique can only yield an essentially  $\Omega(\epsilon \cdot \max(\ell, m))$  lower bound, since TRIBES $_{\ell,m}$  has an  $O(\ell \log m)$ -communication nondeterministic (in particular, SBP-type) protocol and an  $O(m + \log \ell)$ -communication conondeterministic (in particular, coSBP-type) protocol.

Can we leverage the known smooth rectangle lower bound for TRIBES $_{\sqrt{n},\sqrt{n}}$  [13]? The smooth rectangle bound in general characterizes the complexity of so-called WAPP-type protocols [14, 10]. Thus if we could “amplify” an  $\epsilon$ -advantage protocol into a (sufficiently-large-constant-advantage) WAPP-type protocol with  $o(1/\epsilon^2)$  factor overhead, we would get a nontrivial  $\epsilon$ -advantage lower bound for TRIBES $_{\sqrt{n},\sqrt{n}}$ . However, the smooth rectangle lower bound for GAP-HAMMING [7] shows that this cannot always be done, i.e., an  $\Omega(1/\epsilon^2)$  overhead is sometimes necessary (at least for general partial functions).

Instead, our basic approach to prove the lower bound in Theorem 1 is to combine the information complexity techniques of [3] (developed for the  $\epsilon$ -advantage lower bound for SET-INTER) with the information complexity techniques of [17] (developed for the constant-advantage lower bound for TRIBES). However, in trying to combine these techniques, there are a variety of technical hurdles, which require several new ideas to overcome.

## 1.2 What if $\epsilon\ell \leq o(1)$ ?

As mentioned above, when  $\epsilon\ell \leq o(1)$ , our proof of the  $O(\epsilon\ell m)$  upper bound for TRIBES $_{\ell,m}$  breaks down. So what upper bound can we give in this case? Let us restrict our attention to  $\ell = 2$  (and let  $\epsilon > 0$  be arbitrary).

First of all, notice that the communication protocol in Section 1.1 is actually a *query complexity* (a.k.a. *decision tree complexity*) upper bound for the outer function. A communication protocol for any composed function (with constant-size gadget) can simulate a decision tree for the outer function, using constant communication to evaluate the output of each gadget when queried by the decision tree. In the next paragraph, we describe an  $O(\sqrt{\epsilon m})$ -query  $\epsilon$ -advantage randomized decision tree for  $\text{AND}_2 \circ \text{OR}_m^2$  (thus showing that  $R_{1/2+\epsilon}(\text{TRIBES}_{2,m}) \leq O(\sqrt{\epsilon m})$  provided  $\sqrt{\epsilon m} \geq 1$ ).

Say the input is  $z = (z_1, z_2) \in \{0, 1\}^m \times \{0, 1\}^m$ . Consider the following randomized decision tree: Pick  $S_1, S_2 \subseteq [m]$  both of size  $2\sqrt{\epsilon m}$ , independently uniformly at random, and accept iff  $z_1|_{S_1}$  and  $z_2|_{S_2}$  each contain at least one 1. For a 1-input, each of these two events happens with probability at least  $2\sqrt{\epsilon}$ , so they happen simultaneously with probability at least  $4\epsilon$ . For a 0-input, one of the two events never happens, and hence this accepts with probability 0. Now if we modify the above randomized decision tree so it accepts automatically with probability  $1/2 - \epsilon$  and otherwise proceeds as before, then it accepts 0-inputs with

probability  $1/2 - \epsilon$  and 1-inputs with probability at least  $(1/2 - \epsilon) + (1/2 + \epsilon) \cdot 4\epsilon \geq 1/2 + \epsilon$ , and queries at most  $O(\sqrt{\epsilon m})$  bits.

We conjecture that this communication upper bound is tight, i.e.,  $R_{1/2+\epsilon}(\text{TRIBES}_{2,m}) \geq \Omega(\sqrt{\epsilon m})$ . This remains open, but we at least prove the query complexity version of this conjecture, which can be construed as evidence for the communication version. (The query complexity measure  $R_p^{\text{dt}}(f)$  is defined in the natural way.)

► **Theorem 2.**  $R_{1/2+\epsilon}^{\text{dt}}(\text{AND}_2 \circ \text{OR}_m^2) = \Theta(\sqrt{\epsilon m})$  provided  $\sqrt{\epsilon m} \geq 1$ .

We prove the lower bound of Theorem 2 in Section 3. There are some known powerful “simulation theorems” (e.g., [10]) for converting query lower bounds for an outer function into matching communication lower bounds for a composed function; however, we lack a simulation theorem powerful enough to convert Theorem 2 into a communication lower bound. Furthermore, we have not found a way to emulate the query lower bound proof with information complexity tools to get a communication lower bound.

### 1.3 Which part contains the intersecting coordinate?

We now turn our attention away from TRIBES.

Suppose Alice and Bob are given uniquely intersecting subsets  $X$  and  $Y$  from a universe of size  $n$  that is partitioned into  $\ell \geq 2$  equal-size parts, and they wish to identify which part contains the intersection. Of course, they can succeed with probability  $1/\ell$  by random guessing without communicating about their sets. To do better they can use the following protocol.

Alice and Bob publicly sample a uniformly random subset  $S$  of size  $2\epsilon n$   
 They exchange  $X \cap S$  and  $Y \cap S$  using  $4\epsilon n$  bits of communication  
 If  $S \cap X \cap Y \neq \emptyset$  they output the label of the part containing the known point of intersection  
 Otherwise they publicly sample and output a uniformly random part label

This protocol succeeds with probability  $2\epsilon + (1 - 2\epsilon)/\ell = 1/\ell + (1 - 1/\ell) \cdot 2\epsilon \geq 1/\ell + \epsilon$ . We prove that this is optimal:  $\Omega(\epsilon n)$  communication is necessary to achieve advantage  $\epsilon$ .<sup>2</sup>

We state this using the following notation. Define the partial function  $\text{WHICH}_\ell: \{0, 1\}^\ell \rightarrow [\ell]$  that takes a string of Hamming weight 1 and outputs the coordinate of the only 1. Define the “unambiguous-or” function  $\text{UNAMBIG-OR}_m$  as  $\text{OR}_m$  restricted to the domain of strings of Hamming weight 0 or 1. Define the “unambiguous-set-intersection” function<sup>3</sup>  $\text{UNAMBIG-INTER}_m := \text{UNAMBIG-OR}_m \circ \text{AND}_2^m$ .

► **Theorem 3.**  $R_{1/\ell+\epsilon}(\text{WHICH}_\ell \circ \text{UNAMBIG-INTER}_m^\ell) = \Theta(\ell m)$  provided  $\ell m \geq 1$ .

We prove the lower bound in Section 4.

The key to the proof is in relating the complexity of  $\text{WHICH}_\ell \circ F^\ell$  to the complexity of  $F$  (for an arbitrary two-party  $F$  with boolean output). It is natural to conjecture that the complexity goes up by roughly a factor of  $\ell$  after composition with  $\text{WHICH}_\ell$ ; this is an alternative form of direct sum problem. In the standard direct sum setting, the goal is to evaluate  $F$  on each of  $\ell$  independent inputs; our form is equivalent but under the

<sup>2</sup> We mention that there is some prior work studying a peripherally related topic: the randomized complexity of “finding the exact intersection” [2, 5, 6], albeit not restricting the size of the intersection.

<sup>3</sup> Sometimes this is called “unique-set-intersection”, but our terminology is more consistent with classical complexity; see [11].

promise that one of the inputs evaluates to 1 and the rest to 0. Thus proving the direct sum conjecture (factor  $\ell$  increase in complexity) appears qualitatively harder in our setting than in the standard setting. We show an information complexity version of the conjecture, and we combine this with [3] to derive Theorem 3.

For worst-case communication, we at least show that the complexity does not go down after composition with  $\text{WHICH}_\ell$ . In particular, this yields a simple proof of a communication lower bound due to [18] which implies the communication complexity class separation  $\text{UP} \cap \text{coUP} \not\subseteq \text{BPP}$ . The proof in [18] is technically somewhat involved, exploiting a “fine-tuned” version of Razborov’s corruption lemma [21]; our simple proof of the same lower bound is by a black-box reduction to the standard (constant-advantage) lower bound for  $\text{UNAMBIG-INTER}$ .

## 1.4 Preliminaries

We first note that it suffices to prove our lower bounds for  $\text{AND}_\ell \circ \text{OR}_m^\ell \circ \text{AND}_2^{\ell \times m}$  (Theorem 1) and  $\text{WHICH}_\ell \circ \text{UNAMBIG-OR}_m^\ell \circ \text{AND}_2^{\ell \times m}$  (Theorem 3) with  $\text{AND}_2$  replaced by a different two-party gadget, namely the equality function on trits  $3\text{EQ}: \{0, 1, 2\} \times \{0, 1, 2\} \rightarrow \{0, 1\}$  ( $3\text{EQ}(X, Y) = 1$  iff  $X = Y$ ). This is because  $3\text{EQ}$  reduces to  $\text{UNAMBIG-OR}_3 \circ \text{AND}_2^3$  (with Alice and Bob both mapping their trit to its characteristic bit vector of Hamming weight 1), and thus  $\text{UNAMBIG-OR}_m \circ 3\text{EQ}^m$  reduces to  $\text{UNAMBIG-OR}_{3m} \circ \text{AND}_2^{3m}$ , and  $\text{OR}_m \circ 3\text{EQ}^m$  reduces to  $\text{OR}_{3m} \circ \text{AND}_2^{3m}$ .

We now mention some notational conventions. We use  $\mathbb{P}$  for probability,  $\mathbb{E}$  for expectation,  $\mathbb{H}$  for Shannon entropy,  $\mathbb{I}$  for mutual information,  $\mathbb{D}$  for relative entropy, and  $\Delta$  for statistical (total variation) distance. We use bold letters to denote random variables, and non-bold letters for particular outcomes. We use  $\in_{\mathbb{U}}$  to denote that a random variable is distributed uniformly over some set.

All protocols  $\Pi$  are randomized and have both public and private coins, unless otherwise stated, and we use  $CC(\Pi)$  to denote the worst-case communication cost. When we speak of an arbitrary  $F$ , by default it is assumed to be a two-party partial function. Also, complexity class names (such as  $\text{BPP}$ ) refer to classes of (families of) two-party partial functions with polylogarithmic communication protocols of the relevant type.

## 2 Communication Lower Bound for Tribes

The upper bound for Theorem 1 was shown in Section 1.1. In this section we give the proof of the lower bound, which is broken into four steps corresponding to the four subsections.

### 2.1 Step 1: Conditioning and direct sum

In this step, we use known techniques [1, 17, 3] to show that it suffices to prove a certain information complexity lower bound for a constant-size function. There are no substantially new ideas in this step.

As noted in Section 1.4, it suffices to prove the lower bound for  $\text{TRIBES}'_{\ell, m} := \text{AND}_\ell \circ \text{OR}_m^\ell \circ 3\text{EQ}^{\ell \times m}$  instead of  $\text{TRIBES}_{\ell, m}$ . Suppose for contradiction there is a  $(1/2 + \epsilon)$ -correct protocol  $\Pi$  for  $\text{TRIBES}'_{\ell, m}$  with  $CC(\Pi) \leq o(\ell m)$ . As a technicality, we assume  $\Pi$  has been converted into a private-coin-only protocol, where Alice first privately samples the public coins (if any) and sends them to Bob. (This could blow up the communication, but we will only use the fact that the “original communication” part of the transcript has bounded length, not the “public coins” part.)

We can think of the input to  $\text{TRIBES}'_{\ell,m}$  as an  $\ell \times m$  table where each cell has two trits, one for Alice and one for Bob. As is standard in information complexity lower bounds, we define a distribution over inputs, equipped with a “conditioning scheme” that decomposes the distribution into a mixture of product distributions (where Alice’s and Bob’s parts of the input are independent of each other). We do this by placing a uniformly random 1-input to 3EQ at a uniformly random cell in each row, and for each of the remaining cells choosing at random a rectangular “window” of 0-inputs to 3EQ, from which the input to that cell is drawn.

Formally, let us define  $\mathscr{W}_1 := \{\{00\}, \{11\}, \{22\}\}$  as the set of “1-windows” of 3EQ, and define  $\mathscr{W}_0 := \{\{01, 02\}, \{10, 12\}, \{20, 21\}, \{10, 20\}, \{01, 21\}, \{02, 12\}\}$  as the set of “0-windows” of 3EQ. We define a probability space with the following random variables:  $\mathbf{X} \in \{0, 1, 2\}^{\ell \times m}$ ,  $\mathbf{Y} \in \{0, 1, 2\}^{\ell \times m}$ ,  $\boldsymbol{\tau} \in \{0, 1\}^*$ ,  $\mathbf{J} \in [m]^\ell$ , and  $\mathbf{W} \in (2^{\{0,1,2\}^2})^{\ell \times m}$ . Choose  $\mathbf{J}$  uniformly, and for each  $(i, j) \in [\ell] \times [m]$  independently, let

$$\mathbf{W}_{i,j} \in_{\mathbf{u}} \begin{cases} \mathscr{W}_1 & \text{if } j = \mathbf{J}_i \\ \mathscr{W}_0 & \text{if } j \neq \mathbf{J}_i \end{cases}$$

and let  $(\mathbf{X}_{i,j}, \mathbf{Y}_{i,j}) \in_{\mathbf{u}} \mathbf{W}_{i,j}$ . Note that  $\mathbf{X}\mathbf{Y}$  is supported on 1-inputs of  $\text{TRIBES}'_{\ell,m}$ , and that  $\mathbf{X}$  and  $\mathbf{Y}$  are independent conditioned on  $\mathbf{W}$ . Finally, let  $\boldsymbol{\tau}$  be the random transcript on input  $(\mathbf{X}, \mathbf{Y})$ .

Define  $\mathbf{X}_{-\mathbf{J}} := (\mathbf{X}_{i,j})_{j \neq \mathbf{J}_i}$  (and  $\mathbf{Y}_{-\mathbf{J}}$  similarly), and let  $\boldsymbol{\tau}^{\text{C}}$  denote the “original communication” part of  $\boldsymbol{\tau}$ , and  $\boldsymbol{\tau}^{\text{R}}$  denote the “public coins” part of  $\boldsymbol{\tau}$ . We have

$$\mathbb{I}(\boldsymbol{\tau} ; \mathbf{X}_{-\mathbf{J}}\mathbf{Y}_{-\mathbf{J}} \mid \mathbf{W}) = \mathbb{I}(\boldsymbol{\tau}^{\text{C}} ; \mathbf{X}_{-\mathbf{J}}\mathbf{Y}_{-\mathbf{J}} \mid \mathbf{W}\boldsymbol{\tau}^{\text{R}}) \leq \mathbb{H}(\boldsymbol{\tau}^{\text{C}} \mid \mathbf{W}\boldsymbol{\tau}^{\text{R}}) \leq CC(\Pi) \leq o(\ell m)$$

where the equality holds by the chain rule and independence of  $\boldsymbol{\tau}^{\text{R}}$  and  $\mathbf{W}\mathbf{X}\mathbf{Y}$ . If we augment the probability space with random variables  $(i, \mathbf{k})$  sampled uniformly from  $([\ell] \times [m]) \setminus \{(i, \mathbf{J}_i) : i \in [\ell]\}$  (independent of the other random variables, conditioned on  $\mathbf{J}$ ), then by the standard direct sum property for mutual information we have

$$\mathbb{I}(\boldsymbol{\tau} ; \mathbf{X}_{i,\mathbf{k}}\mathbf{Y}_{i,\mathbf{k}} \mid \mathbf{W}\mathbf{i}\mathbf{k}) \leq \frac{1}{\ell(m-1)} \cdot \mathbb{I}(\boldsymbol{\tau} ; \mathbf{X}_{-\mathbf{J}}\mathbf{Y}_{-\mathbf{J}} \mid \mathbf{W}) \leq o(\epsilon).$$

For convenience let  $\mathbf{j} := \mathbf{J}_i$ , let  $\mathbf{h} := \{\mathbf{j}, \mathbf{k}\}$ , let  $\mathbf{W}_{i,\mathbf{h}}$  be the restriction of  $\mathbf{W}$  to the 2 coordinates in  $\{\mathbf{i}\} \times \mathbf{h}$ , and let  $\mathbf{W}_{-\mathbf{i},\mathbf{h}}$  be the restriction of  $\mathbf{W}$  to the remaining  $\ell \times m - 2$  coordinates. There must exist outcomes  $i^*, h^*, \mathbf{W}_{-\mathbf{i}^*,\mathbf{h}^*}^*$  such that

$$\mathbb{I}(\boldsymbol{\tau} ; \mathbf{X}_{i,\mathbf{k}}\mathbf{Y}_{i,\mathbf{k}} \mid \mathbf{W}_{i,\mathbf{h}}\mathbf{k}, \mathbf{i} = i^*, \mathbf{h} = h^*, \mathbf{W}_{-\mathbf{i},\mathbf{h}} = \mathbf{W}_{-\mathbf{i}^*,\mathbf{h}^*}^*) \leq o(\epsilon). \quad (1)$$

Note that given this  $i^*, h^*, \mathbf{W}_{-\mathbf{i}^*,\mathbf{h}^*}^*$ , the remaining conditioning variables  $\mathbf{W}_{i,\mathbf{h}}\mathbf{k}$  have 36 possible outcomes: 2 choices for  $\mathbf{k}$  (it could be either element of  $h^*$ , and  $\mathbf{j}$  is the other), 3 choices for  $\mathbf{W}_{i,\mathbf{j}}$ , and 6 choices for  $\mathbf{W}_{i,\mathbf{k}}$ .

We rephrase the situation by considering a protocol  $\Pi^*$  that interprets its input as  $X_{i^*,h^*}, Y_{i^*,h^*}$ , uses private coins to sample  $X_{-i^*,h^*}, Y_{-i^*,h^*}$  uniformly from  $\mathbf{W}_{-\mathbf{i}^*,\mathbf{h}^*}^*$ , then runs the private-coin protocol  $\Pi$  on the combined input  $X, Y$ . Observe that  $\Pi^*$  is a  $(1/2 + \epsilon)$ -correct protocol for  $\text{OR}_2 \circ 3\text{EQ}^2$  since with probability 1,  $(\text{OR}_2 \circ 3\text{EQ}^2)(X_{i^*,h^*}, Y_{i^*,h^*}) = \text{TRIBES}'_{\ell,m}(X, Y)$  (as the evaluation of the 3EQ functions on  $X_{-i^*,h^*}, Y_{-i^*,h^*}$  is guaranteed to have a 1 in each of the non- $i^*$  rows, and 0’s in the non- $h^*$  columns of the  $i^*$  row). Here, we now think of the two coordinates in  $\{i^*\} \times h^*$  as being labeled 1 and 2.

For convenience, we henceforth recycle notation by letting  $\Pi$  denote the new protocol  $\Pi^*$  and letting  $(\mathbf{j}, \mathbf{k}) \in_{\mathbf{u}} \{(1, 2), (2, 1)\}$ ,  $\mathbf{W}_{\mathbf{j}} \in_{\mathbf{u}} \mathscr{W}_1$ ,  $\mathbf{W}_{\mathbf{k}} \in_{\mathbf{u}} \mathscr{W}_0$ ,  $(\mathbf{X}_1\mathbf{Y}_1) \in_{\mathbf{u}} \mathbf{W}_1$ ,  $(\mathbf{X}_2\mathbf{Y}_2) \in_{\mathbf{u}} \mathbf{W}_2$ . With respect to this recycled notation, the inequality (1) becomes

$$\mathbb{I}(\boldsymbol{\tau} ; \mathbf{X}_{\mathbf{k}}\mathbf{Y}_{\mathbf{k}} \mid \mathbf{W}_{\mathbf{k}}) \leq o(\epsilon). \quad (2)$$

The following lemma, whose proof occupies the remaining three subsections, provides the contradiction, completing the proof of Theorem 1.

► **Lemma 4.** *If (2) holds then  $\Pi$  is not a  $(1/2 + \epsilon)$ -correct protocol for  $\text{OR}_2 \circ 3\text{EQ}^2$ .*

## 2.2 Step 2: Uniformly covering a pair of gadgets

Let us set up some notation (all in reference to the private-coin protocol  $\Pi$ ). If  $x$  is an Alice input and  $Y$  is a Bob input, let  $\pi_{x,Y}$  denote the probability  $\Pi$  accepts on input  $(x, Y)$ . For a  $1 \times 2$  rectangle of inputs  $\{U\} \times \{V, W\}$  let  $\iota_{U,VW}$  denote the mutual information between the random transcript of  $\Pi$  and a uniformly random input from  $\{(U, V), (U, W)\}$ . Similarly, for a  $2 \times 1$  rectangle of inputs  $\{V, W\} \times \{U\}$  let  $\iota_{VW,U}$  denote the mutual information between the random transcript of  $\Pi$  and a uniformly random input from  $\{(V, U), (W, U)\}$ . We write  $U = U_1U_2 \in \{0, 1, 2\}^2$  and similarly for  $V$  and  $W$ .

Since in the inequality (2) there are only a constant number of possible outcomes for  $\mathbf{Wk}$ , the  $o(\epsilon)$  bound holds conditioned on each of those outcomes. Thus, (2) can be further rephrased as

$$\begin{aligned} \iota_{U,VW} \leq o(\epsilon) \text{ and } \iota_{VW,U} \leq o(\epsilon) \text{ if } U_1, V_1, W_1 \text{ are all equal and } U_2, V_2, W_2 \text{ are all distinct,} \\ \text{or } U_2, V_2, W_2 \text{ are all equal and } U_1, V_1, W_1 \text{ are all distinct.} \end{aligned} \quad (3)$$

The following lemma (illustrated in Figure 1) is proved in the remaining two subsections.

► **Lemma 5.** *For any Alice inputs  $A, B, C$  and Bob inputs  $D, E, F$ , we have*

$$\pi_{A,D} - \pi_{A,F} - \pi_{C,D} + \pi_{C,F} \leq 128(\iota_{A,DE} + \iota_{AB,D} + \iota_{C,FE} + \iota_{CB,F}).$$

**Proof of Lemma 4.** First we define a map from  $\{0, 1, 2\}^2 \times \{\pm 1\}^2$  to  $(\{0, 1, 2\}^2)^6$  that takes “data” consisting of  $t_1, t_2 \in \{0, 1, 2\}$  and  $\delta_1, \delta_2 \in \{\pm 1\}$  and maps it to a tuple of Alice inputs  $A, B, C$  and Bob inputs  $D, E, F$  defined by

$$A := t_1, (t_2 + \delta_2) \quad B := t_1, t_2 \quad C := (t_1 + \delta_1), t_2 \quad D := t_1, (t_2 - \delta_2) \quad E := t_1, t_2 \quad F := (t_1 - \delta_1), t_2$$

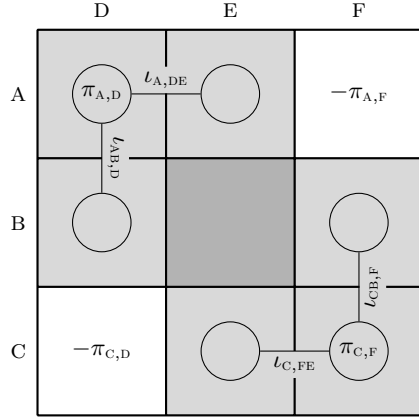
(where the addition is mod 3). For any choice of the data, we have  $(B, E) \in (3\text{EQ}^2)^{-1}(11)$  (hence the dark gray shading in Figure 1),  $(A, D), (B, D), (A, E) \in (3\text{EQ}^2)^{-1}(10)$  and  $(C, F), (C, E), (B, F) \in (3\text{EQ}^2)^{-1}(01)$  (hence the light gray shading), and  $(A, F), (C, D) \in (3\text{EQ}^2)^{-1}(00)$ .

Note that there are 36 possible choices of the data, and that  $|(3\text{EQ}^2)^{-1}(00)| = 36$  and  $|(3\text{EQ}^2)^{-1}(10)| = |(3\text{EQ}^2)^{-1}(01)| = 18$ . It is straightforward to verify the following key properties of our map.

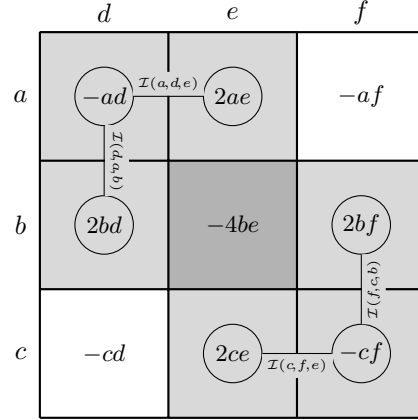
- The  $A, D$  coordinates form a 2-to-1 map onto  $(3\text{EQ}^2)^{-1}(10)$  (since  $\delta_1$  is irrelevant).
- The  $C, F$  coordinates form a 2-to-1 map onto  $(3\text{EQ}^2)^{-1}(01)$  (since  $\delta_2$  is irrelevant).
- The  $A, F$  coordinates form a 1-to-1 map onto  $(3\text{EQ}^2)^{-1}(00)$ .
- The  $C, D$  coordinates form a 1-to-1 map onto  $(3\text{EQ}^2)^{-1}(00)$ .
- The quantities  $\iota_{A,DE}, \iota_{AB,D}, \iota_{C,FE}, \iota_{CB,F}$  are always  $\leq o(\epsilon)$  by (3).

Now we have (letting the dependence of  $A, B, C, D, E, F$  on  $t_1, t_2, \delta_1, \delta_2$  be implicit)

$$\begin{aligned} & \sum_{(x,Y) \in (3\text{EQ}^2)^{-1}(10) \cup (3\text{EQ}^2)^{-1}(01)} \pi_{x,Y} - \sum_{(x,Y) \in (3\text{EQ}^2)^{-1}(00)} \pi_{x,Y} \\ &= \frac{1}{2} \sum_{t_1, t_2, \delta_1, \delta_2} (\pi_{A,D} - \pi_{A,F} - \pi_{C,D} + \pi_{C,F}) \\ &\leq \frac{1}{2} \sum_{t_1, t_2, \delta_1, \delta_2} 128(\iota_{A,DE} + \iota_{AB,D} + \iota_{C,FE} + \iota_{CB,F}) \\ &\leq \frac{1}{2} \cdot 36 \cdot 128 \cdot 4 \cdot o(\epsilon) \\ &= o(\epsilon) \end{aligned}$$



■ **Figure 1** Illustration for Lemma 5.



■ **Figure 2** Illustration for Lemma 8.

where the second line is by the first four key properties of our map, the third line is by Lemma 5, and the fourth line is by the last key property. Hence  $\Pi$  cannot be  $(1/2 + \epsilon)$ -correct for  $\text{OR}_2 \circ 3\text{EQ}^2$  since otherwise the first line would be at least  $36 \cdot (1/2 + \epsilon) - 36 \cdot (1/2 - \epsilon) = 72\epsilon$ . ◀

### 2.3 Step 3: Relating information and probabilities for inputs

We first set up some notation. For numbers  $u, v, w \in [0, 1]$ , define  $\mathcal{I}(u, v, w) := u(v - w)^2 / (v + w)$  (with the convention that  $0/0 = 0$ ). For an input  $(x, Y)$  and a transcript  $\tau$ , we let the numbers  $\tau_x, \tau_Y \in [0, 1]$  be such that  $\mathbb{P}[\Pi(x, Y) \text{ has transcript } \tau] = \tau_x \cdot \tau_Y$  (where  $\tau_x$  does not depend on  $Y$ , and  $\tau_Y$  does not depend on  $x$ ). Note that  $\pi_{x,Y} = \sum_{\text{accepting } \tau} \tau_x \cdot \tau_Y$ .

The following fact was also used in [3]; we provide a proof for completeness.

► **Lemma 6.** *For any rectangle  $\{U\} \times \{V, W\}$  we have  $\iota_{U,V,W} \geq \frac{1}{4} \sum_{\tau} \mathcal{I}(\tau_U, \tau_V, \tau_W)$ . Symmetrically, for any rectangle  $\{V, W\} \times \{U\}$  we have  $\iota_{V,W,U} \geq \frac{1}{4} \sum_{\tau} \mathcal{I}(\tau_U, \tau_V, \tau_W)$ .*

**Proof.** Assume the random variable  $\mathbf{Y} \in_{\mathcal{U}} \{V, W\}$  is jointly distributed with  $\tau$  (the random variable representing the transcript). Note that  $\mathbb{P}[\tau = \tau] = \frac{1}{2} \tau_U (\tau_V + \tau_W)$  and that  $\Delta((\mathbf{Y} | \tau = \tau), \mathbf{Y}) = \frac{1}{2} - \min(\tau_V, \tau_W) / (\tau_V + \tau_W) = \frac{1}{2} |\tau_V - \tau_W| / (\tau_V + \tau_W)$ . Then we have

$$\begin{aligned} \iota_{U,V,W} &:= \mathbb{I}(\tau ; \mathbf{Y}) \\ &= \mathbb{E}_{\tau \sim \tau} \mathbb{D}((\mathbf{Y} | \tau = \tau) \| \mathbf{Y}) \\ &\geq \sum_{\tau} \mathbb{P}[\tau = \tau] \cdot 2\Delta((\mathbf{Y} | \tau = \tau), \mathbf{Y})^2 \\ &= \sum_{\tau} \left(\frac{1}{2} \tau_U (\tau_V + \tau_W)\right) \cdot 2\left(\frac{1}{2} (\tau_V - \tau_W) / (\tau_V + \tau_W)\right)^2 \\ &= \frac{1}{4} \sum_{\tau} \tau_U (\tau_V - \tau_W)^2 / (\tau_V + \tau_W) \end{aligned}$$

where the second line is a general fact, and the third line is by Pinsker's inequality. ◀

Intuitively, Lemma 6 means  $\mathcal{I}(\tau_U, \tau_V, \tau_W)$  lower bounds the “contribution” of  $\tau$  to the information cost. Now that we have related the information costs to the contributions, we need to relate the contributions to the probabilities of observing individual transcripts. The following two lemmas allow us to do this.

► **Lemma 7.** *For any four numbers  $q, r, s, t \in [0, 1]$ , we have*

$$-qs + qt + rs - rt \leq 2(\mathcal{I}(q, s, t) + \mathcal{I}(s, q, r)).$$

► **Lemma 8.** *For any six numbers  $a, b, c, d, e, f \in [0, 1]$ , we have*

$$-ad + 2ae - af + 2bd - 4be + 2bf - cd + 2ce - cf \leq 32(\mathcal{I}(a, d, e) + \mathcal{I}(d, a, b) + \mathcal{I}(c, f, e) + \mathcal{I}(f, c, b)).$$

Lemma 7 is from [3]. Lemma 8 (illustrated in Figure 2) is more involved and constitutes one of the key technical novelties in our proof of Theorem 1. For example, one insight is in finding the proper list of coefficients on the left side of the inequality in Lemma 8, to simultaneously make the lemma true and enable it to be used in our proof approach for Lemma 5.

The proof of Lemma 7 in [3] proceeds by clearing denominators and then decomposing the difference between the right and left sides into a sum of parts, such that the (weighted) AM–GM inequality implies each part is nonnegative. A priori, it is conceivable the same approach could work for Lemma 8; however, the problem of finding an appropriate decomposition can be expressed as a linear program feasibility question, and with the help of an LP solver we found that this approach actually does not work for Lemma 8 (even with 32 replaced by other constants). To get around this, we begin by giving a significantly different proof of Lemma 7,<sup>4</sup> which we *are* able to generalize to prove Lemma 8. We provide our proofs of both lemmas in the remaining subsection, where we also give some intuition.

For now we complete the proof of Lemma 5. Here we employ another key idea (beyond the proof structure of [3]): The corresponding part of the argument in [3] finishes by simply summing Lemma 7 over accepting transcripts, but this approach does not work in our context. We also need to take into account the rejecting transcripts and the fact that the acceptance and rejection probabilities sum to 1, in order to orchestrate all the necessary cancellations.

**Proof of Lemma 5.** We have

$$\begin{aligned} & -\pi_{A,D} + 2\pi_{A,E} - \pi_{A,F} + 2\pi_{B,D} - 4\pi_{B,E} + 2\pi_{B,F} - \pi_{C,D} + 2\pi_{C,E} - \pi_{C,F} \\ &= \sum_{\text{accepting } \tau} (-\tau_A \tau_D + 2\tau_A \tau_E - \tau_A \tau_F + 2\tau_B \tau_D - 4\tau_B \tau_E + 2\tau_B \tau_F - \tau_C \tau_D + 2\tau_C \tau_E - \tau_C \tau_F) \\ &\leq 32 \sum_{\text{accepting } \tau} (\mathcal{I}(\tau_A, \tau_D, \tau_E) + \mathcal{I}(\tau_D, \tau_A, \tau_B) + \mathcal{I}(\tau_C, \tau_F, \tau_E) + \mathcal{I}(\tau_F, \tau_C, \tau_B)). \end{aligned} \quad (4)$$

by Lemma 8 with  $(a, b, c, d, e, f) = (\tau_A, \tau_B, \tau_C, \tau_D, \tau_E, \tau_F)$ . We also have

$$\begin{aligned} 2(\pi_{A,D} - \pi_{A,E} - \pi_{B,D} + \pi_{B,E}) &= 2(-(1 - \pi_{A,D}) + (1 - \pi_{A,E}) + (1 - \pi_{B,D}) - (1 - \pi_{B,E})) \\ &= 2 \sum_{\text{rejecting } \tau} (-\tau_A \tau_D + \tau_A \tau_E + \tau_B \tau_D - \tau_B \tau_E) \\ &\leq 4 \sum_{\text{rejecting } \tau} (\mathcal{I}(\tau_A, \tau_D, \tau_E) + \mathcal{I}(\tau_D, \tau_A, \tau_B)) \end{aligned} \quad (5)$$

by Lemma 7 with  $(q, r, s, t) = (\tau_A, \tau_B, \tau_D, \tau_E)$ . Similarly,

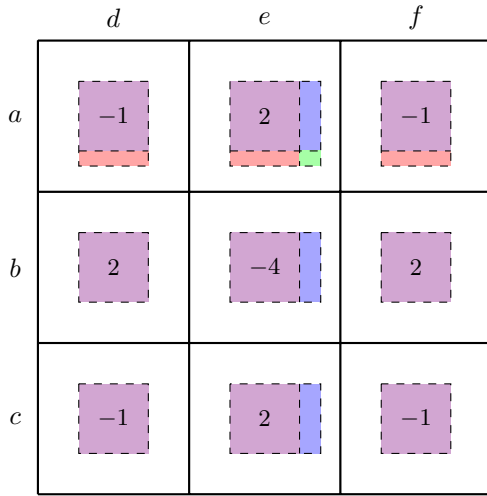
$$2(\pi_{C,F} - \pi_{C,E} - \pi_{B,F} + \pi_{B,E}) \leq 4 \sum_{\text{rejecting } \tau} (\mathcal{I}(\tau_C, \tau_F, \tau_E) + \mathcal{I}(\tau_F, \tau_C, \tau_B)) \quad (6)$$

by Lemma 7 with  $(q, r, s, t) = (\tau_C, \tau_B, \tau_F, \tau_E)$ . Summing the inequalities (4), (5), (6) yields

$$\begin{aligned} \pi_{A,D} - \pi_{A,F} - \pi_{C,D} + \pi_{C,F} &\leq 32 \sum_{\tau} (\mathcal{I}(\tau_A, \tau_D, \tau_E) + \mathcal{I}(\tau_D, \tau_A, \tau_B) + \mathcal{I}(\tau_C, \tau_F, \tau_E) + \mathcal{I}(\tau_F, \tau_C, \tau_B)) \\ &\leq 128(\iota_{A,DE} + \iota_{AB,D} + \iota_{C,FE} + \iota_{CB,F}) \end{aligned}$$

by Lemma 6. ◀

<sup>4</sup> In fact, properly balancing the calculations in our proof of Lemma 7 shows that the factor of 2 can be improved to the golden ratio  $\phi \approx 1.618$ , which does not seem to follow from the proof in [3].



■ **Figure 3** Intuition for Lemma 8.

### 2.4 Step 4: Relating information and probabilities for transcripts

We first give some intuition for why the inequality in Lemma 8 is true. Suppose for some small  $\delta, \epsilon > 0$  we have  $a = 1/2 + \delta$ ,  $e = 1/2 + \epsilon$ , and  $b = c = d = f = 1/2$ , as illustrated in Figure 3. (Although this is just a specific example, the phenomenon it illustrates turns out to hold in general.)

The left side of the inequality is the linear combination of the areas of the 9 rectangles, with coefficients as indicated in the figure. The purple regions are congruent and hence cancel out since the coefficients sum to 0. The red regions are congruent and hence cancel out since the coefficients in the top row sum to 0. The blue regions are congruent and hence cancel out since the coefficients in the middle column sum to 0. Thus the left side is  $2\delta\epsilon$  since only the green region contributes.

Regarding the four terms on the right side of the inequality, the first and third are  $\Theta(\epsilon^2)$ , the second is  $\Theta(\delta^2)$ , and the fourth is 0. Hence left side =  $\Theta(\delta\epsilon) \leq \Theta(\epsilon^2 + \delta^2)$  = right side. The point is that the right side only has terms that are quadratic in  $\delta, \epsilon$ , while the left side has “higher-order” terms (at least linear in  $\delta, \epsilon$ ) but those higher-order terms miraculously cancel out leaving only quadratic terms. The key property for the cancellation is that in every row and every column, the coefficients sum to 0.<sup>5</sup>

We proceed to our formal proofs of Lemma 7 and Lemma 8. To avoid division-by-0 technicalities, we assume the relevant quantities are infinitesimally perturbed so none are 0.

**Proof of Lemma 7.** Define

$$\mathcal{L} := -qs + qt + rs - rt = (q - r)(t - s)$$

to be the left side of the inequality in the statement of Lemma 7, and define

$$\mathcal{R} := \mathcal{I}(q, s, t) + \mathcal{I}(s, q, r) = \frac{q}{t + s}(t - s)^2 + \frac{s}{q + r}(q - r)^2$$

to be the right side except for the factor of 2. The goal is to show that  $\mathcal{R} \geq \mathcal{L}/2$ . If  $q \geq r$  and  $s \geq t$ , or if  $r \geq q$  and  $t \geq s$ , then  $\mathcal{L} \leq 0 \leq \mathcal{R}$ , so we are done in these cases. Now consider

<sup>5</sup> We have not attempted to verify whether an analogue of Lemma 8 holds for every such list of coefficients.



the case that  $q \geq r$  and  $t \geq s$ . (The remaining case, that  $r \geq q$  and  $s \geq t$ , is symmetric.) If  $t \leq 3s$  (so  $s/(t+s) \geq 1/4$ ) then since  $q/(q+r) \geq 1/2$ , the product of the two terms of  $\mathcal{R}$  is  $\geq (q-r)^2(t-s)^2/8$ , so by AM–GM,  $\mathcal{R} \geq 2(q-r)(t-s)/\sqrt{8} \geq \mathcal{L}/2$ . If  $t \geq 3s$  then  $t+s \leq 2(t-s)$  so the first term of  $\mathcal{R}$  is  $\geq (q/2(t-s))(t-s)^2 = q(t-s)/2 \geq \mathcal{L}/2$ . ◀

**Proof of Lemma 8.** Define

$$\mathcal{L} := -ad + 2ae - af + 2bd - 4be + 2bf - cd + 2ce - cf = (a - 2b + c)(-d + 2e - f)$$

to be the left side of the inequality in the statement of Lemma 8, and define

$$\begin{aligned} \mathcal{R} &:= \mathcal{I}(a, d, e) + \mathcal{I}(d, a, b) + \mathcal{I}(c, f, e) + \mathcal{I}(f, c, b) \\ &= \frac{a}{e+d}(e-d)^2 + \frac{d}{a+b}(a-b)^2 + \frac{c}{e+f}(e-f)^2 + \frac{f}{c+b}(c-b)^2 \end{aligned}$$

to be the right side except for the factor of 32. The goal is to show that  $\mathcal{R} \geq \mathcal{L}/32$ . If  $a+c \geq 2b$  and  $d+f \geq 2e$ , or if  $a+c \leq 2b$  and  $d+f \leq 2e$ , then  $\mathcal{L} \leq 0 \leq \mathcal{R}$ , so we are done in these cases. Now consider the case that  $a+c \geq 2b$  and  $d+f \leq 2e$ . (The remaining case, that  $a+c \leq 2b$  and  $d+f \geq 2e$ , is symmetric.) We consider four subcases; the first two are just like our argument for Lemma 7, but the other two are a bit more complicated.

**$c \leq a$  and  $d \leq f$ :** Then  $\mathcal{L} \leq 4(a-b)(e-d)$ . If  $e \leq 3d$  (so  $d/(e+d) \geq 1/4$ ) then since  $a/(a+b) \geq 1/2$  (because  $b \leq a$  follows from  $a+c \geq 2b$  and  $c \leq a$ ), the product of the first two terms of  $\mathcal{R}$  is  $\geq (a-b)^2(e-d)^2/8$ , so by AM–GM, the sum of these two terms is  $\geq 2(a-b)(e-d)/\sqrt{8} \geq \mathcal{L}/6$ . If  $e \geq 3d$  then  $e+d \leq 2(e-d)$  so the first term of  $\mathcal{R}$  is  $\geq (a/2(e-d))(e-d)^2 = a(e-d)/2 \geq (a-b)(e-d)/2 \geq \mathcal{L}/8$ .

**$a \leq c$  and  $f \leq d$ :** Then  $\mathcal{L} \leq 4(c-b)(e-f)$ . If  $e \leq 3f$  (so  $f/(e+f) \geq 1/4$ ) then since  $c/(c+b) \geq 1/2$  (because  $b \leq c$  follows from  $a+c \geq 2b$  and  $a \leq c$ ), the product of the last two terms of  $\mathcal{R}$  is  $\geq (c-b)^2(e-f)^2/8$ , so by AM–GM, the sum of these two terms is  $\geq 2(c-b)(e-f)/\sqrt{8} \geq \mathcal{L}/6$ . If  $e \geq 3f$  then  $e+f \leq 2(e-f)$  so the third term of  $\mathcal{R}$  is  $\geq (c/2(e-f))(e-f)^2 = c(e-f)/2 \geq (c-b)(e-f)/2 \geq \mathcal{L}/8$ .

**$a \leq c$  and  $d \leq f$ :** Then  $\mathcal{L} \leq 4(c-b)(e-d)$ . If  $e \leq 2f$  (so  $f/(e+d) \geq 1/3$ ) and  $c \leq 5a$  (so  $a/(c+b) \geq 1/10$ ) then the product of the first and last terms of  $\mathcal{R}$  is  $\geq (c-b)^2(e-d)^2/30$ , so by AM–GM, the sum of these two terms is  $\geq 2(c-b)(e-d)/\sqrt{30} \geq \mathcal{L}/12$ . If  $e \leq 2f$  and  $c \geq 5a$  then  $f \geq (e-d)/2$  and  $c+b \leq 4(c-b)$  (because  $6c \geq 5c+5a \geq 10b$ ) so the last term of  $\mathcal{R}$  is  $\geq (f/4(c-b))(c-b)^2 = f(c-b)/4 \geq (c-b)(e-d)/8 \geq \mathcal{L}/32$ . If  $e \geq 2f$  then  $e+f \leq 3(e-f)$  and  $e-f \geq e/2 \geq (e-d)/2$  so the third term of  $\mathcal{R}$  is  $\geq (c/3(e-f))(e-f)^2 = c(e-f)/3 \geq c(e-d)/6 \geq (c-b)(e-d)/6 \geq \mathcal{L}/24$ .

**$c \leq a$  and  $f \leq d$ :** Then  $\mathcal{L} \leq 4(a-b)(e-f)$ . If  $e \leq 2d$  (so  $d/(e+f) \geq 1/3$ ) and  $a \leq 5c$  (so  $c/(a+b) \geq 1/10$ ) then the product of the middle two terms of  $\mathcal{R}$  is  $\geq (a-b)^2(e-f)^2/30$ , so by AM–GM, the sum of these two terms is  $\geq 2(a-b)(e-f)/\sqrt{30} \geq \mathcal{L}/12$ . If  $e \leq 2d$  and  $a \geq 5c$  then  $d \geq (e-f)/2$  and  $a+b \leq 4(a-b)$  (because  $6a \geq 5a+5c \geq 10b$ ) so the second term of  $\mathcal{R}$  is  $\geq (d/4(a-b))(a-b)^2 = d(a-b)/4 \geq (a-b)(e-f)/8 \geq \mathcal{L}/32$ . If  $e \geq 2d$  then  $e+d \leq 3(e-d)$  and  $e-d \geq e/2 \geq (e-f)/2$  so the first term of  $\mathcal{R}$  is  $\geq (a/3(e-d))(e-d)^2 = a(e-d)/3 \geq a(e-f)/6 \geq (a-b)(e-f)/6 \geq \mathcal{L}/24$ . ◀

### 3 Query Lower Bound for Tribes

The upper bound for Theorem 2 was shown in Section 1.2; we now prove the matching lower bound.

Suppose for contradiction there is a randomized decision tree, which is a distribution  $\mathcal{T}$  over deterministic decision trees that always make at most  $\sqrt{\epsilon}m/2$  queries, and which accepts 0-inputs with probability at most  $1/2 - \epsilon$  and 1-inputs with probability at least  $1/2 + \epsilon$ . Consider the following pair of distributions  $(D_0, D_1)$  over 0-inputs and 1-inputs respectively: To sample from  $D_0$ , pick  $i \in_{\text{u}} \{1, 2\}, j \in_{\text{u}} [m], k \in_{\text{u}} [m]$  independently and set  $z_{i,j} = z_{i,k} = 1$  (and the rest of the bits to 0). To sample from  $D_1$ , pick  $j \in_{\text{u}} [m], k \in_{\text{u}} [m]$  independently and set  $z_{1,j} = z_{2,k} = 1$  (and the rest of the bits to 0).

We claim that for an arbitrary  $T$  in the support of  $\mathcal{T}$ , for each  $r \in \{0, 1, 2\}$ , letting  $A_r$  be the set of  $z$ 's such that  $T(z)$  accepts after having read exactly  $r$  1's, we have  $\mathbb{P}_{D_1}[A_r] - \mathbb{P}_{D_0}[A_r] \leq \epsilon/4$ . This yields the following contradiction:

$$\begin{aligned}
2\epsilon &= (1/2 + \epsilon) - (1/2 - \epsilon) \\
&\leq \mathbb{E}_{z \sim D_1} [\mathbb{P}_{T \sim \mathcal{T}}[\mathbf{T}(z) \text{ accepts}]] - \mathbb{E}_{z \sim D_0} [\mathbb{P}_{T \sim \mathcal{T}}[\mathbf{T}(z) \text{ accepts}]] \\
&= \mathbb{E}_{T \sim \mathcal{T}} \left[ \mathbb{P}_{z \sim D_1}[\mathbf{T}(z) \text{ accepts}] - \mathbb{P}_{z \sim D_0}[\mathbf{T}(z) \text{ accepts}] \right] \\
&= \mathbb{E}_{T \sim \mathcal{T}} \left[ \sum_r (\mathbb{P}_{D_1}[A_r] - \mathbb{P}_{D_0}[A_r]) \right] \\
&\leq \epsilon/4 + \epsilon/4 + \epsilon/4
\end{aligned}$$

(where the dependence of  $A_r$  on  $T$  is implicit on the fourth line). To prove the claim, we first set up some notation. Consider the execution of  $T$  when it reads only 0's until it halts. Let  $S_i \subseteq [m]$  ( $i \in \{1, 2\}$ ) be the coordinates of  $z_i$  queried on this execution, and let  $\delta_i := |S_i|/m$ ; note that  $\delta_1 + \delta_2 \leq \sqrt{\epsilon}/2$ . For each  $q \in [|S_1| + |S_2|]$ , let

- $B^q$  be the set of  $z$ 's that cause  $T$  to read  $q - 1$  0's then a 1,
- $i^q \in \{1, 2\}, h^q \in [m]$  be such that  $z_{i^q, h^q}$  is the location of that 1,
- $C^q \subseteq B^q$  be the set of  $z$ 's that cause  $T$  to read  $q - 1$  0's, then a 1, then only 0's until it halts,
- $S_i^q \subseteq [m]$  ( $i \in \{1, 2\}$ ) be the coordinates of  $z_i$  queried on the execution corresponding to  $C^q$ ,
- $\delta_i^q := |S_i^q|/m$  ( $i \in \{1, 2\}$ ); note that  $\delta_1^q + \delta_2^q \leq \sqrt{\epsilon}/2$ .

**Case  $r = 0$ :** If the execution that reads only 0's rejects then  $\mathbb{P}_{D_1}[A_0] = \mathbb{P}_{D_0}[A_0] = 0$ ; otherwise

$$\mathbb{P}_{D_1}[A_0] - \mathbb{P}_{D_0}[A_0] = (1 - \delta_1)(1 - \delta_2) - \frac{1}{2}(1 - \delta_1)^2 - \frac{1}{2}(1 - \delta_2)^2 = \delta_1\delta_2 - \frac{1}{2}(\delta_1^2 + \delta_2^2) \leq \epsilon/4.$$

**Case  $r = 1$ :** For each  $q$ , assuming for convenience that  $i^q = 1$ , we have

$$\mathbb{P}_{D_1}[C^q] = \mathbb{P}[j = h^q \text{ and } k \notin S_2^q] = (1 - \delta_2^q)/m \leq 1/m$$

and

$$\begin{aligned}
\mathbb{P}_{D_0}[C^q] &\geq \mathbb{P}[i = 1] \cdot \mathbb{P}[(j = h^q \text{ and } k \notin S_1^q) \text{ or } (k = h^q \text{ and } j \notin S_1^q)] \\
&= \frac{1}{2} \cdot 2 \cdot (1 - \delta_1^q)/m \\
&\geq (1 - \sqrt{\epsilon}/2)/m
\end{aligned}$$

and so  $\mathbb{P}_{D_1}[C^q] - \mathbb{P}_{D_0}[C^q] \leq \sqrt{\epsilon}/(2m)$ . Letting  $Q \subseteq [|S_1| + |S_2|]$  be those  $q$ 's for which the execution corresponding to  $C^q$  accepts, and noting that  $A_1 = \bigcup_{q \in Q} C^q$ , we have

$$\mathbb{P}_{D_1}[A_1] - \mathbb{P}_{D_0}[A_1] = \sum_{q \in Q} (\mathbb{P}_{D_1}[C^q] - \mathbb{P}_{D_0}[C^q]) \leq (\sqrt{\epsilon}m/2) \cdot \sqrt{\epsilon}/(2m) = \epsilon/4.$$

**Case  $r = 2$ :** We have

$$\mathbb{P}_{\mathbf{z} \sim D_1}[T(\mathbf{z}) \text{ reads at least one } 1] = \mathbb{P}[\mathbf{j} \in S_1 \text{ or } \mathbf{k} \in S_2] \leq \delta_1 + \delta_2 \leq \sqrt{\epsilon}/2.$$

For each  $q$ , assuming for convenience that  $i^q = 1$ , we have

$$\mathbb{P}_{\mathbf{z} \sim D_1}[T(\mathbf{z}) \text{ reads two } 1\text{'s} \mid \mathbf{z} \in B^q] = \mathbb{P}_{\mathbf{z} \sim D_1}[\mathbf{k} \in S_2^q \mid \mathbf{z} \in B^q] \leq \delta_2^q \leq \sqrt{\epsilon}/2$$

(the middle inequality may not be an equality, since prior to reading the first 1,  $T$  may have read some 0's in  $\mathbf{z}_2$ ). Hence

$$\begin{aligned} & \mathbb{P}_{D_1}[A_2] - \mathbb{P}_{D_0}[A_2] \\ & \leq \mathbb{P}_{\mathbf{z} \sim D_1}[T(\mathbf{z}) \text{ reads two } 1\text{'s}] \\ & = \mathbb{P}_{\mathbf{z} \sim D_1}[T(\mathbf{z}) \text{ reads at least one } 1] \\ & \quad \cdot \mathbb{P}_{\mathbf{z} \sim D_1}[T(\mathbf{z}) \text{ reads two } 1\text{'s} \mid T(\mathbf{z}) \text{ reads at least one } 1] \\ & \leq (\sqrt{\epsilon}/2) \cdot (\sqrt{\epsilon}/2) \\ & = \epsilon/4. \end{aligned}$$

## 4 Which One is the 1-Input?

We prove Theorem 3 and related results in this section. We state and apply the key lemmas in Section 4.1, and we prove them in Section 4.2. In the full version, we describe some ways to reinterpret Theorem 3, and we discuss some related questions.

### 4.1 Overview

Let us first review some definitions.

**Correctness:** We say  $\Pi$  is  $p$ -correct if for each  $(X, Y)$  in the domain of  $F$ , we have  $\mathbb{P}[\Pi(X, Y) = F(X, Y)] \geq p$  over the randomness of  $\Pi$ . For a distribution  $D$  over the domain of  $F$ , we say  $\Pi$  is  $(p, D)$ -correct if  $\mathbb{P}[\Pi(\mathbf{X}, \mathbf{Y}) = F(\mathbf{X}, \mathbf{Y})] \geq p$  over both the randomness of  $\Pi$  and  $\mathbf{X}\mathbf{Y} \sim D$ .

**Efficiency:** We let  $CC(\Pi)$  denote the worst-case communication cost of  $\Pi$ . Letting  $D'$  be a distribution over the set of all possible inputs to  $\Pi$  (which is a superset of the domain of  $F$ ), define  $IC^{D'}(\Pi) := \mathbb{I}(\tau; \mathbf{X} \mid \mathbf{Y}\mathbf{R}^{\text{pub}}) + \mathbb{I}(\tau; \mathbf{Y} \mid \mathbf{X}\mathbf{R}^{\text{pub}})$  to be the internal information cost with respect to  $\mathbf{X}\mathbf{Y} \sim D'$  (where  $\tau$  denotes the random transcript and  $\mathbf{R}^{\text{pub}}$  denotes the public coins)<sup>6</sup>.

<sup>6</sup> This notation is somewhat different than in Section 2.1, where we found it more convenient to let  $\tau$  denote the concatenation of the communication transcript and the public coins.

**Complexity:** We can define the following complexity measures. (Note that in this notation, the subscripts are related to correctness and the superscripts are related to efficiency.)

$$\begin{aligned} R_p(F) &:= \min_{p\text{-correct } \Pi} CC(\Pi) \\ R_{p,D}(F) &:= \min_{(p,D)\text{-correct } \Pi} CC(\Pi) \\ I_p^{D'}(F) &:= \inf_{p\text{-correct } \Pi} IC^{D'}(\Pi) \\ I_{p,D}^{D'}(F) &:= \inf_{(p,D)\text{-correct } \Pi} IC^{D'}(\Pi) \end{aligned}$$

► **Lemma 9.** For every  $F$  and balanced distribution  $D = \frac{1}{2}D_0 + \frac{1}{2}D_1$  on the domain of  $F$ , we have  $I_{1/2+\epsilon/2,D}^{D_0}(F) \leq R_{1/\ell+\epsilon}(\text{WHICH}_\ell \circ F^\ell)/\ell$ .

► **Lemma 10.** For every  $F$  we have  $R_{1/2+\epsilon/4}(F) \leq R_{1/\ell+\epsilon}(\text{WHICH}_\ell \circ F^\ell)$ .

We provide the (very similar) proofs of these two lemmas in Section 4.2. The key idea is that if we embed a random 1-input of  $F$  into a random coordinate and fill the other  $\ell - 1$  coordinates with random 0-inputs of  $F$ , then the protocol for  $\text{WHICH}_\ell \circ F^\ell$  will find the embedded 1-input with advantage  $\epsilon$ , whereas if we embed a random 0-input in the same way then the protocol cannot achieve any advantage since the coordinate of the embedding becomes independent of the  $\ell$ -tuple of 0-inputs given to the protocol. For Lemma 9 we use a direct sum property for information to get the factor  $\ell$  decrease in cost; for Lemma 10 we do not get a decrease since there is no available analogous direct sum property for communication.

**Proof of Theorem 3.** The upper bound was shown in Section 1.3. Let  $F := \text{UNAMBIG-OR}_m \circ 3\text{EQ}^m$ . As noted in Section 1.4, it suffices to prove the lower bound for  $\text{WHICH}_\ell \circ F^\ell$  instead of  $\text{WHICH}_\ell \circ \text{UNAMBIG-INTER}_m^\ell$ . For  $b \in \{0, 1\}$  let  $D_b$  be the uniform distribution over  $F^{-1}(b)$ , and let  $D := \frac{1}{2}D_0 + \frac{1}{2}D_1$ . It was shown in [3] that  $I_{1/2+\epsilon,D}^{D_0}(F) \geq \Omega(\epsilon m)$ ; <sup>7</sup> the result was not stated in this way in that paper, but careful inspection of the proof yields it. <sup>8</sup> Then  $R_{1/\ell+\epsilon}(\text{WHICH}_\ell \circ F^\ell) \geq \Omega(\epsilon \ell m)$  follows immediately from this and Lemma 9. ◀

Note that for any communication complexity class  $\mathcal{C}$ , if  $F \in \mathcal{C}$  then  $\text{WHICH}_2 \circ F^2 \in \mathcal{C} \cap \text{co}\mathcal{C}$ . Hence for  $\ell = 2$  and  $\epsilon$  a positive constant, Lemma 10 implies that if  $\mathcal{C} \not\subseteq \text{BPP}$  then  $\mathcal{C} \cap \text{co}\mathcal{C} \not\subseteq \text{BPP}$ . In particular, taking  $F = \text{UNAMBIG-INTER}$  (and  $\mathcal{C} = \text{UP}$ ), we have a simple proof of a result of [18, Theorem 2 of the arXiv version], using as a black box the fact that  $F \notin \text{BPP}$ .

## 4.2 Proofs

**Proof of Lemma 9.** Consider an arbitrary  $(1/\ell + \epsilon)$ -correct protocol  $\Pi$  for  $\text{WHICH}_\ell \circ F^\ell$ . Define a probability space with the following random variables:  $\mathbf{i} \in_{\text{u}} [\ell]$ ,  $\mathbf{XY}$  is an input to  $\Pi$  such that  $\mathbf{X}_i \mathbf{Y}_i \sim D$  and  $\mathbf{X}_j \mathbf{Y}_j \sim D_0$  for  $j \in [\ell] \setminus \{i\}$  (with the  $\ell$  coordinates independent conditioned on  $\mathbf{i}$ ),  $\boldsymbol{\tau}$  is the communication transcript of  $\Pi$ , and  $\mathbf{R}^{\text{pub}}, \mathbf{R}_A^{\text{priv}}, \mathbf{R}_B^{\text{priv}}$  are the public, Alice’s private, and Bob’s private coins, respectively. Let  $\Pi'$  be the following protocol with input interpreted as  $\mathbf{X}_i \mathbf{Y}_i$ .

<sup>7</sup> The simplified proof of the main conclusion  $R_{1/2+\epsilon}(\text{UNAMBIG-INTER}_m) \geq \Omega(\epsilon m)$  given in [12] does not yield the needed information complexity lower bound.

<sup>8</sup> For one thing, the write-up in [3] indicates that the information lower bound argument only works for protocols that have been “smoothed” in some sense, but actually this assumption is not necessary.

Publicly sample  $\mathbf{i}$ ,  $\mathbf{X}_{1,\dots,i-1}$ ,  $\mathbf{Y}_{i+1,\dots,\ell}$ , and  $\mathbf{R}^{\text{pub}}$   
 Alice privately samples  $\mathbf{X}_{i+1,\dots,\ell}$  (conditioned on the outcome of  $\mathbf{Y}_{i+1,\dots,\ell}$ ) and  $\mathbf{R}_A^{\text{priv}}$   
 Bob privately samples  $\mathbf{Y}_{1,\dots,i-1}$  (conditioned on the outcome of  $\mathbf{X}_{1,\dots,i-1}$ ) and  $\mathbf{R}_B^{\text{priv}}$   
 Run  $\Pi$  on the combined input  $\mathbf{XY}$  with coins  $\mathbf{R}^{\text{pub}}$ ,  $\mathbf{R}_A^{\text{priv}}$ ,  $\mathbf{R}_B^{\text{priv}}$   
 If  $\Pi$  outputs  $\mathbf{i}$  then output 1, otherwise output 0

For a bit  $b$ , let  $E_b$  denote the event that  $F(\mathbf{X}_i, \mathbf{Y}_i) = b$ . We have

$$\begin{aligned}
 IC^{D_0}(\Pi') &:= \mathbb{I}(\boldsymbol{\tau} ; \mathbf{X}_i \mid \mathbf{Y}_i, \mathbf{i}, \mathbf{X}_{1,\dots,i-1}, \mathbf{Y}_{i+1,\dots,\ell}, \mathbf{R}^{\text{pub}}, E_0) \\
 &\quad + \mathbb{I}(\boldsymbol{\tau} ; \mathbf{Y}_i \mid \mathbf{X}_i, \mathbf{i}, \mathbf{X}_{1,\dots,i-1}, \mathbf{Y}_{i+1,\dots,\ell}, \mathbf{R}^{\text{pub}}, E_0) \\
 &= \frac{1}{\ell} \cdot \sum_{i=1}^{\ell} \left( \mathbb{I}(\boldsymbol{\tau} ; \mathbf{X}_i \mid \mathbf{X}_{1,\dots,i-1}, \mathbf{Y}_{i,\dots,\ell}, \mathbf{R}^{\text{pub}}, E_0) \right. \\
 &\quad \left. + \mathbb{I}(\boldsymbol{\tau} ; \mathbf{Y}_i \mid \mathbf{X}_{1,\dots,i}, \mathbf{Y}_{i+1,\dots,\ell}, \mathbf{R}^{\text{pub}}, E_0) \right) \\
 &\leq \frac{1}{\ell} \cdot IC^{D_0}(\Pi) \\
 &\leq \frac{1}{\ell} \cdot CC(\Pi)
 \end{aligned}$$

where the inequalities follow by known facts (see [3, Fact 2.3 of the ECCC Revision #1 version] and [4, Lemma 3.14 of the ECCC Revision #1 version]). We also have  $\mathbb{P}[\Pi' \text{ outputs } 1 \mid E_1] = \mathbb{P}[\Pi \text{ outputs } \mathbf{i} \mid E_1] \geq 1/\ell + \epsilon$  by the correctness of  $\Pi$  (since  $\mathbf{i} = (\text{WHICH}_\ell \circ F^\ell)(\mathbf{X}, \mathbf{Y})$  assuming  $E_1$ ). We also have  $\mathbb{P}[\Pi' \text{ outputs } 1 \mid E_0] = \mathbb{P}[\Pi \text{ outputs } \mathbf{i} \mid E_0] = 1/\ell$  since conditioned on  $E_0$ ,  $\mathbf{i}$  is independent of  $\mathbf{XY}$ . Hence over the randomness of the whole experiment, the probability  $\Pi'$  is correct is at least  $(1/2) \cdot (1/\ell + \epsilon) + (1/2) \cdot (1 - 1/\ell) = 1/2 + \epsilon/2$ . ◀

**Proof of Lemma 10.** By the minimax theorem, it suffices to show that for every distribution  $D$  over the domain of  $F$ ,  $R_{1/2+\epsilon/4,D}(F) \leq R_{1/\ell+\epsilon}(\text{WHICH}_\ell \circ F^\ell)$ . If either  $F^{-1}(0)$  or  $F^{-1}(1)$  has probability at least  $1/2 + \epsilon/4$  under  $D$ , then a protocol that outputs a constant witnesses  $R_{1/2+\epsilon/4,D}(F) = 0$ , so we may assume otherwise. For a bit  $b$ , let  $D_b$  be the distribution  $D$  conditioned on  $F^{-1}(b)$ .

Consider an arbitrary  $(1/\ell + \epsilon)$ -correct protocol  $\Pi$  for  $\text{WHICH}_\ell \circ F^\ell$ . Define a probability space with the following random variables:  $\mathbf{i} \in_u [\ell]$ ,  $\mathbf{XY}$  is an input to  $\Pi$  such that  $\mathbf{X}_i \mathbf{Y}_i \sim D$  and  $\mathbf{X}_j \mathbf{Y}_j \sim D_0$  for  $j \in [\ell] \setminus \{\mathbf{i}\}$  (with the  $\ell$  coordinates independent conditioned on  $\mathbf{i}$ ), and  $\mathbf{R}^{\text{pub}}$ ,  $\mathbf{R}_A^{\text{priv}}$ ,  $\mathbf{R}_B^{\text{priv}}$  are the public, Alice's private, and Bob's private coins, respectively. Let  $\mathbf{X}_{-i} \mathbf{Y}_{-i}$  denote  $\mathbf{XY}$  restricted to coordinates in  $[\ell] \setminus \{\mathbf{i}\}$ . Let  $\Pi'$  be the following protocol with input interpreted as  $\mathbf{X}_i \mathbf{Y}_i$ .

Publicly sample  $\mathbf{i}$ ,  $\mathbf{X}_{-i}$ ,  $\mathbf{Y}_{-i}$ , and  $\mathbf{R}^{\text{pub}}$   
 Alice and Bob privately sample  $\mathbf{R}_A^{\text{priv}}$  and  $\mathbf{R}_B^{\text{priv}}$ , respectively  
 Run  $\Pi$  on the combined input  $\mathbf{XY}$  with coins  $\mathbf{R}^{\text{pub}}$ ,  $\mathbf{R}_A^{\text{priv}}$ ,  $\mathbf{R}_B^{\text{priv}}$   
 If  $\Pi$  outputs  $\mathbf{i}$  then output 1, otherwise output 0

Note that  $CC(\Pi') \leq CC(\Pi)$ . For a bit  $b$ , let  $E_b$  denote the event that  $F(\mathbf{X}_i, \mathbf{Y}_i) = b$ . We have  $\mathbb{P}[\Pi' \text{ outputs } 1 \mid E_1] = \mathbb{P}[\Pi \text{ outputs } \mathbf{i} \mid E_1] \geq 1/\ell + \epsilon$  by the correctness of  $\Pi$  (since  $\mathbf{i} = (\text{WHICH}_\ell \circ F^\ell)(\mathbf{X}, \mathbf{Y})$  assuming  $E_1$ ). We also have  $\mathbb{P}[\Pi' \text{ outputs } 1 \mid E_0] = \mathbb{P}[\Pi \text{ outputs } \mathbf{i} \mid E_0] = 1/\ell$  since conditioned on  $E_0$ ,  $\mathbf{i}$  is independent of  $\mathbf{XY}$ . Hence over the randomness of the whole experiment, the probability  $\Pi'$  is correct is at least the minimum of  $(1/2 + \epsilon/4) \cdot (1/\ell + \epsilon) + (1/2 - \epsilon/4) \cdot (1 - 1/\ell)$  and  $(1/2 - \epsilon/4) \cdot (1/\ell + \epsilon) + (1/2 + \epsilon/4) \cdot (1 - 1/\ell)$ , both of which are at least  $1/2 + \epsilon/4$ . ◀

## References

- 1 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.
- 2 Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. 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 151–160. ACM, 2013. doi:10.1145/2488608.2488628.
- 3 Mark Braverman and Ankur Moitra. An information complexity approach to extended formulations. 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 161–170. ACM, 2013. doi:10.1145/2488608.2488629.
- 4 Mark Braverman and Anup Rao. Information equals amortized communication. *IEEE Trans. Information Theory*, 60(10):6058–6069, 2014. doi:10.1109/TIT.2014.2347282.
- 5 Joshua Brody, Amit Chakrabarti, Ranganath Kondapally, David P. Woodruff, and Grigory Yaroslavtsev. Beyond set disjointness: the communication complexity of finding the intersection. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 106–113. ACM, 2014. doi:10.1145/2611462.2611501.
- 6 Joshua Brody, Amit Chakrabarti, Ranganath Kondapally, David P. Woodruff, and Grigory Yaroslavtsev. Certifying equality with limited interaction. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, volume 28 of *LIPICs*, pages 545–581. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPICs.APPROX-RANDOM.2014.545.
- 7 Amit Chakrabarti and Oded Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM J. Comput.*, 41(5):1299–1317, 2012. doi:10.1137/120861072.
- 8 Arkadev Chattopadhyay and Sagnik Mukhopadhyay. Tribes is hard in the message passing model. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPICs*, pages 224–237. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPICs.STACS.2015.224.
- 9 Mika Göös and T. S. Jayram. A composition theorem for conical juntas. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 5:1–5:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.5.
- 10 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 257–266. ACM, 2015. doi:10.1145/2746539.2746596.
- 11 Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. 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 86:1–86:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.ICALP.2016.86.

- 12 Mika Göös and Thomas Watson. Communication complexity of set-disjointness for all probabilities. *Theory of Computing*, 12(1):1–23, 2016. doi:10.4086/toc.2016.v012a009.
- 13 Prahladh Harsha and Rahul Jain. A strong direct product theorem for the tribes function via the smooth-rectangle bound. In Anil Seth and Nisheeth K. Vishnoi, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2013, December 12-14, 2013, Guwahati, India*, volume 24 of *LIPICs*, pages 141–152. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. doi:10.4230/LIPICs.FSTTCS.2013.141.
- 14 Rahul Jain and Hartmut Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, June 9-12, 2010*, pages 247–258. IEEE Computer Society, 2010. doi:10.1109/CCC.2010.31.
- 15 Rahul Jain, Hartmut Klauck, and Shengyu Zhang. Depth-independent lower bounds on the communication complexity of read-once boolean formulas. In My T. Thai and Sartaj Sahni, editors, *Computing and Combinatorics, 16th Annual International Conference, COCOON 2010, Nha Trang, Vietnam, July 19-21, 2010. Proceedings*, volume 6196 of *Lecture Notes in Computer Science*, pages 54–59. Springer, 2010. doi:10.1007/978-3-642-14031-0\_8.
- 16 T. S. Jayram, Swastik Kopparty, and Prasad Raghavendra. On the communication complexity of read-once ac<sup>0</sup> formulae. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 329–340. IEEE Computer Society, 2009. doi:10.1109/CCC.2009.39.
- 17 T. S. Jayram, Ravi Kumar, and D. Sivakumar. Two applications of information complexity. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 673–682. ACM, 2003. doi:10.1145/780542.780640.
- 18 Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *18th Annual IEEE Conference on Computational Complexity (Complexity 2003), 7-10 July 2003, Aarhus, Denmark*, pages 118–134. IEEE Computer Society, 2003. doi:10.1109/CCC.2003.1214415.
- 19 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.
- 20 Nikos Leonardos and Michael E. Saks. Lower bounds on the randomized communication complexity of read-once functions. *Computational Complexity*, 19(2):153–181, 2010. doi:10.1007/s00037-010-0292-2.
- 21 Alexander A. Razborov. On the distributional complexity of disjointness. *Theor. Comput. Sci.*, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.
- 22 Alexander A. Sherstov. The communication complexity of gap hamming distance. *Theory of Computing*, 8(1):197–208, 2012. doi:10.4086/toc.2012.v008a008.
- 23 Thomas Vidick. A concentration inequality for the overlap of a vector on a large set, with application to the communication complexity of the Gap-Hamming-Distance problem. *Chicago Journal of Theoretical Computer Science*, 2012(1):1–12, 2012. doi:10.4086/cjtcsc.2012.001.





# Algebraic Dependencies and PSPACE Algorithms in Approximative Complexity

Zeyu Guo<sup>1</sup>

Department of Computer Science & Engineering, Indian Institute of Technology Kanpur  
zguo@cse.iitk.ac.in

Nitin Saxena<sup>2</sup>

Department of Computer Science & Engineering, Indian Institute of Technology Kanpur  
nitin@cse.iitk.ac.in

Amit Sinhababu<sup>3</sup>

Department of Computer Science & Engineering, Indian Institute of Technology Kanpur  
amitks@cse.iitk.ac.in

---

## Abstract

Testing whether a set  $\mathbf{f}$  of polynomials has an algebraic dependence is a basic problem with several applications. The polynomials are given as algebraic circuits. Algebraic independence testing question is wide open over finite fields (Dvir, Gabizon, Wigderson, FOCS'07). Previously, the best complexity known was  $\text{NP}^{\#P}$  (Mittmann, Saxena, Scheiblechner, Trans.AMS'14). In this work we put the problem in  $\text{AM} \cap \text{coAM}$ . In particular, dependence testing is unlikely to be NP-hard and joins the league of problems of “intermediate” complexity, eg. graph isomorphism & integer factoring. Our proof method is algebro-geometric—estimating the size of the image/preimage of the polynomial map  $\mathbf{f}$  over the finite field. A *gap* in this size is utilized in the AM protocols.

Next, we study the open question of testing whether every annihilator of  $\mathbf{f}$  has zero constant term (Kayal, CCC'09). We give a geometric characterization using Zariski closure of the image of  $\mathbf{f}$ ; introducing a new problem called *approximate* polynomials satisfiability (APS). We show that APS is NP-hard and, using projective algebraic-geometry ideas, we put APS in PSPACE (prior best was EXPSPACE via Gröbner basis computation). As an unexpected application of this to approximative complexity theory we get—over *any* field, hitting-sets for  $\overline{\text{VP}}$  can be verified in PSPACE. This solves an open problem posed in (Mulmuley, FOCS'12, J.AMS 2017); greatly mitigating the GCT Chasm (exponentially in terms of space complexity).

**2012 ACM Subject Classification** Theory of computation → Algebraic complexity theory, Theory of computation → Complexity classes, Mathematics of computing → Computations on polynomials, Mathematics of computing → Computations in finite fields

**Keywords and phrases** algebraic dependence, Jacobian, Arthur-Merlin, approximate polynomial, satisfiability, hitting-set, border VP, finite field, PSPACE, EXPSPACE, GCT Chasm

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.10

**Acknowledgements** We thank Anurag Pandey and Sumanta Ghosh for insightful discussions on the approximate polynomials satisfiability and the hitting-set construction problems.

---

<sup>1</sup> Z.G. is funded by DST and Research I Foundation of CSE, IITK.

<sup>2</sup> N.S. thanks the funding support from DST (DST/SJF/MSA-01/2013-14).

<sup>3</sup> A.S. thanks the travel fund support from Indian Association for Research in Computing Science and ACM India.

## 1 Introduction

Algebraic dependence is a generalization of linear dependence. Polynomials  $f_1, \dots, f_m \in \mathbb{F}[x_1, \dots, x_n]$  are called *algebraically dependent* over field  $\mathbb{F}$  if there exists a nonzero polynomial (called *annihilator*)  $A(y_1, \dots, y_m) \in \mathbb{F}[y_1, \dots, y_m]$  such that  $A(f_1, \dots, f_m) = 0$ . If no  $A$  exists, then the given polynomials are called *algebraically independent* over  $\mathbb{F}$ . The *transcendence degree* ( $\text{trdeg}$ ) of a set of polynomials is the analog of rank in linear algebra. It is defined as the maximal number of algebraically independent polynomials in the set. Both algebraic dependence and linear dependence share combinatorial properties of the matroid structure [15]. The algebraic matroid examples may not be linear (esp. over  $\mathbb{F}_p$ ) [20].

The simplest examples of algebraically independent polynomials are  $x_1, \dots, x_n \in \mathbb{F}[x_1, \dots, x_n]$ . As an example of algebraically dependent polynomials, we can take  $f_1 = x$ ,  $f_2 = y$  and  $f_3 = x^2 + y^2$ . Then,  $y_1^2 + y_2^2 - y_3$  is an annihilator. The underlying field is crucial in this concept. For example, polynomials  $x + y$  and  $x^p + y^p$  are algebraically dependent over  $\mathbb{F}_p$ , but algebraically independent over  $\mathbb{Q}$ .

Thus, the following computational question  $AD(\mathbb{F})$  is natural and it is the first problem we consider in this paper: Given algebraic circuits  $f_1, \dots, f_m \in \mathbb{F}[x_1, \dots, x_n]$ , test if they are algebraically dependent. It can be solved in PSPACE using a classical result due to Perron [35, 36, 11]. Perron proved that given a set of algebraically dependent polynomials, there exists an annihilator whose degree is upper bounded by the product of the degrees of the polynomials in the set. This exponential degree bound on the annihilator is tight [22].

Computing the annihilator may be quite hard, but it turns out that the decision version is easy over zero (or large) characteristic using a classical result known as the Jacobian criterion [21, 6]. The Jacobian efficiently reduces algebraic dependence testing of  $f_1, \dots, f_m$  over  $\mathbb{F}$  to linear dependence testing of the differentials  $df_1, \dots, df_m$  over  $\mathbb{F}(x_1, \dots, x_n)$ , where we view  $df_i$  as the vector  $(\frac{\partial f_i}{\partial x_1}, \dots, \frac{\partial f_i}{\partial x_n})$ . Placing  $df_i$  as the  $i$ -th row gives us the Jacobian matrix  $J$  of  $f_1, \dots, f_m$ . If the characteristic of the field is zero (or larger than the product of the degrees  $\deg(f_i)$ ) then the  $\text{trdeg}$  equals  $\text{rank}(J)$ . It follows from [42] that, with high probability,  $\text{rank}(J)$  is equal to the rank of  $J$  evaluated at a random point in  $\mathbb{F}^n$ . This gives a simple randomized polynomial time algorithm solving  $AD(\mathbb{F})$  for certain  $\mathbb{F}$ .

For fields of positive characteristic, if the polynomials are algebraically dependent, then their Jacobian matrix is not full rank. But the converse is not true. There are infinitely many input instances (set of algebraically independent polynomials) for which Jacobian fails. The failure can be characterized by the notion of ‘inseparable extension’ [34]. For example,  $x^p, y^p$  are algebraically independent over  $\mathbb{F}_p$ , yet their Jacobian determinant vanishes. Another example is,  $\{x^{p-1}y, xy^{p-1}\}$  over  $\mathbb{F}_p$  for prime  $p > 2$ . [31] gave a criterion, called Witt-Jacobian, that works over fields of prime characteristic  $p$ ; improving the complexity of independence testing problem from PSPACE to  $\text{NP}^{\#P}$ . [34] gave another generalization of Jacobian criterion that is efficient in special cases.

Given that an efficient algorithm to tackle prime characteristic is not in close sight, one could speculate the problem to be NP-hard or even outside the polynomial hierarchy PH. In this work we show that: *For finite fields,  $AD(\mathbb{F})$  is in  $AM \cap coAM$*  (Theorem 1). This rules out the possibility of NP-hardness, under standard complexity theory assumptions [4].

**Constant term of the annihilators.** We come to the second problem *AnnAtZero* that we discuss in this paper: Testing if the constant term of *every* annihilator, of the set of algebraic circuits  $\mathbf{f} = \{f_1, \dots, f_m\}$ , is zero. Note that the annihilators of  $\mathbf{f}$  constitute an ideal of the polynomial ring  $\mathbb{F}[y_1, \dots, y_m]$ ; this ideal is principal when  $\text{trdeg}$  of  $\mathbf{f}$  is  $m - 1$  [22, Lem.7]. In this case, we can decide if the constant term of the minimal annihilator is zero in PSPACE, as the *unique* annihilator (up to scaling) can be computed in PSPACE.

If  $\text{trdeg}$  of  $\mathbf{f}$  is less than  $m - 1$ , the ideal of the annihilators of  $\mathbf{f}$  is no longer principal. Although the ideal is finitely generated, finding the generators of this ideal is computationally very hard. (Eg. using Gröbner basis techniques, we can do it in EXPSPACE [12, Sec.1.2.1].) In this case, can we decide if all the annihilators of  $\mathbf{f}$  have constant term zero? *We give two equivalent characterizations of AnnAtZero— one geometric and the other algebraic —using which we devise a PSPACE algorithm to solve it in all cases* (Theorem 2).

Interestingly, there is an algebraic-complexity application of the above algorithm. *We give a PSPACE-explicit construction of a hitting-set of the class  $\overline{\text{VP}}_{\overline{\mathbb{F}}_q}$*  (Theorem 3).  $\overline{\text{VP}}_{\overline{\mathbb{F}}_q}$  consists of  $n$ -variate degree  $d = n^{O(1)}$  polynomials, over the field  $\overline{\mathbb{F}}_q$ , that can be ‘infinitesimally approximated’ by size  $s = n^{O(1)}$  algebraic circuits. This problem is interesting as natural questions like explicit construction of the normalization map (in Noether’s Normalization Lemma NNL) reduce to the construction of a hitting-set of  $\overline{\text{VP}}$  [32]; which was previously known to be only in EXPSPACE [32, 33]. This was recently improved greatly, over the field  $\mathbb{C}$ , by [16]. Their proof technique uses real analysis and does not apply to finite fields. We need to develop purely algebraic concepts to solve the finite field case (namely through AnnAtZero), which then apply to *any* field. Moreover, we solve the problem of verifying whether an arbitrary set of points (of small size) is a hitting-set for  $\overline{\text{VP}}$ , which was not solved in [16] even over the field  $\mathbb{C}$ .

To further motivate the concept of algebraic dependence, we list a few recent problems in computer science. The first problem is about constructing an explicit randomness extractor for sources which are polynomial maps over finite fields. Using Jacobian criterion, [13, 14] solved the problem for fields with large characteristic. The second application is in the famous polynomial identity testing (PIT) problem. To efficiently design hitting-sets, for some interesting models, [6, 2, 26] constructed a family of  $\text{trdeg}$ -preserving maps. For more background and applications of algebraic dependence testing, see [34]. The annihilator has been a key concept to prove the connection between hitting-sets and lower bounds [19], and in bootstrapping ‘weak’ hitting-sets [3].

## 1.1 Our results

In this paper, we give Arthur-Merlin protocols & algorithms, with proofs using basic tools from algebraic geometry. The first theorem we prove is about  $\text{AD}(\mathbb{F}_q)$ .

► **Theorem 1.** *Algebraic dependence testing of circuits in  $\mathbb{F}_q[\mathbf{x}]$  is in  $\text{AM} \cap \text{coAM}$ .*

This result vastly improves the current best upper bound known for  $\text{AD}(\mathbb{F}_q)$ — from being ‘outside’ the polynomial hierarchy (namely  $\text{NP}^{\#\text{P}}$  [31]) to ‘lower’ than the second-level of polynomial hierarchy (namely  $\text{AM} \cap \text{coAM}$ ). This rules out the possibility of  $\text{AD}(\mathbb{F}_q)$  being NP-hard (unless polynomial hierarchy collapses to the second-level [4]). Recall that, for zero or large characteristic  $\mathbb{F}$ ,  $\text{AD}(\mathbb{F})$  is in  $\text{coRP}$  (Section 2). We conjecture such a result for  $\text{AD}(\mathbb{F}_q)$  too.

Our second result is about the problem AnnAtZero (i.e. testing whether all the annihilators of given  $\mathbf{f}$  have constant term zero). A priori it is unclear why it should have complexity better than EXPSPACE (note: ideal membership is EXPSPACE-complete [30]). Firstly, we relate to a (new) version of polynomial system satisfiability, over the algebraic closure  $\overline{\mathbb{F}}$ :

► **Problem 1** (Approximate polynomials satisfiability (APS)). *Given algebraic circuits  $f_1, \dots, f_m \in \mathbb{F}[x_1, \dots, x_n]$ , does there exist  $\beta \in \overline{\mathbb{F}}(\varepsilon)^n$  such that for all  $i$ ,  $f_i(\beta)$  is in the ideal  $\varepsilon\overline{\mathbb{F}}[\varepsilon]$ ? If yes, then we say that  $\mathbf{f} := \{f_1, \dots, f_m\}$  is in APS.*

It is easy to show: Function field  $\overline{\mathbb{F}}(\varepsilon)$  here can be equivalently replaced by *Laurent polynomials*  $\overline{\mathbb{F}}[\varepsilon, \varepsilon^{-1}]$ , or, the field  $\overline{\mathbb{F}}((\varepsilon))$  of *formal Laurent series* (use mod  $\varepsilon\overline{\mathbb{F}}[\varepsilon]$ ). A reason why these objects appear in algebraic complexity can be found in [8, Sec.5.2] & [29, Sec.5]. They help algebraize the notion of ‘infinitesimal approximation’ (in real analysis think of  $\varepsilon \rightarrow 0$  &  $1/\varepsilon \rightarrow \infty$ ). A notable computational issue involved is that the degree bound of  $\varepsilon$  required for  $\beta$  is exponential in the input size [29, Prop.3]; this may again be a “justification” for APS requiring that much space.

Classically, the *exact* version of APS has been extremely well-studied– Does there exist  $\beta \in \overline{\mathbb{F}}^n$  such that for all  $i$ ,  $f_i(\beta) = 0$ ? This is what Hilbert’s Nullstellensatz (HN) characterizes and yields an impressive PSPACE algorithm [24, 25]. Note that if system  $\mathbf{f}$  has an exact solution, then it is trivially in APS. But the converse is not true. For example,  $\{x, xy - 1\}$  is in APS, but there is no exact solution in  $\overline{\mathbb{F}}$ . To see the former, assign  $x = \varepsilon$  and  $y = 1/\varepsilon$ . Also, the instance  $\{x, x + 1\}$  is neither in APS nor has an exact solution. Finally, note that if we restrict  $\beta$  to come from  $\overline{\mathbb{F}}[\varepsilon]^n$  then APS becomes equivalent to exact satisfiability and HN applies. This can be seen by going modulo  $\varepsilon\overline{\mathbb{F}}[\varepsilon]$ , as the quotient  $\overline{\mathbb{F}}[\varepsilon]/\varepsilon\overline{\mathbb{F}}[\varepsilon]$  is  $\overline{\mathbb{F}}$ .

Coming back to AnnAtZero, we show that it is equivalent both to a geometric question and to deciding APS. This gives us, with more work, the following surprising consequence.

► **Theorem 2.** *APS is NP-hard and is in PSPACE.*

We apply this to designing hitting-sets and solving NNL (refer [32] for the background).

► **Theorem 3.** *There is a PSPACE algorithm that (given input  $n, s, r$  in unary & suitably large  $\mathbb{F}_q$ ) outputs a set, of points from  $\mathbb{F}_q^n$  of size  $\text{poly}(nsr, \log q)$ , that hits all  $n$ -variate degree- $r$  polynomials over  $\overline{\mathbb{F}}_q$  that can be infinitesimally approximated by size  $s$  circuits.*

**More applications?** The exact polynomials satisfiability question HN (over  $\overline{\mathbb{F}}$ ) is highly expressive and, naturally, most computer science problems get expressed that way. We claim that in a similar spirit, the APS question expresses those computer science problems that involve ‘infinitesimal approximation’. Since finite fields do not seem to have a natural topology allowing approximations, algebraic approximations over arbitrary fields is needed. The latter has been useful in fast matrix multiplication algorithms.

One prominent example of algebraic approximation is the concept of *border rank* of tensor polynomials (used in matrix multiplication algorithms and GCT, see [9, 27, 28]). Border rank computation of a given tensor (over  $\overline{\mathbb{F}}$ ) can easily be reduced to an APS instance and, hence, now solved in PSPACE; this matches the complexity of tensor rank itself [40]. From the point of view of Gröbner basis theory, APS is a problem that seems a priori much harder than HN. Now that both of them have a PSPACE algorithm, one may wonder whether it can be brought all the way down to NP or AM? (In fact,  $\text{HN}_{\mathbb{C}}$  is known to be in AM, conditionally under GRH [24].)

Our methods in the proof of Theorem 2 imply an interesting “degree bound” related to the (prime) ideal  $I$  of annihilators of polynomials  $\mathbf{f}$ . Namely,  $I = \sqrt{I_{\leq d}}$ , where  $I_{\leq d}$  refers to the subideal generated by degree  $\leq d$  polynomials of  $I$ ,  $d$  is the Perron-like bound  $(\max_{i \in [m]} \deg(f_i))^k$ , and  $k := \text{trdeg}(\mathbf{f})$ . This is equivalent to the geometric fact, which we prove, that the varieties defined by the two ideals  $I$  and  $I_{\leq d}$  are equal (Theorem 18). This again is an exponential improvement over what one expects to get from the general Gröbner basis methods; because, the generators of  $I$  may well have doubly-exponential degree.

The hitting-set result (Theorem 3) can be applied to compute, in PSPACE, the explicit system of parameters (esop) of the *invariant ring* of the variety  $\Delta[\det, s]$ , over  $\overline{\mathbb{F}}_q$ , with a given group action [32, Thm.4.9]. Also, we can now construct, in PSPACE, polynomials in

$\mathbb{F}_q[x_1, \dots, x_n]$  that cannot even be approximated by ‘small’ algebraic circuits. Such results were previously known only for characteristic zero fields, see [16, Thms.1.1-1.4]. Bringing this complexity down to  $\mathbb{P}$  is the longstanding problem of blackbox PIT (& lower bounds), see [38, 43, 39]. Mulmuley [33] pointed out that small hitting-sets for  $\overline{\text{VP}}$  can be designed in  $\text{EXPSPACE}$  which is a far worse complexity than that for  $\text{VP}$ . He called it the GCT Chasm. We bridge it somewhat, as the proof of Theorem 3 shows that small hitting-sets for  $\overline{\text{VP}}_{\mathbb{F}}$  can be designed in  $\text{PSPACE}$  (like those for  $\text{VP}$ ) for *any* field  $\mathbb{F}$ .

In another application, the null-cone problem defined in [10] can be seen as a special case of APS and using our algorithm, it can be solved in  $\text{PSPACE}$ . Bürgisser et al. [10] gave an exponential time algorithm for the above problem (bringing it down from  $\text{EXPSPACE}$ ).

## 1.2 Proof ideas

**Proof idea of Theorem 1.** Suppose we are given algebraic circuits  $\mathbf{f} := \{f_1, \dots, f_m\}$  computing in  $\mathbb{F}_q[x_1, \dots, x_n]$ . For the AM and coAM protocols, we consider the following system of equations over a ‘small’ extension  $\mathbb{F}_{q'}$ :

For  $b = (b_1, \dots, b_m) \in \mathbb{F}_{q'}^m$ , define the system of equations  $f_i(x_1, \dots, x_n) = b_i$ , for  $i \in [m]$ . We denote the number of solutions of the above system in  $\mathbb{F}_{q'}^n$  as  $N_b$ . Let  $f : \mathbb{F}_{q'}^n \rightarrow \mathbb{F}_{q'}^m$  be the polynomial map  $a \mapsto (f_1(a), \dots, f_m(a))$ .

*AM gap.* [Theorem 9] We establish bounds for the number  $N_{f(a)}$ , where  $a$  is a random point in  $\mathbb{F}_{q'}^n$ . If  $f_1, \dots, f_m$  are independent, we show that  $N_{f(a)}$  is relatively small. Whereas, if the polynomials are algebraically dependent then  $N_{f(a)}$  is much more.

Assume  $\mathbf{f}$  are algebraically independent. Wlog (see the full version of [34, Sec.2]) we can assume that  $m = n$  and for all  $i \in [n]$ ,  $\{x_i, f_1, \dots, f_n\}$  are algebraically dependent. The first step is to show that the zeroset defined by the system of equations, for random  $f(a)$ , has dimension  $\leq 0$  with high probability. This is proved using the Perron degree bound on the annihilator of  $\{x_i, f_1, \dots, f_n\}$ . Next, one can apply an affine version of Bezout’s theorem to upper bound  $N_{f(a)}$ . On the other hand, suppose  $\mathbf{f}$  are algebraically dependent, say with annihilator  $Q$ . Let  $\text{Im}(f) := f(\mathbb{F}_{q'}^n)$  be the image of  $f$ . Since  $Q$  vanishes on  $\text{Im}(f)$ , we know that  $\text{Im}(f)$  is relatively small, whence we deduce that  $N_{f(a)}$  is large for ‘most’  $a$ ’s.

*coAM gap.* [Theorem 12] We pick a random point  $b = (b_1, \dots, b_m) \in \mathbb{F}_{q'}^m$  and bound  $N_b$ , which is the number of solutions of the system defined above. In the dependent case, we show that  $N_b = 0$  for ‘most’  $b$ ’s. But in the independent case, we show that  $N_b \geq 1$  for ‘many’ (may be not ‘most’!)  $b$ ’s. The ideas are based on those sketched above.

The two kinds of gaps shown above are based on the set  $f^{-1}(f(\mathbf{x}))$  resp.  $\text{Im}(f)$ . Note that membership in either of these sets is testable in NP (the latter requires nondeterminism). Based on this and the gaps between the respective cardinalities, we can invoke Lemma 4 and devise the AM and coAM protocols for  $\text{AD}(\mathbb{F}_{q'})$ , which also apply to  $\text{AD}(\mathbb{F}_q)$ .

*Remark*– One advantage in our problem is that we could sample a random point in the set  $\text{Im}(f)$ . In contrast, it is not clear how to sample a random point in the zeroset  $\text{Zer}(\mathbf{f}) := \{\mathbf{x} \in \mathbb{F}_{q'}^n \mid f(\mathbf{x}) = \mathbf{0}\}$ . Thus, we manage to side-step the NP-hardness associated with most zeroset properties. Eg. computing the dimension of  $\text{Zer}(\mathbf{f})$  is NP-hard.

**Proof idea of Theorem 2.** Let algebraic circuits  $\mathbf{f} := \{f_1, \dots, f_m\}$  in  $\mathbb{F}[x_1, \dots, x_n]$  be given over a field  $\mathbb{F}$ . We want to determine if the constant term of every annihilator for  $\mathbf{f}$  is zero. Redefine the polynomial map  $f : \overline{\mathbb{F}}^n \rightarrow \overline{\mathbb{F}}^m$ ;  $a \mapsto (f_1(a), \dots, f_m(a))$ . For a subset  $S$  of an affine (resp. projective) space, write  $\overline{S}$  for its *Zariski closure* in that space, i.e. it is the smallest subset that contains  $S$  and equals the zeroset  $\text{Zer}(I)$  of some polynomial ideal  $I$ .

*APS vs AnnAtZero.* [Theorem 15] Now, we interpret the problem AnnAtZero in a geometric way through Lemma 13:

The constant term of every annihilator of  $\mathbf{f}$  is zero iff the origin point  $\mathbf{0} \in \overline{\text{Im}(f)}$ .

This has a simple proof using the ideal-variety correspondence [17]. Note that the stronger condition  $\mathbf{0} \in \text{Im}(f)$  is equivalent to the existence of a common solution to the equations  $f_i(x_1, \dots, x_n) = 0, i = 1, \dots, m$ . The latter problem (call it HN for Hilbert’s Nullstellensatz) is known to be in AM if  $\mathbb{F} = \mathbb{Q}$  and GRH is assumed [24]. However,  $\text{Im}(f)$  is not necessarily Zariski closed; equivalently, it may be strictly smaller than  $\overline{\text{Im}(f)}$ . So, we need new ideas to test  $\mathbf{0} \in \overline{\text{Im}(f)}$ .

Next, we observe that although  $\mathbf{0} \in \overline{\text{Im}(f)}$  is not equivalent to the existence of a solution  $\mathbf{x} \in \overline{\mathbb{F}}^n$  to  $f(\mathbf{x}) = \mathbf{0}$ , it *is* equivalent to the existence of an “approximate solution”  $\mathbf{x} \in \overline{\mathbb{F}}(\varepsilon)^n$ , which is an  $n$ -tuple of rational functions in a formal variable  $\varepsilon$ . The proof idea of this uses a degree bound on  $\varepsilon$  due to [29]. We called this problem APS. As AnnAtZero problem is already known to be NP-hard [22], APS is also NP-hard.

*Upper bounding APS.* We now know that: Solving APS for  $\mathbf{f}$  is equivalent to solving AnnAtZero for  $\mathbf{f}$ . AnnAtZero was previously known to be in PSPACE in the special case when the  $\text{trdeg } k$  of  $\mathbb{F}(\mathbf{f})/\mathbb{F}$  equals  $m$  or  $m - 1$ , but the general case remained open (best being EXPSPACE).

In this work we prove that AnnAtZero is in PSPACE even when  $k < m - 1$ . Our simple idea is to reduce the input to a smaller  $m = k + 1$  instance, by choosing new polynomials  $g_1, \dots, g_{k+1}$  that are random linear combinations of  $f_i$ ’s. We show that with high probability, replacing  $\{f_1, \dots, f_m\}$  by  $\{g_1, \dots, g_{k+1}\}$  preserves YES/NO instances as well as the  $\text{trdeg}$ . This gives a randomized poly-time reduction from the case  $k < m - 1$  to  $k = m - 1$  (Theorem 18). The latter has a standard PSPACE algorithm.

For notational convenience view  $\overline{\mathbb{F}}$  as the *affine line*  $\mathbb{A}$ . Define  $V := \overline{\text{Im}(f)} \subseteq \mathbb{A}^m$ . Proving that the above reduction (of  $m$ ) does preserve YES/NO instances amounts to proving the following geometric statement: If  $V$  does not contain the origin  $O \in \mathbb{A}^m$ , then with high probability, the variety  $V' := \overline{\pi(V)}$  does not contain the origin  $O' \in \mathbb{A}^{k+1}$  either, where  $\pi : \mathbb{A}^m \rightarrow \mathbb{A}^{k+1}$  is a random linear map.

As  $\pi$  is picked at random, the kernel  $W$  of  $\pi$  is a random linear subspace of  $\mathbb{A}^m$ . We have  $O' \notin \pi(V)$  whenever  $V \cap W = \emptyset$ , but this is not sufficient for proving  $O' \notin \overline{\pi(V)}$ , since  $V$  may “get arbitrarily close to  $W$ ” in  $\mathbb{A}^m$  and meet  $W$  “at infinity”. Inspired by this observation, we consider projective geometry instead of affine geometry, and prove that  $O' \notin V'$  holds as long as the projective closure of  $V$  and that of  $W$  are disjoint. The proof uses a construction of a projective subvariety– the *join* –to characterize  $\pi^{-1}(V')$ , and eventually rules out  $W \subseteq \pi^{-1}(V')$  (Lemma 19).

Moreover, we show that this holds with high probability if  $O \notin V$ : by (repeatedly) using the fact that a generic (=random) hyperplane section reduces the dimension of a variety by one.

**Proof idea of Theorem 3.** Define  $\mathbb{A} := \overline{\mathbb{F}}_q$  and assume  $wlog\ q \geq \Omega(sr^2)$  [1]. [19, Thm.4.4] showed that a hitting-set, of size  $h := O(s^2n^2 \log q)$  in  $\mathbb{F}_q^n$ , *exists* for the class of degree- $r$  polynomials, in  $\mathbb{A}[x_1, \dots, x_n]$ , that can be infinitesimally approximated by size- $s$  algebraic circuits. So, we can search over all possible subsets of size  $h$  from  $\mathbb{F}_q^n$  and ‘most’ of them are hitting-sets.

How do we certify that a candidate set  $\mathcal{H}$  is a hitting-set? The idea is to use universal circuits. A *universal circuit* has  $n$  essential variables  $\mathbf{x} = \{x_1, \dots, x_n\}$  and  $s' := O(sr^4)$  auxiliary variables  $\mathbf{y} = \{y_1, \dots, y_{s'}\}$ . We can fix the auxiliary variables, from  $\mathbb{A}(\varepsilon)$ , in such

a way so that it can output any homogeneous circuit of size- $s$ , approximating a degree- $r$  polynomial in  $\overline{\text{VP}}_{\mathbb{A}}$ . Given a universal circuit  $\Psi$ , certification of a hitting-set  $\mathcal{H}$  is based on the following observation, that follows from the definitions:

Candidate set  $\mathcal{H} = \{\mathbf{v}_1, \dots, \mathbf{v}_h\}$  is a hitting-set iff  $\forall \mathbf{y} \in \mathbb{A}(\varepsilon)^{s'}$ ,  $\Psi(\mathbf{y}, \mathbf{x}) \notin \varepsilon\mathbb{A}[\varepsilon][\mathbf{x}] \Rightarrow \exists i \in [h]$ ,  $\Psi(\mathbf{y}, \mathbf{v}_i) \notin \varepsilon\mathbb{A}[\varepsilon]$ .

Equivalently: Candidate set  $\mathcal{H} = \{\mathbf{v}_1, \dots, \mathbf{v}_h\}$  is *not* a hitting-set iff  $\exists \mathbf{y} \in \mathbb{A}(\varepsilon)^{s'}$ ,  $\Psi(\mathbf{y}, \mathbf{x}) \notin \varepsilon\mathbb{A}[\varepsilon][\mathbf{x}]$  and  $\forall i \in [h]$ ,  $\Psi(\mathbf{y}, \mathbf{v}_i) \in \varepsilon\mathbb{A}[\varepsilon]$ .

Note that this hitting-set certification is more challenging than the one against polynomials in VP; because the degree bounds for  $\varepsilon$  are exponentially high and moreover, we do not know how to frame the first ‘non-containment’ condition as an APS instance. To translate it to an APS instance, our key idea is the following.

Pick  $q \geq \Omega(s'r^2)$  so that a hitting-set exists, in  $\mathbb{F}_q^n$ , that works against polynomials approximated by the specializations of  $\Psi$ . Suppose  $\Psi(\alpha, \mathbf{x})$  is not in  $\varepsilon\mathbb{A}[\varepsilon][\mathbf{x}]$ , for some  $\alpha \in \mathbb{A}(\varepsilon)^{s'}$ . This means that we can write it as  $\sum_{-m \leq i \leq m'} \varepsilon^i g_i(\mathbf{x})$  with  $g_{-m} \neq 0$  and  $m \geq 0$ . Clearly,  $\varepsilon^m \cdot \Psi(\alpha, \mathbf{x})$  infinitesimally approximates the nonzero polynomial  $g_{-m} \in \mathbb{A}[\mathbf{x}]$ . By the conditions on  $\Psi$ , we know that  $g_{-m}$  is a homogeneous degree- $r$  polynomial (and approximative complexity  $s'$ ). Thus, by [42], there exists a  $\beta \in \mathbb{F}_q^n$  such that  $g_{-m}(\beta) =: a$  is a nonzero element in  $\mathbb{A}$ . We can normalize by this and consider  $a^{-1}\varepsilon^m \cdot \Psi(\mathbf{y}, \mathbf{x})$ , which evaluates to  $1 + \varepsilon\mathbb{A}[\varepsilon]$  at  $(\alpha, \beta)$ . Since this normalization factor only affects the auxiliary variables  $\mathbf{y}$ , we get another equivalent criterion:

Candidate set  $\mathcal{H} = \{\mathbf{v}_1, \dots, \mathbf{v}_h\}$  is *not* a hitting-set iff  $\exists \mathbf{y} \in \mathbb{A}(\varepsilon)^{s'}$  and  $\exists \mathbf{x} \in \mathbb{F}_q^n$  such that,  $\Psi(\mathbf{y}, \mathbf{x}) - 1 \in \varepsilon\mathbb{A}[\varepsilon]$  and  $\forall i \in [h]$ ,  $\Psi(\mathbf{y}, \mathbf{v}_i) \in \varepsilon\mathbb{A}[\varepsilon]$ .

We reach closer to APS, but how do we implement  $\exists \mathbf{x} \in \mathbb{F}_q^n$  (it takes exponential space)?

The idea is to rewrite it, instead using the  $(r+1)$ -th roots of unity  $Z_{r+1} \subset \mathbb{A}$ , as:  $\exists \mathbf{x} \in \mathbb{A}(\varepsilon)^n$ ,  $\forall i \in [n]$ ,  $x_i^{r+1} - 1 \in \varepsilon\mathbb{A}[\varepsilon]$ . This gives us a criterion that is an instance of APS with  $n+h+1$  input polynomials (Theorem 22). By Theorem 2 it can be done in PSPACE; finishing the proof. Moreover, this PSPACE algorithm idea is independent of the field characteristic. (Eg. it can be seen as an alternative to [16] over the complex field.)

## 2 Preliminaries

**Jacobian.** Although this work would not need it, we define the classical Jacobian: For polynomials  $\mathbf{f} = \{f_1, \dots, f_m\}$  in  $\mathbb{F}[x_1, \dots, x_n]$ , *Jacobian* is the matrix  $\mathcal{J}_{\mathbf{x}}(\mathbf{f}) := (\partial_{x_j} f_i)_{m \times n}$ , where  $\partial_{x_j} f_i := \partial f_i / \partial x_j$ .

Jacobian criterion [21, 6] states: For degree  $\leq d$  and  $\text{trdeg} \leq r$  polynomials  $\mathbf{f}$ , if  $\text{char}(\mathbb{F}) = 0$  or  $\text{char}(\mathbb{F}) > d^r$ , then  $\text{trdeg}(\mathbf{f}) = \text{rank}_{\mathbb{F}(\mathbf{x})} \mathcal{J}_{\mathbf{x}}(\mathbf{f})$ . This yields a randomized poly-time algorithm [42]. For other fields, Jacobian criterion fails due to inseparability and  $\text{AD}(\mathbb{F})$  is open.

**AM protocol.** Arthur-Merlin class AM is a randomized version of the class NP (see [4]). Arthur-Merlin protocols, introduced by Babai [5], can be considered as a special type of interactive proof system in which the randomized poly-time verifier (Arthur) and the all-powerful prover (Merlin) have only constantly many rounds of exchange. AM contains interesting problems like determining if two graphs are non-isomorphic.  $\text{AM} \cap \text{coAM}$  is the class of decision problems for which both YES and NO answers can be verified by an AM protocol. It can be thought of as the randomized version of  $\text{NP} \cap \text{coNP}$ . See [23] for a few natural algebraic problems in  $\text{AM} \cap \text{coAM}$ . If such a problem is NP-hard (even under random reductions) then polynomial hierarchy collapses to the second-level, i.e.  $\text{PH} = \Sigma_2$ .



In this work AM protocol will only be used to distinguish whether a set  $S$  is ‘small’ or ‘large’. Formally, we refer to the Goldwasser-Sipser Set Lowerbound method:

► **Lemma 4.** [4, Chap.9] *Let  $m \in \mathbb{N}$  be given in binary. Suppose  $S$  is a set whose membership can be tested in nondeterministic polynomial time and its size is promised to be either  $\leq m$  or  $\geq 2m$ . Then, the problem of deciding whether  $|S| \stackrel{?}{\geq} 2m$  is in AM.*

**Geometry.** Due to limited space we have moved the geometry preliminaries to Appendix A. One can also refer to a standard text, eg. [17, 18]. Basically, we need terms about affine (resp. projective) zerosets and the underlying Zariski topology. The latter gives a way to ‘impose’ geometry even in very discrete situations, eg. finite fields in this work.

### 3 Algebraic dependence testing: Proof of Theorem 1

Given  $f_1, \dots, f_m \in \mathbb{F}_q[x_1, \dots, x_n]$ , we want to decide if they are algebraically dependent. For this problem  $\text{AD}(\mathbb{F}_q)$  we could assume, with some preprocessing, that  $m = n$ . For,  $m > n$  means that its a YES instance. If  $m < n$  then we could apply a ‘random’ linear map on the variables to reduce them to  $m$ , preserving the YES/NO instances. Also, the  $\text{trdeg}$  does not change when we move to the algebraic closure  $\overline{\mathbb{F}}_q$ . The details can be found in [34, Lem.2.7-2.9]. So, we assume the input instance to be  $\mathbf{f} := \{f_1, \dots, f_n\}$  with nonconstant polynomials.

In the following, let  $D := \prod_{i \in [n]} \deg(f_i) > 0$  and  $D' := \max_{i \in [n]} \deg(f_i) > 0$ . Let  $d \in \mathbb{N}^+$  and  $q' = q^d$ . The value of  $d$  will be determined later. Let  $f : \mathbb{F}_{q'}^n \rightarrow \mathbb{F}_{q'}^n$  be the polynomial map  $a \mapsto (f_1(a), \dots, f_n(a))$ . For  $b = (b_1, \dots, b_n) \in \mathbb{F}_{q'}^n$ , denote by  $N_b$  the size of the preimage  $f^{-1}(b) = \{\mathbf{x} \in \mathbb{F}_{q'}^n \mid f(\mathbf{x}) = b\}$ .

Define  $\mathbb{A} := \overline{\mathbb{F}}_q$  and  $\overline{N}_b := \#\{\mathbf{x} \in \mathbb{A}^n \mid f_i(\mathbf{x}) = b_i, \text{ for all } i \in [n]\}$  which might be  $\infty$ . Let  $Q \in \mathbb{F}_q[y_1, \dots, y_n]$  be a nonzero annihilator, of minimal degree, of  $f_1, \dots, f_n$ . If it exists then  $\deg(Q) \leq D$  by Perron’s bound.

#### 3.1 AM protocol

First, we study the independent case.

► **Lemma 5 (Dim=0 preimage).** *Suppose  $\mathbf{f}$  are independent. Then  $\overline{N}_{f(a)}$  is finite for all but at most  $(nDD'/q')$ -fraction of  $a \in \mathbb{F}_{q'}^n$ .*

**Proof.** For  $i \in [n]$ , let  $G_i \in \mathbb{F}_q[z, y_1, \dots, y_n]$  be the annihilator of  $\{x_i, f_1, \dots, f_n\}$ . We have  $\deg(G_i) \leq D$  by Perron’s bound. Consider  $a \in \mathbb{F}_{q'}^n$  such that  $G'_i(z) := G_i(z, f_1(a), \dots, f_n(a)) \in \mathbb{F}_q[z]$  is a nonzero polynomial for every  $i \in [n]$ . We claim that  $\overline{N}_{f(a)}$  is finite for such  $a$ .

To see this, note that for any  $b = (b_1, \dots, b_n) \in \mathbb{A}^n$  satisfying the equations  $f_i(b) = f_i(a)$ ,  $i \in [n]$ , we have

$$0 = G_i(b_i, f_1(b), \dots, f_n(b)) = G_i(b_i, f_1(a), \dots, f_n(a)) = G'_i(b_i), \quad \forall i \in [n].$$

Hence, each  $b_i$  is a root of  $G'_i$ . It follows that  $\overline{N}_{f(a)} \leq \prod_{i \in [n]} \deg(G'_i) < \infty$ , as claimed.

It remains to prove that the number of  $a \in \mathbb{F}_{q'}^n$  satisfying  $G'_i = 0$ , for some index  $i \in [n]$ , is bounded by  $nDD'q'^{-1} \cdot q^n$ . Fix  $i \in [n]$ . Suppose  $G_i = \sum_{j=0}^{d_i} G_{i,j}z^j$ , where  $d_i := \deg_z(G_i)$  and  $G_{i,j} \in \mathbb{F}_q[y_1, \dots, y_n]$ , for  $0 \leq j \leq d_i$ . The leading coefficient  $G_{i,d_i}$  is nonzero. As  $f_1, \dots, f_n$  are algebraically independent, the polynomial  $G_{i,d_i}(f_1, \dots, f_n) \in \mathbb{F}_q[x_1, \dots, x_n]$  is



also nonzero. Its degree is  $\leq D' \deg(G_{i,d_i}) \leq D' \deg(G_i) \leq DD'$ . By [42], for all but at most  $(DD'/q')$ -fraction of  $a \in \mathbb{F}_q^n$ , we have  $G_{i,d_i}(f_1(a), \dots, f_n(a)) \neq 0$  which implies

$$G'_i(z) = G_i(z, f_1(a), \dots, f_n(a)) = \sum_{j=0}^{d_i} G_{i,j}(f_1(a), \dots, f_n(a))z^j \neq 0.$$

The claim now follows from the union bound.  $\blacktriangleleft$

We need the following affine version of Bézout's Theorem. Its proof can be found in [41, Thm.3.1].

► **Theorem 6 (Bézout's).** *Let  $g_1, \dots, g_n \in \mathbb{A}[x_1, \dots, x_n]$ . Then the number of common zeros of  $g_1, \dots, g_n$  in  $\mathbb{A}^n$  is either infinite, or at most  $\prod_{i \in [n]} \deg(g_i)$ .*

Combining Lemma 5 with Bézout's Theorem, we obtain

► **Lemma 7 (Small preimage).** *Suppose  $\mathbf{f}$  are independent. Then  $N_{f(a)} \leq D$  for all but at most  $(nDD'/q')$ -fraction of  $a \in \mathbb{F}_q^n$ .*

Next, we study the dependent case (with an annihilator  $Q$ ).

► **Lemma 8 (Large preimage).** *Suppose  $\mathbf{f}$  are dependent. Then for  $k > 0$ , we have  $N_{f(a)} > k$  for all but at most  $(kD/q')$ -fraction of  $a \in \mathbb{F}_q^n$ .*

**Proof.** Let  $\text{Im}(f) := f(\mathbb{F}_q^n)$  be the image of the map. Note that  $Q$  vanishes on all the points in  $\text{Im}(f)$ . So,  $|\text{Im}(f)| \leq Dq^{n-1}$  by [42].

Let  $B := \{b \in \text{Im}(f) : N_b \leq k\}$  be the “bad” images. We can estimate the bad domain points as,

$$\#\{a \in \mathbb{F}_q^n : N_{f(a)} \leq k\} = \#\{a \in \mathbb{F}_q^n : f(a) \in B\} \leq k|B| \leq k|\text{Im}(f)| \leq kDq^{n-1}.$$

which proves the lemma.  $\blacktriangleleft$

► **Theorem 9 (AM).** *Testing algebraic dependence of  $\mathbf{f}$  is in AM.*

**Proof.** Fix  $q' = q^d > 4nDD' + 4kD$  and  $k := 2D$ . Note that  $d$  will be polynomial in the input size. For an  $a \in \mathbb{F}_q^n$ , consider the set  $f^{-1}(f(a)) := \{\mathbf{x} \in \mathbb{F}_q^n \mid f(\mathbf{x}) = f(a)\}$ .

By Lemmas 7 & 8: When Arthur picks  $a$  randomly, with high probability,  $|f^{-1}(f(a))| = N_{f(a)}$  is more than  $2D$  in the dependent case while  $\leq D$  in the independent case. Note that an upper bound on  $\prod_{i \in [n]} \deg(f_i)$  can be deduced from the size of the input circuits for  $f_i$ 's; thus, we know  $D$ . Moreover, containment in  $f^{-1}(f(a))$  can be tested in P. Thus, by Lemma 4,  $\text{AD}(\mathbb{F}_q)$  is in AM.  $\blacktriangleleft$

### 3.2 coAM protocol

We first study the independent case.

► **Lemma 10 (Large image).** *Suppose  $\mathbf{f}$  are independent. Then  $N_b > 0$  for at least  $(D^{-1} - nD'q'^{-1})$ -fraction of  $b \in \mathbb{F}_q^n$ .*

**Proof.** Let  $S := \{a \in \mathbb{F}_q^n : N_{f(a)} \leq D\}$ . Then  $|S| \geq (1 - nDD'q'^{-1}) \cdot q^n$  by Lemma 7. As every  $b \in f(S)$  has at most  $D$  preimages in  $S$  under  $f$ , we have  $|f(S)| \geq |S|/D \geq (D^{-1} - nD'q'^{-1}) \cdot q^n$ . This proves the lemma since  $N_b > 0$  for all  $b \in f(S)$ .  $\blacktriangleleft$

Next, we study the dependent case.

## 10:10 Algebraic dependencies

► **Lemma 11** (Small image). *Suppose  $\mathbf{f}$  are dependent. Then  $N_b = 0$  for all but at most  $(D/q')$ -fraction of  $b \in \mathbb{F}_q^n$ .*

**Proof.** By definition:  $N_b > 0$  iff  $b \in \text{Im}(f) := f(\mathbb{F}_q^n)$ . It was shown in the proof of Lemma 8 that  $|\text{Im}(f)| \leq Dq'^{n-1}$ . The lemma follows. ◀

► **Theorem 12** (coAM). *Testing algebraic dependence of  $\mathbf{f}$  is in coAM.*

**Proof.** Fix  $q' = q^d > D(2D + nD')$ . Note that  $d$  will be polynomial in the input size. For  $b \in \mathbb{F}_q^n$ , consider the set  $f^{-1}(b) := \{\mathbf{x} \in \mathbb{F}_q^n \mid f(\mathbf{x}) = b\}$  of size  $N_b$ .

Define  $S := \text{Im}(f)$ . Note that:  $b \in \mathbb{F}_q^n$  has  $N_b > 0$  iff  $b \in S$ . Thus, by Lemma 10 (resp. Lemma 11),  $|S| \geq (D^{-1} - nD'q'^{-1})q'^n > 2Dq'^{n-1}$  (resp.  $|S| \leq Dq'^{n-1}$ ) when  $\mathbf{f}$  are independent (resp. dependent). Note that an upper bound on  $\prod_{i \in [n]} \deg(f_i)$  can be deduced from the size of the input circuits for  $f_i$ 's; thus, we know  $Dq'^{n-1}$ . Moreover, containment in  $S$  can be tested in NP. Thus, by Lemma 4,  $\text{AD}(\mathbb{F}_q)$  is in coAM. ◀

**Proof of Theorem 1.** The statement immediately follows from Theorems 9 & 12. ◀

## 4 Approximate polynomials satisfiability: Proof of Theorem 2

Theorem 2 is proved in two parts. First, we show that APS is equivalent to AnnAtZero problem; which means that it is NP-hard [22]. Next, we utilize the beautiful underlying geometry to devise a PSPACE algorithm.

### 4.1 APS is equivalent to AnnAtZero

Let  $\mathbb{A}$  be the algebraic closure of  $\mathbb{F}$ . Note that for the given polynomials  $\mathbf{f} := \{f_1, \dots, f_m\}$  in  $\mathbb{F}[\mathbf{x}]$ , there is an annihilator over  $\mathbb{F}$  with nonzero constant term iff there is an annihilator over  $\mathbb{A}$  with nonzero constant term. This is because if  $Q$  is an annihilator over  $\mathbb{A}$  with nonzero constant term, wlog 1, then by basic linear algebra, the linear system defined by the equation  $Q(\mathbf{f}) = 0$ , in terms of the unknown coefficients of  $Q$ , would also have a solution in  $\mathbb{F}$ . Thus, there is an annihilator over  $\mathbb{F}$  with constant term 1. This proves that it suffices to solve AnnAtZero over the algebraically closed field  $\mathbb{A}$ . This provides us with a better geometry.

Write  $f : \mathbb{A}^n \rightarrow \mathbb{A}^m$  for the polynomial map sending a point  $x = (x_1, \dots, x_n) \in \mathbb{A}^n$  to  $(f_1(x), \dots, f_m(x)) \in \mathbb{A}^m$ . For a subset  $S$  of an affine or projective space, write  $\overline{S}$  for its Zariski closure in that space. We will use  $O$  to denote the origin  $\mathbf{0}$  of an affine space.

The following lemma reinterprets APS in a geometric way.

► **Lemma 13** ( $O$  in the closure). *The constant term of every annihilator for  $\mathbf{f}$  is zero iff  $O \in \overline{\text{Im}(f)}$ .*

**Proof.** Note that:  $Q \in \mathbb{A}[Y_1, \dots, Y_m]$  vanishes on  $\text{Im}(f)$  iff  $Q(\mathbf{f})$  vanishes on  $\mathbb{A}^n$ , which holds iff  $Q(\mathbf{f}) = 0$ , i.e.,  $Q$  is an annihilator for  $\mathbf{f}$ . So  $\overline{\text{Im}(f)} = V(I)$ , where the ideal  $I \subseteq \mathbb{A}[Y_1, \dots, Y_m]$  consists of the annihilators for  $\mathbf{f}$ . Also note that  $\{O\} = V(\mathfrak{m})$ , where  $\mathfrak{m}$  is the maximal ideal  $\langle Y_1, \dots, Y_m \rangle$ .

Let us study the condition  $O \in \overline{\text{Im}(f)}$ . By the ideal-variety correspondence,  $\{O\} = V(\mathfrak{m}) \subseteq \overline{\text{Im}(f)} = V(I)$  is equivalent to  $I \subseteq \mathfrak{m}$ , i.e.,  $Q \bmod \mathfrak{m} = 0$  for  $Q \in I$ . But  $Q \bmod \mathfrak{m}$  is just the constant term of the annihilator  $Q$ . Hence, we have the equivalence. ◀

As an interesting corner case, the above lemma proves that whenever  $\mathbf{f}$  are algebraically independent, we have  $\mathbb{A}^m = \overline{\text{Im}(f)}$ . Eg.  $f_1 = X_1$  and  $f_2 = X_1X_2 - 1$ . Even in the dependent cases,  $\text{Im}(f)$  is not necessarily closed in the Zariski topology.

► **Example 14.** Let  $n = 2$ ,  $m = 3$ . Consider  $f_1 = f_2 = X_1$  and  $f_3 = X_1X_2 - 1$ . The annihilators are multiples of  $(Y_1 - Y_2)$ , which means by Lemma 13 that  $O \in \overline{\text{Im}(f)}$ . But there is no solution to  $f_1 = f_2 = f_3 = 0$ , i.e.  $O \notin \text{Im}(f)$ .

**Approximation.** Although  $O \in \overline{\text{Im}(f)}$  is not equivalent to the existence of a solution  $x \in \mathbb{A}^n$  to  $f_i = 0$ ,  $i \in [m]$ , it is equivalent to the existence of an “approximate solution”  $x \in \mathbb{A}[\varepsilon, \varepsilon^{-1}]^n$ , which is a tuple of Laurent polynomials in a formal variable  $\varepsilon$ . The formal statement is as follows. Wlog we assume  $\mathbf{f}$  to be  $m$  nonconstant polynomials.

► **Theorem 15 (Approx. wrt  $\varepsilon$ ).**  $O \in \overline{\text{Im}(f)}$  iff there exists  $x = (x_1, \dots, x_n) \in \mathbb{A}(\varepsilon)^n$  such that  $f_i(x) \in \varepsilon\mathbb{A}[\varepsilon]$ , for all  $i \in [m]$ . Moreover, when such  $x$  exists, it may be chosen such that

$$x_i \in \varepsilon^{-D}\mathbb{A}[\varepsilon] \cap \varepsilon^{D'}\mathbb{A}[\varepsilon^{-1}] = \left\{ \sum_{j=-D}^{D'} c_j \varepsilon^j : c_j \in \mathbb{A} \right\}, \quad i \in [n],$$

where  $D := \prod_{i \in [m]} \deg(f_i) > 0$  and  $D' := (\max_{i \in [m]} \deg(f_i)) \cdot D > 0$ .

The proof of Theorem 15 is almost the same as that in [29]. First, we recall a tool to reduce the domain from a variety to a curve, proven in [29].

► **Lemma 16.** [29, Prop.1] Let  $V \subseteq \mathbb{A}^n$ ,  $W \subseteq \mathbb{A}^m$  be affine varieties,  $\varphi : V \rightarrow W$  dominant, and  $t \in W \setminus \varphi(V)$ . Then there exists a curve  $C \subseteq \mathbb{A}^n$  such that  $t \in \overline{\varphi(C)}$  and  $\deg(C) \leq \deg(\Gamma_\varphi)$ , where  $\Gamma_\varphi$  denotes the graph of  $\varphi$  embedded in  $\mathbb{A}^n \times \mathbb{A}^m$ .

Next, [29] essentially shows that in the case of a curve one can approximate the preimage of  $f$  by using a *single* formal variable  $\varepsilon$  and working in  $\mathbb{A}(\varepsilon)$ .

► **Lemma 17.** [29, Cor. of Prop.3] Let  $C \subseteq \mathbb{A}^n$  be an affine curve. Let  $f : C \rightarrow \mathbb{A}^m$  be a morphism sending  $x \in C$  to  $(f_1(x), \dots, f_m(x)) \in \mathbb{A}^m$ , where  $f_1, \dots, f_m \in \mathbb{A}[X_1, \dots, X_n]$ . Let  $t = (t_1, \dots, t_m) \in \overline{f(C)}$ . Then there exists  $p_1, \dots, p_n \in \varepsilon^{-\deg(C)}\mathbb{A}[[\varepsilon]]$  such that  $f_i(p_1, \dots, p_n) - t_i \in \varepsilon\mathbb{A}[[\varepsilon]]$ , for all  $i \in [m]$ .

Finally, we can use the above two lemmas to prove the connection of APS with  $O \in \overline{\text{Im}(f)}$ , and hence with  $\text{AnnAtZero}$  (by Lemma 13).

**Proof of Theorem 15.** First assume that an  $x$ , satisfying the conditions in Theorem 15, exists. Pick such an  $x$ . If  $\mathbf{f}$  are algebraically independent then by Lemma 13 we have that  $\mathbb{A}^m = \overline{\text{Im}(f)}$  and we are done. So, assume that there is a nonzero annihilator  $Q$  for  $\mathbf{f}$ . We have  $Q(f_1(x), \dots, f_m(x)) = 0 \in \varepsilon\mathbb{A}[\varepsilon]$ . On the other hand, as  $f_i(x) \in \varepsilon\mathbb{A}[\varepsilon]$ , for all  $i \in [m]$ ; we deduce that  $Q(f_1(x), \dots, f_m(x)) \bmod \varepsilon\mathbb{A}[\varepsilon]$  is  $Q(\mathbf{0})$ , which is the constant term of  $Q$ . So it equals zero. By Lemma 13, we have  $O \in \overline{\text{Im}(f)}$  and again we are done.

Conversely, assume  $O \in \overline{\text{Im}(f)}$  and we will prove that  $x$  exists. If  $O \in \text{Im}(f)$ , then we can choose  $x \in \mathbb{A}^n$  and we are done. So assume  $O \in \overline{\text{Im}(f)} \setminus \text{Im}(f)$ . Regard  $f$  as a dominant morphism from  $\mathbb{A}^n$  to  $\overline{\text{Im}(f)}$ . Its graph  $\Gamma_f$  is cut out in  $\mathbb{A}^n \times \mathbb{A}^m$  by  $Y_i - f_i(X_1, \dots, X_n)$ ,  $i \in [m]$ . So  $\deg(\Gamma_f) \leq \prod_{i=1}^m \deg(f_i) = D$  by Bézout’s Theorem.

By Lemma 16, there exists a curve  $C \subseteq \mathbb{A}^n$  such that  $O \in \overline{f(C)}$  and  $\deg(C) \leq \deg(\Gamma_f) \leq D$ . Pick such a curve  $C$ . Apply Lemma 17 to  $C$ ,  $f|_C$  and  $O$ , and let  $p_1, \dots, p_n \in \varepsilon^{-\deg(C)}\mathbb{A}[[\varepsilon]] \subseteq \varepsilon^{-D}\mathbb{A}[[\varepsilon]]$  be as given by the lemma. Then  $f_i(p_1, \dots, p_n) \in \varepsilon\mathbb{A}[[\varepsilon]]$ , for all  $i \in [m]$ .

For  $i \in [n]$ , let  $x_i$  be the Laurent polynomial obtained from  $p_i$  by truncating the terms of degree greater than  $D'$ . When evaluating  $f_1, \dots, f_m$ , at  $(p_1, \dots, p_n)$ , such truncation does not affect the coefficient of  $\varepsilon^k$  for  $k \leq 0$  by the choice of  $D'$ . So  $f_i(x_1, \dots, x_n) \in \varepsilon\mathbb{A}[\varepsilon]$ , for all  $i \in [m]$ . ◀

*Remark*– The lower bound  $-D = -\prod_{i=1}^m \deg(f_i)$  for the least degree of  $x_i$  in  $\varepsilon$  can be achieved up to a factor of  $1 + o(1)$ . Consider the polynomials  $f_1 = f_2 = X_1$ ,  $f_3 = X_1^{d-1}X_2 - 1$ , and  $f_i = X_{i-2}^d - X_{i-1}$  for  $i = 4, \dots, m$ , where  $m = n + 1$ . Then we are forced to choose  $x_1 \in \varepsilon\mathbb{A}[\varepsilon]$  and  $x_i \in \varepsilon^{-(d-1)d^{i-2}} \cdot \mathbb{A}[\varepsilon^{-1}]$ , for  $i = 2, \dots, n$ . So the least degree of  $x_n$  in  $\varepsilon$  is at most  $-(d-1)d^{n-2}$ , while  $-D = -d^{n-1}$ .

## 4.2 Putting APS in PSPACE

Owing to the exponential upper bound on the precision (= degree wrt  $\varepsilon$ ) shown in Theorem 15, one expects to solve APS in EXPSPACE only. Surprisingly, in this section, we give a PSPACE algorithm. This we do by reducing the general AnnAtZero instance to a very special instance, that is easy to solve.

Let  $\mathbb{A}$  be the algebraic closure of the field  $\mathbb{F}$ . Let  $f_1, \dots, f_m \in \mathbb{F}[X_1, \dots, X_n]$  be given. Denote by  $k$  the trdeg of  $\mathbb{F}(f_1, \dots, f_m)/\mathbb{F}$ . Computing  $k$  can be done in PSPACE using linear algebra [36, 11]. We assume  $k < m - 1$ , since the cases  $k = m - 1$  and  $k = m$  are again easy. In the case  $k = m$ , the input instances are always in APS since  $\overline{\text{Im}(f)} = \mathbb{A}^m$ . And in the case  $k = m - 1$ , the ideal of the annihilators is a principal ideal, and hence has a unique generator (up to scaling). The degree of this generator is at most  $\prod_{i=1}^m \deg(f_i)$ . Thus checking whether it has a nonzero constant term can be solved in PSPACE by solving an exponential sized linear system of equations using [11].

We reduce the number of polynomials from  $m$  to  $k + 1$  as follows: Fix a finite subset  $S \subseteq \mathbb{F}$ , and choose  $c_{i,j} \in S$  at random for  $i \in [k + 1]$  and  $j \in [m]$ . For this to work, we need a large enough  $S$  and  $\mathbb{F}$ . For  $i \in [k + 1]$ , let  $g_i := \sum_{j=1}^m c_{i,j} f_j$ .

Let  $\delta := (k + 1)(\max_{i \in [m]} \deg(f_i))^k / |S|$ . Our algorithm is immediate once we prove the following claim.

- **Theorem 18** (Random reduction). *It holds, with probability  $\geq (1 - \delta)$ , that*
- (1) *the transcendence degree of  $\mathbb{F}(g_1, \dots, g_{k+1})/\mathbb{F}$  equals  $k$ , and*
  - (2) *the constant term of every annihilator for  $g_1, \dots, g_{k+1}$  is zero iff the constant term of every annihilator for  $f_1, \dots, f_m$  is zero.*

First, we reformulate the two items of Theorem 18 in a geometric way, and later we will analyze the error probability.

For  $d \in \mathbb{N}$ , denote by  $\mathbb{A}^d$  (resp.  $\mathbb{P}^d$ ) the  $d$ -dimensional affine space (resp. projective space) over  $\mathbb{A} := \overline{\mathbb{F}}$ . Let  $f : \mathbb{A}^n \rightarrow \mathbb{A}^m$  (resp.  $g : \mathbb{A}^n \rightarrow \mathbb{A}^{k+1}$ ) be the polynomial map sending  $x$  to  $(f_1(x), \dots, f_m(x))$  (resp.  $(g_1(x), \dots, g_{k+1}(x))$ ). Let  $O$  and  $O'$  be the origin of  $\mathbb{A}^m$  and that of  $\mathbb{A}^{k+1}$  respectively. Define the affine varieties  $V := \overline{\text{Im}(f)} \subseteq \mathbb{A}^m$  and  $V' := \overline{\text{Im}(g)} \subseteq \mathbb{A}^{k+1}$ . Then  $\dim V = \text{trdeg } \mathbf{f} = k$ .

Let  $\pi : \mathbb{A}^m \rightarrow \mathbb{A}^{k+1}$  be the linear map sending  $(x_1, \dots, x_m)$  to  $(y_1, \dots, y_{k+1})$  where  $y_i = \sum_{j=1}^m c_{i,j} x_j$ . Then  $g = \pi \circ f$  and  $V' = \overline{\pi(V)}$ .<sup>4</sup> Now (1) of Theorem 18 is equivalent to  $\dim V' = k$ , and (2) is equivalent to  $O' \in V'$  iff  $O \in V$ .

$$\begin{array}{ccccc}
 \mathbb{A}^n & \xrightarrow{f} & V = \overline{\text{Im}(f)} & \xrightarrow{\subseteq} & \mathbb{A}^m \\
 & \searrow g & \downarrow \pi|_V & & \downarrow \pi \\
 & & V' = \overline{\text{Im}(g)} & \xrightarrow{\subseteq} & \mathbb{A}^{k+1}
 \end{array}$$

<sup>4</sup> To see  $V' \supseteq \overline{\pi(V)}$ , note that  $\pi^{-1}(V')$  contains  $\text{Im}(f)$  and is closed, and hence contains  $V = \overline{\text{Im}(f)}$ .

We will give sufficient conditions of (1) and (2) in terms of incidence properties. Note that  $O \in V$  implies  $O' \in V'$ , since  $\pi(O) = O'$ . Now suppose  $O \notin V$ . Let  $W := \pi^{-1}(O')$ , which is a linear subspace of  $\mathbb{A}^m$ . Then  $O' \notin \pi(V)$  iff  $V \cap W = \emptyset$ . However,  $V \cap W = \emptyset$  does not imply  $O' \notin V'$ , as  $V$  may “get infinitesimally close to  $W$ ” without actually meeting  $W$ , so that  $O' \in \overline{\pi(V)} = V'$ . See Example 23 in the appendix.

To overcome this problem, we consider projective geometry instead of affine geometry. Suppose  $\mathbb{A}^m$  have coordinates  $X_1, \dots, X_m$  and  $\mathbb{P}^m$  have homogeneous coordinates  $X_0, \dots, X_m$ . Regard  $\mathbb{A}^m$  as a dense open subset of  $\mathbb{P}^m$  via  $(x_1, \dots, x_m) \mapsto (1, x_1, \dots, x_m)$ . Then  $H := \mathbb{P}^m \setminus \mathbb{A}^m \cong \mathbb{P}^{m-1}$  is the *hyperplane at infinity*, defined by  $X_0 = 0$ . Denote by  $V_c$  (resp.  $W_c$ ) the *projective closure* of  $V$  (resp.  $W$ ) in  $\mathbb{P}^m$ . Then  $V = V_c \cap \mathbb{A}^m$ . Let  $W_H := W_c \cap H$ , which is a projective subspace of  $H$ .

For distinct points  $P, Q \in \mathbb{P}^m$ , write  $\overline{PQ}$  for the projective line passing through them.

► **Lemma 19** (Sufficient conditions). *We have:*

- (1)  $\dim V' = k$ , if  $V_c \cap W_H = \emptyset$ , and
- (2)  $O' \notin V'$ , if  $V_c \cap W_c = \emptyset$ .

**Proof.** (1): Assume  $\dim V' < k$ . Choose  $P \in \pi(V)$ . The dimension of  $\pi^{-1}(P) \cap V$  is at least  $\dim V - \dim V' \geq 1$  [17, Thm.11.12]. Denote by  $Y$  and  $Z$  the projective closure of  $\pi^{-1}(P)$  and that of  $\pi^{-1}(P) \cap V$  in  $\mathbb{P}^m$  respectively. Then  $Z \subseteq Y \cap V_c$ . As  $\dim Z = \dim \pi^{-1}(P) \cap V \geq 1$  and  $\dim H = m - 1$ , we have  $Z \cap H \neq \emptyset$  [17, Prop.11.4].

As  $\pi$  is a linear map,  $\pi^{-1}(P) = Y \cap \mathbb{A}^m$  is a translate of  $\pi^{-1}(O') = W = W_c \cap \mathbb{A}^m$ . It is well known that two projective subspaces  $W_1, W_2 \not\subseteq H$  have the same intersection with  $H$  iff  $W_1 \cap \mathbb{A}^m$  and  $W_2 \cap \mathbb{A}^m$  are translates of each other.<sup>5</sup> So,  $Y \cap H = W_c \cap H = W_H$ . Therefore,  $V_c \cap W_H = V_c \cap Y \cap H \supseteq Z \cap H \neq \emptyset$ .

(2): Assume to the contrary that  $V_c \cap W_c = \emptyset$  but  $O' \in V'$ . We will derive a contradiction. As  $W_H \subseteq W_c$ , we have  $V_c \cap W_H = \emptyset$  and hence  $\dim V' = k$  by (1).

Denote by  $J(V_c, W_H)$  the *join* of  $V_c$  and  $W_H$ , which is defined to be the union of the projective lines  $\overline{PQ}$ , where  $P \in V_c$  and  $Q \in W_H$ . It is known that  $J(V_c, W_H)$ , as the join of two *disjoint* projective subvarieties, is again a projective subvariety of  $\mathbb{P}^m$  [17, Example 6.17]. Consider  $P \in V_c$  and  $Q \in W_H$ . If  $P \in H$ , the line  $\overline{PQ}$  lies in  $H$  and does not meet  $\mathbb{A}^m$ . Now suppose  $P \in V_c \setminus H = V$ . Then  $\overline{PQ}$  meets  $\overline{OQ}$  at the point  $Q$ . So  $\overline{PQ} \cap \mathbb{A}^m$  is a translate of  $\overline{OQ} \cap \mathbb{A}^m \subseteq W_c \cap \mathbb{A}^m = W$ .

Conversely, let  $P \in V$ . Let  $W_P$  denote the unique translate of  $W$  containing  $P$ . Let  $\ell_P$  be an affine line contained in  $W_P$  and passing through  $P$  (note that  $W_P$  is the union of such lines). Then  $\ell_P$  is a translate of an affine line  $\ell \subseteq W$ . As  $\ell_P$  and  $\ell$  are translates of each other, their projective closures intersect  $H$  at the same point  $Q$ . We have  $Q \in \ell \cap H \subseteq W_H$ . So  $\ell_P = \overline{PQ} \cap \mathbb{A}^m \subseteq J(V_c, W_H) \cap \mathbb{A}^m$ . We conclude that

$$J(V_c, W_H) \cap \mathbb{A}^m = \bigcup_{P \in V} W_P. \tag{1}$$

We claim that  $J(V_c, W_H) \cap \mathbb{A}^m = \pi^{-1}(V')$ . As  $\pi$  is a linear map, Equation (1) implies  $J(V_c, W_H) \cap \mathbb{A}^m \subseteq \pi^{-1}(V')$ . We prove the other direction by comparing dimensions. It is known that for two *disjoint* projective subvarieties  $V_1$  and  $V_2$ ,  $\dim J(V_1, V_2) = \dim V_1 + \dim V_2 + 1$  [17, Prop.11.37-Ex.11.38]. Therefore,

$$\dim J(V_c, W_H) = \dim V_c + \dim W_H + 1 = \dim V + \dim W = k + \dim W.$$

<sup>5</sup> Indeed,  $W_i \cap \mathbb{A}^m$  is defined by linear equations  $\sum_{j=1}^m a_{j,t} X_j + a_{0,t} = 0$  iff  $W_i \cap H$  is defined by homogeneous linear equations  $X_0 = 0$  and  $\sum_{j=1}^m a_{j,t} X_j = 0$ . So the constant terms  $a_{0,t}$  do not matter.

10:14 Algebraic dependencies

So,  $\dim J(V_c, W_H) \cap \mathbb{A}^m = k + \dim W$ . On the other hand, we have  $\pi^{-1}(V') \cong V' \times W$ . So  $\dim \pi^{-1}(V') = \dim V' + \dim W = k + \dim W$ . Now  $J(V_c, W_H) \cap \mathbb{A}^m$  and  $\pi^{-1}(V')$  are (irreducible) affine varieties of the same dimension, and one is contained in the other. So they must be equal. This proves the claim.

As  $O' \in V'$ , we have  $W = \pi^{-1}(O') \subseteq \pi^{-1}(V') = \bigcup_{P \in V} W_P$ . So  $W_P = W$  for some  $P \in V$ , since  $W$  is a linear space. But then  $P \in V \cap W_P = V \cap W \subseteq V_c \cap W_c$ , contradicting the assumption  $V_c \cap W_c = \emptyset$ . ◀

► Remark. The converse of Lemma 19 (Condition 2) is false; see Example 24 in the appendix.

**Error probability.** It remains to bound the probability of failure of the conditions  $V_c \cap W_H = \emptyset$  and (in the case  $O \notin V$ )  $V_c \cap W_c = \emptyset$  in Lemma 19. We need the following lemma.

► **Lemma 20** (Cut by hyperplanes). *Let  $V \subseteq \mathbb{P}^m$  be a projective subvariety of dimension  $r$  and degree  $d$ . Let  $r' \geq r + 1$ . Choose  $c_{i,j} \in S$  at random, for  $i \in [r']$  and  $0 \leq j \leq m$ . Let  $W \subseteq \mathbb{P}^m$  be the projective subspace cut out by the equations  $\sum_{j=0}^m c_{i,j} X_j = 0$ ,  $i = 1, \dots, r'$ , where  $X_0, \dots, X_m$  are homogeneous coordinates of  $\mathbb{P}^m$ . Then  $V \cap W = \emptyset$  holds with probability at least  $1 - (r + 1)d/|S|$ .*

**Proof.** For  $i \in [r']$ , let  $H_i \subseteq \mathbb{P}^m$  be the hyperplane defined by  $\sum_{j=0}^m c_{i,j} X_j = 0$ . By ignoring  $H_i$  for  $i > r + 1$ , we may assume  $r' = r + 1$ . Let  $V_0 := V$  and  $V_i := V_{i-1} \cap H_i$  for  $i \in [r']$ . It suffices to show that  $\dim V_i = \dim V_{i-1} - 1$  holds with probability at least  $1 - d/|S|$ , for each  $i \in [r']$  (the dimension of the empty set is  $-1$  by convention).

Fix  $i \in [r']$  and  $c_{i',j}$ , for  $i' \in [i - 1]$  and  $0 \leq j \leq m$ . So  $V_{i-1}$  is also fixed. Note that  $V_{i-1} \neq \emptyset$  since taking a hyperplane section reduces the dimension by at most one. If  $\dim V_i \neq \dim V_{i-1} - 1$ , then  $\dim V_i = \dim V_{i-1}$ , and  $H_i$  contains some irreducible component of  $V_{i-1}$  [17, Exercise 11.6]. Let  $Y$  be an irreducible component of  $V_{i-1}$ , and fix a point  $P \in Y$ . Then  $Y \subseteq H_i$  only if  $P \in H_i$ , which holds only if  $c_{i,0}, \dots, c_{i,m}$  satisfy a nonzero linear equation determined by  $P$ . This occurs with probability at most  $1/|S|$  (eg. by fixing all but one  $c_{i,j}$ ). We also have  $\deg(V_{i-1}) \leq \deg(V) \leq d$ , and hence the number of irreducible components of  $V_{i-1}$  is bounded by  $d$ . By the union bound,  $H_i$  contains an irreducible component of  $V_{i-1}$  with probability at most  $d/|S|$ . ◀

**Proof of Theorem 18.** As mentioned above, Theorem 18 is equivalent to showing that, with probability at least  $1 - \delta$ : (1)  $\dim V' = k$ , and (2)  $O' \in V'$  iff  $O \in V$ . Note that  $W_c$  is cut out in  $\mathbb{P}^m$  by the linear equations  $\sum_{j=1}^m c_{i,j} X_j = 0$ ,  $i = 1, \dots, k + 1$ . So  $W_H$  is cut out in  $H \cong \mathbb{P}^{m-1}$  (corresponding to  $X_0 = 0$ ) by the linear equations  $\sum_{j=1}^m c_{i,j} X_j = 0$ ,  $i = 1, \dots, k + 1$ . We also have  $\deg(V_c \cap H) \leq \deg(V_c) \leq (\max_{i \in [m]} \deg(f_i))^k$  (see, e.g., [9, Thm.8.48]).

Assume  $O \in V$ . Then  $O' \in V'$  since  $\pi(O) = O'$ . Applying Lemma 20 to each of the irreducible components of  $V_c \cap H$  and  $W_H$ , as subvarieties of  $H \cong \mathbb{P}^{m-1}$ , we see  $V_c \cap W_H = (V_c \cap H) \cap W_H = \emptyset$  holds with probability at least  $1 - k \deg(V_c \cap H)/|S| \geq 1 - \delta$ . So by Lemma 19,  $\dim V' = k$  holds with probability at least  $1 - \delta$ .

Now assume  $O \notin V$ . Let  $\pi_{O,H} : V_c \rightarrow H$  be the projection of  $V_c$  from  $O$  to  $H$ , defined by  $P \mapsto \overline{OP} \cap H$  for  $P \in V_c$ . It is well defined since  $O \notin V_c$ . The image  $\pi_{O,H}(V_c)$  is a projective subvariety of  $H$  [17, Thm.3.5]. If  $V_c \cap W_c$  contains a point  $P$ , then  $\pi_{O,H}(V_c) \cap W_H$  contains  $\pi_{O,H}(P)$ . Conversely, if  $\pi_{O,H}(V_c) \cap W_H$  contains a point  $Q$ , then there exists  $P \in V_c$  such that  $Q = \pi_{O,H}(P)$ , and we have  $P \in \overline{OQ} \subseteq W_c$ . We conclude that  $\pi_{O,H}(V_c) \cap W_H = \emptyset$  iff  $V_c \cap W_c = \emptyset$ , which implies  $V_c \cap W_H = \emptyset$ .

Note that  $\dim \pi_{O,H}(V_c) = \dim V_c = k$ , since  $\pi_{O,H}(V_c) = J(\{O\}, V_c) \cap H$ . We also have  $\deg(\pi_{O,H}(V_c)) \leq \deg(V_c)$  [17, Eg.18.16]. Applying Lemma 20 to  $\pi_{O,H}(V_c)$  and  $W_H$ , as subvarieties of  $H \cong \mathbb{P}^{m-1}$ , we see  $\pi_{O,H}(V_c) \cap W_H = \emptyset$  holds with probability at least  $1 - (k + 1) \deg(\pi_{O,H}(V_c)) / |S| \geq 1 - \delta$ .

By Lemma 19 and the previous paragraphs, it holds with probability at least  $1 - \delta$  that  $\dim V' = k$  and  $O' \notin V'$ . ◀

**Proof of Theorem 2.** AnnAtZero is known to be NP-hard [22]. The NP-hardness of APS follows from Lemma 13 and Theorem 15.

Given an instance  $\mathbf{f}$  of APS, we can first find the  $\text{trdeg } k$ . Fix a subset  $S \subset \mathbb{A}$  to be larger than  $2(k + 1)(\max_{i \in [m]} \deg(f_i))^k$  (which can be scanned using only polynomial-space). Consider the points  $((c_{i,j} \mid i \in [k + 1], j \in [m])) \in S^{(k+1) \times m}$ ; for each such point define  $\mathbf{g} := \{g_i := \sum_{j=1}^m c_{i,j} f_j \mid i \in [k + 1]\}$ . Compute the  $\text{trdeg}$  of  $\mathbf{g}$ , and if it is  $k$  then solve AnnAtZero for the instance  $\mathbf{g}$ . Output NO iff some  $\mathbf{g}$  failed the AnnAtZero test.

All these steps can be achieved in space polynomial in the input size, using the uniqueness of the annihilator for  $\mathbf{g}$  [22, Lem.7], Perron’s degree bound [36] and linear algebra [11]. ◀

## 5 Hitting-set for $\overline{\text{VP}}$ : Proof of Theorem 3

Suppose  $p$  is a prime. Define  $\mathbb{A} := \overline{\mathbb{F}}_p$ . We want to find hitting-sets for certain polynomials in  $\mathbb{A}[x_1, \dots, x_n]$ . Fix a  $p$ -power  $q \geq \Omega(sr^6)$ , for the given parameters  $s, r$ . Assume that  $p \nmid (r + 1)$ . Also, fix a model for the finite field  $\mathbb{F}_q$  [1]. We now define the notion of ‘infinitesimally approximating’ a polynomial by a small circuit.

**Approximative closure of VP.** [7] A family  $(f_n|n)$  of polynomials from  $\mathbb{A}[\mathbf{x}]$  is in the class  $\overline{\text{VP}}_{\mathbb{A}}$  if there are polynomials  $f_{n,i}$  and a function  $t : \mathbb{N} \mapsto \mathbb{N}$  such that  $g_n$  has a  $\text{poly}(n)$ -size  $\text{poly}(n)$ -degree algebraic circuit, over the field  $\mathbb{A}(\varepsilon)$ , computing  $g_n(\mathbf{x}) = f_n(\mathbf{x}) + \varepsilon f_{n,1}(\mathbf{x}) + \varepsilon^2 f_{n,2}(\mathbf{x}) + \dots + \varepsilon^{t(n)} f_{n,t(n)}(\mathbf{x})$ . That is,  $g_n \equiv f_n \pmod{\varepsilon \mathbb{A}[\varepsilon][\mathbf{x}]}$ .

The smallest possible circuit size of  $g_n$  is called the *approximative complexity* of  $f_n$ , namely  $\overline{\text{size}}(f_n)$ .

It may happen that  $g_n$  is much easier than  $f_n$  in terms of traditional circuit complexity. That possibility makes the definition interesting and opens up a long line of research.

**Hitting-set for  $\overline{\text{VP}}_{\mathbb{A}}$ .** Given functions  $s = s(n)$  and  $r = r(n)$ , a finite subset  $\mathcal{H} \subset \mathbb{A}^n$  is called a *hitting-set* for degree- $r$  polynomials of approximative complexity  $s$ , if for every such nonzero polynomial  $f$ :  $\exists \mathbf{v} \in \mathcal{H}, f(\mathbf{v}) \neq 0$ .

**Explicitness.** We are interested in computing such a hitting-set in  $\text{poly}(s, \log r, \log q)$ -time.

Before our work, the best result known was EXPSPACE [33, 32]. Heintz and Schnorr [19] proved that  $\text{poly}(s, \log qr)$ -sized hitting-sets exist aplenty (for degree- $r$   $\overline{\text{size}}\text{-}s$  polynomials).

► **Lemma 21.** [19, Thm.4.4] *There exists a hitting-set  $\mathcal{H} \subset \mathbb{F}_q^n$  of size  $O(s^2 n^2)$  (assuming  $q \geq \Omega(sr^2)$ ) that hits all nonzero degree- $r$   $n$ -variate polynomials in  $\mathbb{A}[\mathbf{x}]$  that can be infinitesimally approximated by size- $s$  algebraic circuits.*

Note that for the hitting-set design problem it suffices to focus only on homogeneous polynomials. They are known to be computable by homogeneous circuits, where each gate computes a homogeneous polynomial (see [43]).

**Universal circuit.** It can simulate any circuit of size- $s$  computing a degree- $r$  homogeneous polynomial in  $\mathbb{A}(\varepsilon)[x_1, \dots, x_n]$ . We define the *universal circuit*  $\Psi(\mathbf{y}, \mathbf{x})$  as a circuit in  $n$  essential variables  $\mathbf{x}$  and  $s' := O(sr^4)$  auxiliary variables  $\mathbf{y}$ . The variables  $\mathbf{y}$  are the ones that one can specialize in  $\mathbb{A}(\varepsilon)$ , to compute a specific polynomial in  $\mathbb{A}(\varepsilon)[x_1, \dots, x_n]$ . Every specialization gives a homogeneous degree- $r$  size- $s'$  polynomial. Moreover, the set of these polynomials is closed under constant multiples (see [16, Thm.2.2]).

Note that by [19] there is a hitting-set, with  $m := O(s'^2 n^2)$  points in  $\mathbb{F}_q^n$  ( $q \geq \Omega(s'r^2)$ ), for the set of polynomials  $\mathcal{P}$  approximated by the specializations of  $\Psi(\mathbf{y}, \mathbf{x})$ . A universal circuit construction can be found in [37, 43]. Using the above notation, we give a criterion to decide whether a candidate set is a hitting-set.

► **Theorem 22 (hs criterion).** *Set  $\mathcal{H} =: \{\mathbf{v}_1, \dots, \mathbf{v}_m\} \subset \mathbb{F}_q^n$  is not a hitting-set for the family of polynomials  $\mathcal{P}$  iff there is a satisfying assignment  $(\alpha, \beta) \in \mathbb{A}(\varepsilon)^{s'} \times \mathbb{A}(\varepsilon)^n$  such that:*

- (1)  $\forall i \in [n], \beta_i^{r+1} - 1 \in \varepsilon \mathbb{A}[\varepsilon]$ , and
- (2)  $\Psi(\alpha, \beta) - 1 \in \varepsilon \mathbb{A}[\varepsilon]$ , and
- (3)  $\forall i \in [m], \Psi(\alpha, \mathbf{v}_i) \in \varepsilon \mathbb{A}[\varepsilon]$ .

► **Remark.** The above criterion holds for algebraically closed fields  $\mathbb{A}$  of *any* characteristic. Thus, it reduces those hitting-set verification problems to APS as well.

**Proof.** First we show that:  $\exists x \in \mathbb{A}(\varepsilon), x^{r+1} - 1 \in \varepsilon \mathbb{A}[\varepsilon]$  implies  $x \in \mathbb{A}[[\varepsilon]] \cap \mathbb{A}(\varepsilon)$  (= rational functions defined at  $\varepsilon = 0$ ).

Recall the formal power series  $\mathbb{A}[[\varepsilon]]$  and its group of units  $\mathbb{A}[[\varepsilon]]^*$ . Note that for any polynomial  $a = (\sum_{i_0 \leq i \leq d} a_i \varepsilon^i)$  with  $a_{i_0} \neq 0$ , the inverse  $a^{-1} = \varepsilon^{-i_0} \cdot (\sum_{i_0 \leq i \leq d} a_i \varepsilon^{i-i_0})^{-1}$  is in  $\varepsilon^{-i_0} \cdot \mathbb{A}[[\varepsilon]]^*$ . This is just a consequence of the identity  $(1 - \varepsilon)^{-1} = \sum_{i \geq 0} \varepsilon^i$ . In other words, any rational function  $a \in \mathbb{A}(\varepsilon)$  can be written as an element in  $\varepsilon^{-i} \mathbb{A}[[\varepsilon]]^*$ , for some  $i \geq 0$ . Thus, write  $x$  as  $\varepsilon^{-i} \cdot (b_0 + b_1 \varepsilon + \dots)$  for  $i \geq 0$  and  $b_0 \in \mathbb{A}^*$ . This gives

$$x^{r+1} - 1 = \varepsilon^{-i(r+1)} (b_0 + b_1 \varepsilon + b_2 \varepsilon^2 + \dots)^{r+1} - 1.$$

For this to be in  $\varepsilon \mathbb{A}[\varepsilon]$ , clearly  $i$  has to be 0 (otherwise,  $\varepsilon^{-i(r+1)}$  remains uncanceled); implying that  $x \in \mathbb{A}[[\varepsilon]]$ .

Moreover, we deduce that  $b_0^{r+1} - 1 = 0$ . Thus, condition (1) implies that  $b_0$  is one of the  $(r+1)$ -th roots of unity  $Z_{r+1} \subset \mathbb{A}$  (recall that, since  $p \nmid (r+1)$ ,  $|Z_{r+1}| = r+1$ ). Thus,  $x \in Z_{r+1} + \varepsilon \mathbb{A}[[\varepsilon]]$ .

[ $\Rightarrow$ ]: Suppose  $\mathcal{H}$  is not a hitting-set for  $\mathcal{P}$ . Then, there is a specialization  $\alpha \in \mathbb{A}(\varepsilon)^{s'}$  of the universal circuit such that  $\Psi(\alpha, \mathbf{x})$  computes a polynomial in  $\mathbb{A}[\varepsilon][\mathbf{x}] \setminus \varepsilon \mathbb{A}[\varepsilon][\mathbf{x}]$ , but still ‘fools’  $\mathcal{H}$ , i.e.:  $\forall i \in [m], \Psi(\alpha, \mathbf{v}_i) \in \varepsilon \mathbb{A}[\varepsilon]$ . What remains to show is that conditions (1) and (2) can be satisfied too.

Consider the polynomial  $g(\mathbf{x}) := \Psi(\alpha, \mathbf{x})|_{\varepsilon=0}$ . It is a nonzero polynomial, in  $\mathbb{A}[\mathbf{x}]$  of degree- $r$ , that ‘fools’  $\mathcal{H}$ . By [42], there is a  $\beta \in Z_{r+1}^n$  such that  $a := g(\beta)$  is in  $\mathbb{A}^*$ . Clearly,  $\beta_i^{r+1} - 1 = 0$ , for all  $i$ . Consider  $\psi' := a^{-1} \cdot \Psi(\alpha, \mathbf{x})$ . Note that  $\psi'(\beta) - 1 \in \varepsilon \mathbb{A}[\varepsilon]$ , and  $\psi'(\mathbf{v}_i) \in \varepsilon \mathbb{A}[\varepsilon]$  for all  $i$ . Moreover, the normalized polynomial  $\psi'(\mathbf{x})$  can easily be obtained from the universal circuit  $\Psi$  by changing one of the coordinates of  $\alpha$  (eg. the incoming wires of the root of the circuit). This means that the three conditions (1)-(3) can be simultaneously satisfied by (some)  $(\alpha', \beta) \in \mathbb{A}(\varepsilon)^{s'} \times Z_{r+1}^n$ .

[ $\Leftarrow$ ]: Suppose the satisfying assignment is  $(\alpha, \beta') \in \mathbb{A}(\varepsilon)^{s'} \times \mathbb{A}(\varepsilon)^n$ . As shown before, condition (1) implies:  $\beta'_i \in Z_{r+1} + \varepsilon \mathbb{A}[[\varepsilon]]$  for all  $i \in [n]$ . Let us define  $\beta_i := \beta'_i|_{\varepsilon=0}$ , for all  $i \in [n]$ ; they are in  $Z_{r+1} \subset \mathbb{A}$ . By Condition (3):  $\forall i \in [m], \Psi(\alpha, \mathbf{v}_i) \in \varepsilon \mathbb{A}[\varepsilon]$ .



Previous calculations suggest that  $\Psi(\alpha, \mathbf{x})$  is in  $\varepsilon^{-j}\mathbb{A}[[\varepsilon]][\mathbf{x}]$ , for some  $j \geq 0$ . Expand the polynomial  $\Psi(\alpha, \mathbf{x})$ , wrt  $\varepsilon$ , as:

$$g_{-j}(\mathbf{x})\varepsilon^{-j} + \dots + \varepsilon^{-2}g_{-2}(\mathbf{x}) + g_{-1}(\mathbf{x})\varepsilon^{-1} + g_0(\mathbf{x}) + \varepsilon g_1(\mathbf{x}) + \varepsilon^2 g_2(\mathbf{x}) + \dots$$

Let us study Condition (2). If for each  $0 \leq \ell \leq j$ , polynomial  $g_{-\ell}(\mathbf{x})$  is zero, then  $\Psi(\alpha, \beta')|_{\varepsilon=0} = 0$  contradicting the condition. Thus, we can pick the largest  $0 \leq \ell \leq j$  such that the polynomial  $g_{-\ell}(\mathbf{x}) \neq 0$ .

Note that the normalized circuit  $\varepsilon^\ell \cdot \Psi(\alpha, \mathbf{x})$  equals  $g_{-\ell}$  at  $\varepsilon = 0$ . This means that  $g_{-\ell} \in \mathcal{P}$ , and it is a nonzero polynomial fooling  $\mathcal{H}$ . Thus,  $\mathcal{H}$  cannot be a hitting-set for  $\mathcal{P}$  and we are done.  $\blacktriangleleft$

**Proof of Theorem 3.** Given a prime  $p$  and parameters  $n, r, s$  in unary ( $w \log p \nmid (r+1)$ ), fix a field  $\mathbb{F}_q$  with  $q \geq \Omega(sr^6)$ . Fix the universal circuit  $\Psi(\mathbf{y}, \mathbf{x})$  with  $n$  essential variables  $\mathbf{x}$  and  $s' := \Omega(sr^4)$  auxiliary variables  $\mathbf{y}$ . Fix  $m := \Omega(s'^2 n^2)$ .

For every subset  $\mathcal{H} =: \{\mathbf{v}_1, \dots, \mathbf{v}_m\} \subset \mathbb{F}_q^n$  solve the APS instance described by Conditions (1)-(3) in Theorem 22. These are  $(n+m+1)$  algebraic circuits of degree  $\text{poly}(srn, \log p)$  and a similar bitsize. Using the algorithm from Theorem 2 it can be solved in  $\text{poly}(srn, \log p)$ -space.

The number of subsets  $\mathcal{H}$  is  $q^{nm}$ . So, in  $\text{poly}(nm \log q)$ -space we can go over all of them. If APS fails on one of them (say  $\mathcal{H}$ ) then we know that  $\mathcal{H}$  is a hitting-set for  $\mathcal{P}$ . Since  $\Psi$  is universal, for homogeneous degree- $r$  size- $s$  polynomials in  $\mathbb{A}[\mathbf{x}]$ , we output  $\mathcal{H}$  as the desired hitting-set.  $\blacktriangleleft$

**► Remark.** One advantage in our method compared to the one in [16] is that we can check whether any given set of points is a hitting-set for  $\overline{\text{VP}}_{\mathbb{A}}$ . The method in [16] can not do this, as it only designed robust hitting sets. Another improvement over [16] is that in our case the bit-complexity of the coordinates in hitting-set points is  $O(\log rs)$ , whereas the bit-complexity of the hitting-set points in [16] is  $\text{poly}(n, s, r)$ .

## 6 Conclusion

Our result on algebraic dependence testing in  $\text{AM} \cap \text{coAM}$  gives further indication that a randomized polynomial time algorithm for the problem exists. Studying the following special case might be helpful to get an idea for designing better algorithms.

Given quadratic polynomials  $f_1, \dots, f_n \in \mathbb{F}_2[x_1, \dots, x_n]$ , test if they are algebraically dependent in randomized polynomial time [34].

As indicated in this paper, approximate polynomials satisfiability, or equivalently testing zero-membership in the Zariski closure of the image, may have further applications to problems in computational algebraic geometry and algebraic complexity.

We know that HN is in AM over fields of characteristic zero, assuming GRH [24]. Can we solve AnnAtZero (or APS) in AM for fields of characteristic zero assuming GRH [22]? This would also imply a better hitting-set construction for  $\overline{\text{VP}}$ .

---

## References

- 1 L. M. Adleman and H. W. Lenstra. Finding irreducible polynomials over finite fields. In *STOC*, pages 350–355, 1986.
- 2 M. Agrawal, C. Saha, R. Satharishi, and N. Saxena. Jacobian hits circuits: Hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits. In *Proceedings of the 44th ACM Symposium on Theory of Computing (STOC)*, pages 599–614, 2012. (In SICOMP special issue).

- 3 Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits, 2017. (To appear in 50th ACM Symposium on Theory of Computing (STOC), 2018). URL: <https://www.cse.iitk.ac.in/users/nitin/research.html>.
- 4 S. Arora and B. Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.
- 5 László Babai. Trading group theory for randomness. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 421–429. ACM, 1985.
- 6 M. Beecken, J. Mittmann, and N. Saxena. Algebraic Independence and Blackbox Identity Testing. *Inf. Comput.*, 222:2–19, 2013. (Conference version in ICALP 2011).
- 7 Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 20:1–20:31, 2017.
- 8 Peter Bürgisser. The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics*, 4(4):369–396, 2004. (Preliminary version in FOCS 2001).
- 9 Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.
- 10 Peter Bürgisser, Ankit Garg, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Alternating Minimization, Scaling Algorithms, and the Null-Cone Problem from Invariant Theory. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPICs*, pages 24:1–24:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPICs.ITCS.2018.24.
- 11 Laszlo Csanky. Fast parallel matrix inversion algorithms. *SIAM Journal on Computing*, 5(4):618–623, 1976. (Conference version in FOCS 1975).
- 12 Harm Derksen and Gregor Kemper. *Computational invariant theory*. Springer, 2015.
- 13 Z. Dvir, A. Gabizon, and A. Wigderson. Extractors and rank extractors for polynomial sources. *Comput. Complex.*, 18(1):1–58, 2009. (Conference version in FOCS 2007).
- 14 Zeev Dvir. Extractors for varieties. In *Proceedings of the 24th IEEE Conference on Computational Complexity (CCC)*, pages 102–113, 2009.
- 15 Richard Ehrenborg and Gian-Carlo Rota. Apolarity and canonical forms for homogeneous polynomials. *European Journal of Combinatorics*, 14(3):157–181, 1993.
- 16 Michael A. Forbes and Amir Shpilka. A PSPACE construction of a hitting set for the closure of small algebraic circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:163, 2017. (To appear in 50th ACM Symposium on Theory of Computing (STOC), 2018).
- 17 Joe Harris. *Algebraic Geometry: A First Course*. Springer, 1992.
- 18 Robin Hartshorne. *Algebraic geometry*, volume 52. Springer Science & Business Media, 2013.
- 19 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute. In *Proceedings of the twelfth annual ACM symposium on Theory of computing*, pages 262–272. ACM, 1980.
- 20 Aubrey W Ingleton. Representation of matroids. *Combinatorial mathematics and its applications*, 23, 1971.
- 21 C. G. J. Jacobi. De determinantibus functionalibus. *J. Reine Angew. Math.*, 22(4):319–359, 1841.
- 22 N. Kayal. The Complexity of the Annihilating Polynomial. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC)*, pages 184–193, 2009.
- 23 Neeraj Kayal and Nitin Saxena. Complexity of ring morphism problems. *computational complexity*, 15(4):342–390, 2006.

- 24 Pascal Koiran. Hilbert's Nullstellensatz is in the polynomial hierarchy. *Journal of complexity*, 12(4):273–286, 1996.
- 25 János Kollár. Sharp effective Nullstellensatz. *Journal of the American Mathematical Society*, 1(4):963–975, 1988.
- 26 Mrinal Kumar and Shubhangi Saraf. Arithmetic circuits with locally low algebraic rank. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 34:1–34:27, 2016.
- 27 Joseph M Landsberg. *Tensors: geometry and applications*, volume 128. American Mathematical Society Providence, RI, 2012.
- 28 François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303. ACM, 2014.
- 29 Thomas Lehmkuhl and Thomas Lickteig. On the order of approximation in approximative triadic decompositions of tensors. *Theoretical Computer Science*, 66(1):1–14, 1989.
- 30 Ernst W Mayr and Albert R Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in mathematics*, 46(3):305–329, 1982.
- 31 Johannes Mittmann, Nitin Saxena, and Peter Scheiblechner. Algebraic independence in positive characteristic: A  $p$ -adic calculus. *Transactions of the American Mathematical Society*, 366(7):3425–3450, 2014.
- 32 Ketan Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.
- 33 Ketan D. Mulmuley. Geometric complexity theory V: Equivalence between blackbox derandomization of polynomial identity testing and derandomization of Noether's normalization lemma. In *FOCS*, pages 629–638, 2012.
- 34 Anurag Pandey, Nitin Saxena, and Amit Sinhababu. Algebraic independence over positive characteristic: New criterion and applications to locally low algebraic rank circuits. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 74:1–74:15, 2016. (In print, *Computational Complexity*, 2018).
- 35 O. Perron. *Algebra I (Die Grundlagen)*. W. de Gruyter, Berlin, 1927.
- 36 Arkadiusz Płoski. Algebraic dependence of polynomials after o. perron and some applications. *Computational Commutative and Non-Commutative Algebraic Geometry*, pages 167–173, 2005.
- 37 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 711–720. ACM, 2008.
- 38 Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009.
- 39 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>.
- 40 Marcus Schaefer and Daniel Štefankovič. The complexity of tensor rank. *Theory of Computing Systems*, Aug 2017. doi:10.1007/s00224-017-9800-y.
- 41 Joachim Schmid. On the affine Bezout inequality. *manuscripta mathematica*, 88(1):225–232, 1995.
- 42 J.T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.
- 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.

## A From Section 2: Algebraic-Geometry

Let  $\mathbb{A} := \overline{\mathbb{F}}$  be the algebraic closure of a field  $\mathbb{F}$ . For  $d \in \mathbb{N}^+$ , write  $\mathbb{A}^d$  for the  $d$ -dimensional affine space over  $\mathbb{A}$ . It is defined to be the set  $\mathbb{A}^d$ , equipped with the *Zariski topology*, defined as follows: A subset  $S$  of  $\mathbb{A}^d$  is *closed* iff it is the set of common zeros of some subset of polynomials in  $\mathbb{A}[X_1, \dots, X_d]$ . For other subsets  $S$  it makes sense to consider the *closure*  $\overline{S}$ —the smallest closed set containing  $S$ . Set  $S$  is *dense* in  $\mathbb{A}^d$  if  $\overline{S} = \mathbb{A}^d$ . Complement of closed sets are called *open*.

A closed set is called a *hypersurface* (resp. *hyperplane*) if it is definable by a single polynomial (resp. single linear polynomial).

Define  $\mathbb{A}^\times := \mathbb{A} \setminus \{0\}$ . Write  $\mathbb{P}^d$  for the  $d$ -dimensional projective space over  $\mathbb{A}$ , defined to be the quotient set  $(\mathbb{A}^{d+1} \setminus \{(0, \dots, 0)\}) / \sim$ . Where  $(x_0, \dots, x_d) \sim (y_0, \dots, y_d)$  iff there exists  $c \in \mathbb{A}^\times$  such that  $y_i = cx_i$  for  $0 \leq i \leq d$ . The set  $\mathbb{P}^d$  is again equipped with the *Zariski topology*, where a subset is closed iff it is the set of common zeros of some subset of *homogeneous* polynomials in  $\mathbb{A}[X_0, \dots, X_d]$ . We use  $(d+1)$ -tuples  $(x_0, \dots, x_d)$  to represent points in  $\mathbb{P}^d$ .

Closed subsets of  $\mathbb{A}^d$  or  $\mathbb{P}^d$  are also called *algebraic sets* or *zerosets*. An algebraic set is *irreducible* if it cannot be written as the union of finitely many proper algebraic sets. An irreducible algebraic subset of an affine (resp. projective) space is also called an *affine variety* (resp. *projective variety*). (In some references, varieties are not required to be irreducible, but in this work we always assume it.) An algebraic set  $V$  can be uniquely represented as the union of finitely many varieties, and these varieties are called the *irreducible components* of  $V$ .

Affine zerosets (resp. varieties) are in 1-1 correspondence with *radical* (resp. *prime*) ideals. Irreducible decomposition of an affine variety mirrors the factoring of an ideal into primary ideals. Finally, note that the affine points are in 1-1 correspondence with *maximal* ideals; it is a simple reformulation of Hilbert's Nullstellensatz.

The affine space  $\mathbb{A}^d$  may be regarded as a subset of  $\mathbb{P}^d$  via the map  $(x_1, \dots, x_d) \mapsto (1, x_1, \dots, x_d)$ . Then the subspace topology of  $\mathbb{A}^d$  induced from the Zariski topology of  $\mathbb{P}^d$  is just the Zariski topology of  $\mathbb{A}^d$ . The set  $\mathbb{P}^d \setminus \mathbb{A}^d$  is the projective subspace of  $\mathbb{P}^d$  defined by  $X_0 = 0$ , called the *hyperplane at infinity*.

For an algebraic subset  $V$  of  $\mathbb{A}^d \subseteq \mathbb{P}^d$ , the smallest algebraic subset  $V'$  of  $\mathbb{P}^d$  containing  $V$  (i.e. the intersection of all algebraic subsets containing  $V$ ) is the *projective closure* of  $V$ , and we have  $V' \cap \mathbb{A}^d = V$ . To see this, note that for  $P = (x_1, \dots, x_d) \in \mathbb{A}^d \setminus V$ , there exists a polynomial  $Q \in \mathbb{A}[X_1, \dots, X_d]$  of degree  $D \in \mathbb{N}$  not vanishing on  $P$  (but vanishing on  $V$ ). Then its homogenization  $Q' \in \mathbb{A}[X_0, \dots, X_d]$ , defined by replacing each monomial  $M = \prod_{i=1}^d X_i^{d_i}$  by  $X_0^{D-\deg(M)} \prod_{i=1}^d X_i^{d_i}$ , does not vanish on  $(1, x_1, \dots, x_d)$ . So,  $(1, \mathbf{x}) \notin V'$ .

For distinct points  $P = (x_0, \dots, x_d), Q = (y_0, \dots, y_d) \in \mathbb{P}^d$ , write  $\overline{PQ}$  for the *projective line* passing through them, i.e.,  $\overline{PQ}$  consists of the points  $(ux_0 + vy_0, \dots, ux_d + vy_d)$ , where  $(u, v) \in \mathbb{A}^2 \setminus \{(0, 0)\}$ .

The *dimension* of a variety  $V$  is defined to be the largest integer  $m$  such that there exists a chain of varieties  $\emptyset \subsetneq V_0 \subsetneq V_1 \subsetneq \dots \subsetneq V_m = V$ . More generally, the dimension of an algebraic set  $V$ , denoted by  $\dim V$ , is the maximal dimension of its irreducible components. Eg. we have  $\dim \mathbb{A}^d = \dim \mathbb{P}^d = d$ . The dimension of the empty set is  $-1$  by convention. One dimensional varieties are called *curves*.

The *degree* of a variety  $V$  in  $\mathbb{A}^d$  (resp.  $\mathbb{P}^d$ ) is the number of intersections of  $V$  with a general affine subspace (resp. projective subspace) of dimension  $d - \dim V$ . More generally, we define the degree of an algebraic set  $V$ , denoted by  $\deg(V)$ , to be the sum of the degrees

of its irreducible components. The degree of an algebraic subset of  $\mathbb{A}^d$  coincides with the degree of its projective closure in  $\mathbb{P}^d$ .

Suppose  $V \subseteq \mathbb{A}^d$  is an algebraic set, defined by polynomials  $f_1, \dots, f_k$ . Let  $(a_1, \dots, a_d) \in \mathbb{A}^d$ . Then the set  $\{(x_1 + a_1, \dots, x_d + a_d) : (x_1, \dots, x_d) \in V\}$  is called a *translate* of  $V$ . It is also an algebraic set, defined by  $f_i(X_1 - a_1, \dots, X_d - a_d)$ ,  $i = 1, \dots, k$ .

Let  $V \subseteq \mathbb{A}^n$ ,  $W \subseteq \mathbb{A}^m$  be affine varieties. A *morphism* from  $V$  to  $W$  is a function  $f : V \rightarrow W$  that is a restriction of a polynomial map  $\mathbb{A}^n \rightarrow \mathbb{A}^m$ . A morphism  $f : V \rightarrow W$  is called *dominant* if  $\overline{\text{Im}(f)} = W$ . The preimage of a closed subset under a morphism is closed (i.e. morphisms are *continuous* in the Zariski topology).

For a polynomial map  $f : \mathbb{A}^n \rightarrow \mathbb{A}^m$  and an affine variety  $V \subseteq \mathbb{A}^n$ ,  $W := \overline{f(V)}$  is also an affine variety (i.e., it is irreducible). To see this, assume to the contrary that  $W$  is the union of two proper closed subsets  $W_1$  and  $W_2$ . By the definition of closure,  $f(V)$  is not contained in either  $W_1$  or  $W_2$ , i.e., it intersects both. Then  $f^{-1}(W_1) \cap V$  and  $f^{-1}(W_2) \cap V$  are two proper closed subsets of  $V$ , and their union is  $V$ . This contradicts the irreducibility of  $V$ .

The *graph*  $\Gamma_f$  of a morphism  $f$  is the set  $\{(x, f(x)) : x \in V\} \subseteq V \times W \subseteq \mathbb{A}^n \times \mathbb{A}^m$ . Here  $V \times W = \{(x, y) : x \in V, y \in W\}$  denotes the *product* of  $V$  and  $W$ , which is a subvariety of the  $(n + m)$ -dimensional affine space  $\mathbb{A}^n \times \mathbb{A}^m \cong \mathbb{A}^{n+m}$ . Note the graph  $\Gamma_f$  is closed in  $\mathbb{A}^n \times \mathbb{A}^m$ : Suppose  $f$  sends  $x \in V$  to  $(f_1(x), \dots, f_m(x)) \in \mathbb{A}^m$ , where  $f_i \in \mathbb{A}[X_1, \dots, X_n]$  for  $i \in [m]$ . And suppose  $V$  is defined by an ideal  $I \subseteq \mathbb{A}[X_1, \dots, X_n]$ . Then  $\Gamma_f$  is defined by the ideal of  $\mathbb{A}[X_1, \dots, X_n, Y_1, \dots, Y_m]$  generated by  $I$  and the polynomials  $Y_i - f_i(X_1, \dots, X_n)$ ,  $i = 1, \dots, m$ .

## B From Section 4

► **Example 23.** Let  $m = 4$ ,  $(f_1, f_2, f_3, f_4) = (X_1, X_2, X_1X_2 - 1, X_1 + X_2)$ . Then  $k := \text{trdeg}f = 2$ . Let  $(g_1, g_2, g_3) = (f_1, f_3, f_1 + f_2 - f_4) = (X_1, X_1X_2 - 1, 0)$ . Suppose  $\mathbb{A}^m$  has coordinates  $Y_1, \dots, Y_4$  and  $\mathbb{A}^{k+1}$  has coordinates  $Z_1, \dots, Z_3$ .

Then  $V \subseteq \mathbb{A}^m$  is defined by  $Y_1Y_2 - Y_3 - 1 = 0$  and  $Y_1 + Y_2 - Y_4 = 0$ , and  $W$  is defined by  $Y_1 = 0$ ,  $Y_3 = 0$ , and  $Y_2 - Y_4 = 0$ . So  $V \cap W = \emptyset$ . But  $V' \subseteq \mathbb{A}^{k+1}$  is the plane  $Z_3 = 0$ , which contains the origin.

► **Example 24.** Consider Example 23 but choose  $f_4$  to be  $X_1 + X_2 + 1$  instead of  $X_1 + X_2$ . Now we have  $g_3 = 1$ ,  $V$  is defined by  $Y_1Y_2 - Y_3 - 1 = 0$  and  $Y_1 + Y_2 - Y_4 + 1 = 0$ , and  $V'$  is the plane  $Z_3 = 1$ . So  $O' \notin V'$ .

On the other hand, suppose  $\mathbb{P}^m$  has coordinates  $Y_0, \dots, Y_4$ . Then  $V_c \cap H$  is defined by  $Y_0 = Y_1Y_2 = Y_1 + Y_2 - Y_4 = 0$ , and  $W_H$  is defined by  $Y_0 = Y_1 = Y_2 - Y_4 = Y_3 = 0$ . So  $(0, 0, 1, 0, 1) \in V_c \cap W_H \subseteq V_c \cap W_c$ .



# Unbalancing Sets and an Almost Quadratic Lower Bound for Syntactically Multilinear Arithmetic Circuits

Noga Alon<sup>1</sup>

Sackler School of Mathematics and Blavatnik School of Computer Science  
Tel Aviv, 6997801, Israel, and  
Center for Mathematical Sciences and Applications, Harvard University,  
Cambridge, MA 02138, USA  
nogaa@tau.ac.il

Mrinal Kumar<sup>2</sup>

Center for Mathematical Sciences and Applications, Harvard University  
Cambridge, MA 02138, USA  
mrinalkumar08@gmail.com

Ben Lee Volk<sup>3</sup>

Blavatnik School of Computer Science, Tel Aviv University  
Tel Aviv, 6997801, Israel  
benleevolk@gmail.com

---

## Abstract

We prove a lower bound of  $\Omega(n^2/\log^2 n)$  on the size of any syntactically multilinear arithmetic circuit computing some explicit multilinear polynomial  $f(x_1, \dots, x_n)$ . Our approach expands and improves upon a result of Raz, Shpilka and Yehudayoff ([31]), who proved a lower bound of  $\Omega(n^{4/3}/\log^2 n)$  for the same polynomial. Our improvement follows from an asymptotically optimal lower bound for a generalized version of Galvin's problem in extremal set theory.

**2012 ACM Subject Classification** Theory of computation → Algebraic complexity theory

**Keywords and phrases** Algebraic Complexity, Multilinear Circuits, Circuit Lower Bounds

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.11

**Related Version** <https://arxiv.org/abs/1708.02037>

**Acknowledgements** Part of this work was done while the second author was visiting Tel Aviv University. We thank Amir Shpilka for the visit, for many insightful discussions, and for comments on an earlier version of this text. We are also thankful to Andy Drucker for pointing out a correction in a previous version of this paper.

## 1 Introduction

An arithmetic circuit is one of the most natural and standard computational models for computing multivariate polynomials. Such circuits provide a succinct representation of

---

<sup>1</sup> Research supported in part by an ISF grant and by a GIF grant.

<sup>2</sup> Part of this work was done while visiting Tel Aviv University.

<sup>3</sup> The research leading to these results has received funding from the Israel Science Foundation (grant number 552/16).



multivariate polynomials, and in some sense, they can be thought of as algebraic analogs of boolean circuits. Formally, an arithmetic circuit over a field  $\mathbb{F}$  and a set of variables  $X = \{x_1, x_2, \dots, x_n\}$  is a directed acyclic graph in which every vertex has in-degree either zero or two. The vertices of in-degree zero (called *leaves*) are labeled by variables in  $X$  or elements of  $\mathbb{F}$ , and the vertices of in-degree two are labeled by either  $+$  (called *sum* gates) or  $\times$  (called *product* gates). A circuit can have one or more vertices of out degree zero, known as the output gates. The polynomial computed by a vertex in any<sup>4</sup> given circuit is naturally defined in an inductive way: a leaf computes the polynomial which is equal to its label. A sum gate computes the polynomial which is the sum of the polynomials computed at its children and a product gate computes the polynomial which is the product of the polynomials at its children. The polynomials computed by a circuit are the polynomials computed by its output gates. The size of an arithmetic circuit is the number of vertices in it.

It is not hard to show (see, e.g., [7]) that a random polynomial of degree  $d = \text{poly}(n)$  in  $n$  variables cannot be computed by an arithmetic circuit of size  $\text{poly}(n)$  with overwhelmingly high probability. A fundamental problem in this area of research is to prove a similar super-polynomial lower bound for an *explicit* polynomial family. Unfortunately, the problem continues to remain wide open and the current best lower bound known for general arithmetic circuits<sup>5</sup> is an  $\Omega(n \log n)$  lower bound due to Strassen [37] and Baur and Strassen [5] from more than three decades ago. The absence of substantial progress on this general question has led to focus on the question of proving better lower bounds for restricted and more structured subclasses of arithmetic circuits. Arithmetic formulas [19], non-commutative arithmetic circuits [26], algebraic branching programs [22], and low depth arithmetic circuits [27, 13, 14, 30, 15, 11, 20, 24, 23] are some such subclasses which have been studied from this perspective. For an overview of the definition of these models and the state of art for lower bounds for them, we refer the reader to the surveys of Shpilka and Yehudayoff [35] and Saptharishi [34].

Several of the most important polynomials in algebraic complexity and in mathematics in general are multilinear. Notable examples include the determinant, the permanent, and the elementary symmetric polynomials. Therefore, one subclass which has received a lot of attention in the last two decades and will be the focus of this paper is the class of *multilinear* arithmetic circuits.

## 1.1 Multilinear arithmetic circuits

For an arithmetic circuit  $\Psi$  and a vertex  $v$  in  $\Psi$ , we denote by  $X_v$  the set of variables  $x_i$  such that there is a directed path from a leaf labeled by  $x_i$  to  $v$ ; in this case, we also say that  $v$  *depends* on  $x_i$ <sup>6</sup>. A polynomial  $P$  is said to be multilinear if the individual degree of every variable in  $P$  is at most one.

An arithmetic circuit  $\Psi$  is said to be *syntactically* multilinear if for every multiplication gate  $v$  in  $\Psi$  with children  $u$  and  $w$ , the sets of variables  $X_u$  and  $X_w$  are disjoint. We say that  $\Psi$  is *semantically* multilinear if the polynomial computed at every vertex is a multilinear polynomial. Observe that if  $\Psi$  is a syntactically multilinear circuit, then it is also semantically multilinear. However, it is not clear if every semantically multilinear circuit can be efficiently simulated by a syntactically multilinear circuit.

---

<sup>4</sup> Throughout this paper, we will use the terms gates and vertices interchangeably.

<sup>5</sup> In the rest of the paper, when we say a lower bound, we always mean it for an explicit polynomial family.

<sup>6</sup> We remark that this is a syntactic notion of dependency, since it is possible that every monomial with  $x_i$  might get canceled in the intermediate computation and might not eventually appear in the polynomial computed at  $v$ .



A multilinear circuit is a natural model for computing multilinear polynomials, but it is not necessarily the most efficient one. Indeed, it is remarkable that all the constructions of polynomial size arithmetic circuits for the determinant [8, 6, 25], which are fundamentally different from one another, nevertheless share the property of being *non*-multilinear, namely, they involve non-multilinear intermediate computations which eventually cancel out. There are no subexponential-size multilinear circuits known for the determinant, and one may very well conjecture these do not exist at all.

Multilinear circuits were first studied by Nisan and Wigderson [27]. Subsequently, Raz [29] defined the notion of multilinear formulas<sup>7</sup> and showed that any multilinear formula computing the determinant or the permanent of an  $n \times n$  variable matrix must have super-polynomial size. In a follow up work [28], Raz further strengthened the results in [29] and showed that there is a family of multilinear polynomials in  $n$  variables which can be computed by a  $\text{poly}(n)$  size syntactically multilinear arithmetic circuits but require multilinear formulas of size  $n^{\Omega(\log n)}$ .

Building on the ideas and techniques developed in [29], Raz and Yehudayoff [33] showed an exponential lower bound for syntactically multilinear circuits of constant depth. Interestingly, they also showed a super-polynomial separation between depth  $\Delta$  and depth  $\Delta+1$  syntactically multilinear circuits for constant  $\Delta$ .

In spite of the aforementioned progress on the question of lower bounds for multilinear formulas and bounded depth syntactically multilinear circuits, there was no  $\Omega(n^{1+\varepsilon})$  lower bounds known for general syntactically multilinear circuits for any constant  $\varepsilon > 0$ . In fact, the results in [28] show that the main technical idea underlying the results in [29, 28, 33] is unlikely to directly give a super-polynomial lower bound for general syntactically multilinear circuits. However, a weaker super-linear lower bound still seemed conceivable via similar techniques.

Raz, Shpilka and Yehudayoff [31] showed that this is indeed the case. By a sophisticated and careful application of the techniques in [29] along with several additional ideas, they established an  $\Omega\left(\frac{n^{4/3}}{\log^2 n}\right)$  lower bound for an explicit  $n$  variate polynomial. Since then, this has remained the best lower bound known for syntactically multilinear circuits. In this paper, we improve this result by showing an almost quadratic lower bound for syntactically multilinear circuits for an explicit  $n$  variate polynomial. In fact, the family of hard polynomials in this paper is the same as the one used in [31]. We now formally state our result.

► **Theorem 1.** *There is an explicit family of polynomials  $\{f_n\}$ , where  $f_n$  is an  $n$  variate multilinear polynomial, such that any syntactically multilinear arithmetic circuit computing  $f_n$  must have size at least  $\Omega(n^2/\log^2 n)$ .*

For our proof, we follow the strategy in [31]. Our improvement comes from an improvement in a key lemma in [31] which addresses the following combinatorial problem.

► **Question 2.** *What is the minimal integer  $m = m(n)$  for which there is a family of subsets  $S_1, S_2, \dots, S_m \subseteq [n]$ , each  $S_i$  satisfying  $6 \log n \leq |S_i| \leq n - 6 \log n$  such that for every  $T \subseteq [n]$ ,  $|T| = \lfloor n/2 \rfloor$ , there exists an  $i \in [m]$  with  $|T \cap S_i| \in \{\lfloor |S_i|/2 \rfloor - 3 \log n, \lfloor |S_i|/2 \rfloor - 3 \log n + 1, \dots, \lfloor |S_i|/2 \rfloor + 3 \log n\}$ ?*

Raz, Shpilka and Yehudayoff [31] showed that  $m(n) \geq \Omega(n^{1/3}/\log n)$ . For our proof, we show that  $m(n) \geq \Omega(n/\log n)$ .

<sup>7</sup> For formulas, it is known that syntactic multilinearity and semantically multilinearity are equivalent (See, e.g., [29]).

In addition to its application to the proof of Theorem 1, Question 2 seems to be a natural problem in extremal combinatorics and might be of independent interest, and special cases thereof were studied in the combinatorics literature. In the next section, we briefly discuss the state of the art of this question and state our main technical result about it in Theorem 3.

## 1.2 Unbalancing Sets

The following question, which is of very similar nature to Question 2, is known as Galvin's problem (see [12, 9]): What is the minimal integer  $m = m(n)$ , for which there exists a family of subsets  $S_1, \dots, S_m \subseteq [4n]$ , each of size  $2n$ , such that for every subset  $T \subseteq [4n]$  of size  $2n$  there exists some  $i \in [m]$  such that  $|T \cap S_i| = n$ ?

It is not hard to show that  $m(n) \leq 2n$ . Indeed, let  $S_i = \{i, i+1, \dots, i+2n-1\}$ , for  $i \in \{1, 2, \dots, 2n+1\}$ , and let  $\alpha_i(T) = |T \cap S_i| - |([4n] \setminus T) \cap S_i|$ . Then  $\alpha_i(T)$  is always an even integer,  $\alpha_1(T) = -\alpha_{2n+1}(T)$ , and  $\alpha_i - \alpha_{i+1}(T) \in \{0, \pm 2\}$  if  $i \leq 2n$ . By a discrete version of the intermediate value theorem, it follows there exists  $j \in [2n]$  such that  $\alpha_j(T) = 0$ , which implies that exactly  $n$  elements of  $S_j$  belong to  $T$ . Thus, the family  $\{S_1, \dots, S_{2n}\}$  satisfies this property.

As for lower bounds, a counting argument shows that  $m(n) = \Omega(\sqrt{n})$ , since for each fixed  $S$  of size  $[2n]$  and random  $T$  of size  $2n$ ,

$$\Pr[|T \cap S| = n] = \frac{\binom{2n}{n} \cdot \binom{2n}{n}}{\binom{4n}{2n}} = \Theta\left(\frac{1}{\sqrt{n}}\right).$$

Frankl and Rödl [12] were able to show that  $m(n) \geq \varepsilon n$  for some  $\varepsilon > 0$  if  $n$  is odd, and Enomoto, Frankl, Ito and Nomura [9] proved that  $m(n) \geq 2n$  if  $n$  is even, which implies that even the constant in the construction given above is optimal. Until this work, the question was still open for even values of  $n$ : in fact, Markert and West (unpublished, see [9]) showed that for  $n \in \{2, 4\}$ ,  $m(n) < 2n$ .

For our purposes, we need to generalize Galvin's problem in two ways. The first is to lift the restriction on the set sizes. The second is to ask how small can the size of the family  $\mathcal{F} = \{S_1, \dots, S_m\} \subseteq 2^{[n]}$  be if we merely assume each balanced partition  $T$  is " $\tau$ -balanced" on some  $S \in \mathcal{F}$ , namely, if  $||T \cap S| - |S|/2|| \leq \tau$  for some  $S$  (the main case of interest for us is  $\tau = O(\log n)$ ). Of course, since  $T$  itself is balanced, very small or very large sets are always  $\tau$ -balanced, and thus we impose the (tight) non-triviality condition  $2\tau \leq |S| \leq n - 2\tau$  for every  $S \in \mathcal{F}$ .

Once again, by defining  $S_i = \{i, i+1, \dots, i+n/2-1\}$  ( $n$  is always assumed to be even), the family  $\mathcal{F} = \{S_1, S_{1+\tau}, S_{1+2\tau}, \dots, S_{1+\lfloor n/(2\tau) \rfloor \cdot \tau}\}$  gives a construction of size  $O(n/\tau)$  such that every balanced partition  $T$  is  $\tau$ -balanced on some  $S \in \mathcal{F}$ .

It is natural to conjecture that, perhaps up to a constant, this construction is optimal. Indeed, this is what we prove here.

► **Theorem 3.** *Let  $n$  be any large enough even number, and let  $\tau \geq 1$  be an integer. Let  $S_1, \dots, S_m \subseteq [n]$  be sets such that for all  $i \in [m]$ ,  $2\tau \leq |S_i| \leq n - 2\tau$ . Further, assume that for every  $Y \subseteq [n]$  of size  $n/2$  there exists  $i \in [m]$  such that  $||Y \cap S_i| - |S_i|/2| < \tau$ . Then,  $m \geq \Omega(n/\tau)$ .*

In particular, Theorem 3 proves a linear lower bound  $m = \Omega(n)$  for the original problem of Galvin, even when the universe size is of the form  $4k$  for even  $k$ .

We remark that the relevance of problems of this form to lower bounds in algebraic complexity was also observed by Jansen [18] who considered the problem of obtaining a

lower bound on homogenous syntactically multilinear algebraic branching program (which is a weaker model than syntactically multilinear circuits), and essentially proposed Theorem 3 as a conjecture. In fact, a special case of this theorem (see Theorem 9), which has a simpler proof, is already enough to derive the improved lower bounds for syntactically multilinear circuits.

Alon, Bergmann, Coppersmith and Odlyzko [1] considered a very similar problem of balancing  $\pm 1$ -vectors: they studied families of vectors  $\mathcal{F} = \{v_1, \dots, v_m\}$  such that  $v_i \in \{\pm 1\}^n$  for  $i \in [m]$ , which satisfy the properties that for every  $w \in \{\pm 1\}^n$  (not necessarily balanced), there exists  $i \in [m]$  such that  $|\langle v_i, w \rangle| \leq d$ . They generalized a construction of Knuth [21] and proved a matching lower bound which together showed that  $m = \lceil n/(d+1) \rceil$  is both necessary and sufficient for such a set to exist. Galvin's problem seems like "the  $\{0, 1\}$  version" of the same problem, but, to quote from [1], there does not seem to be any simple dependence between the problems.

### 1.3 Proof overview

In this section, we discuss the main ideas and give a brief sketch of the proofs of Theorem 1 and Theorem 3. Since our proof heavily depends on the proof in [31] and follows the same strategy, we start by revisiting the main steps in their proof and noting the key differences between the proof in [31] and our proof. We also outline the reduction to the combinatorial problem of unbalancing set families in Question 2.

#### Proof sketch of [31]

The proof in [31] starts by proving a syntactically multilinear analog of a classical result of Baur and Strassen [5], where it was shown that if an  $n$  variate polynomial  $f$  is computable by an arithmetic circuit  $\Psi$  of size  $s(n)$ , then there is an arithmetic circuit  $\Psi'$  of size at most  $5s(n)$  with  $n$  outputs such that the  $i$ -th output gate of  $\Psi'$  computes  $f_i = \frac{\partial f}{\partial x_i}$ . Raz, Shpilka and Yehudayoff show that if  $\Psi$  is syntactically multilinear, then the circuit  $\Psi'$  continues to be syntactically multilinear. Additionally, there is no directed path from a leaf labeled by  $x_i$  to the output gate computing  $f_i$ .<sup>8</sup>

Once we have this structural result, it would suffice to prove a lower bound on the size of  $\Psi'$ . For brevity, we denote the subcircuit of  $\Psi'$  rooted at the output gate computing  $f_i$  by  $\Psi'_i$ . As a key step of the proof in [31], the authors identify certain sets of vertices  $\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_n$  in  $\Psi'$  with the following properties.

- For every  $i \in [n]$ ,  $\mathcal{U}_i$  is a subset of vertices in  $\Psi'_i$ .
- For every  $i \in [n]$  and  $v \in \mathcal{U}_i$ , the number of  $j \neq i$  such that  $v \in \mathcal{U}_j$  is not too large (at most  $O(\log n)$ ).

Observe that at this point, showing a lower bound of  $s'(n)$  on the size of each  $\mathcal{U}_i$  implies a lower bound of  $\Omega(ns'(n)/\log n)$  on the size of  $\Psi'$  and hence  $\Psi$ . In [31], the authors show that there is an explicit  $f$  such that each  $\mathcal{U}_i$  must have size at least  $\Omega(n^{1/3}/\log n)$ , thereby getting a lower bound of  $\Omega(n^{4/3}/\log^2 n)$  on the size of  $\Psi$ .

For our proof, we follow precisely this high level strategy. Our improvement in the lower bound comes from showing that each  $\mathcal{U}_i$  must be of size at least  $\Omega(n/\log n)$  and not just  $\Omega(n^{1/3}/\log n)$  as shown in [31]. We now elaborate further on the main ideas in this step in [31] and the differences with the proofs in this paper.

<sup>8</sup> See Theorem 15 for a formal statement.

We start with some intuition into the definition of the sets  $\mathcal{U}_i$  in [31]. Consider a vertex  $v$  in  $\Psi'$  which depends on at least  $k$  variables. Without loss of generality, let these variables be  $\{x_1, x_2, \dots, x_k\}$ . From item 4 in Theorem 15, we know that the variable  $x_i$  does not appear in the subcircuit  $\Psi'_i$ . Therefore, the vertex  $v$  cannot appear in the subcircuits  $\Psi'_1, \Psi'_2, \dots, \Psi'_k$ . So, if we define the set  $\mathcal{U}_i$  as the set of vertices in  $\Psi'_i$  which depend on at least  $k$  variables, then  $\mathcal{U}_i$  must be disjoint from vertices in at least  $k$  of the subcircuits  $\Psi'_1, \Psi'_2, \dots, \Psi'_n$ . Picking  $k \geq n - O(\log n)$  would give us the desired property. So, if we can prove a lower bound on the size of the set  $\mathcal{U}_i$ , we would be done. However, the definition of the set  $\mathcal{U}_i$  so far turns out to be too general: indeed, it is not even a priori clear that the  $\mathcal{U}_i$  has any other gates apart from the output gate of  $\Psi'_i$ .

As is often the case, the solution to this obstacle is to prove a stronger claim by imposing additional structure on the set  $\mathcal{U}_i$ . In [31], the set  $\mathcal{U}_i$  (called the *upper leveled* gates in  $\Psi'_i$ ) is defined as the set of all vertices in  $\Psi'_i$  which depend on at least  $n - 6 \log n$  variables and have a child which depends on more than  $6 \log n$  variables and less than  $n - 6 \log n$  variables. This additional structure is helpful in proving a lower bound on the size of  $\mathcal{U}_i$ . We now discuss this in some more detail.

For every  $i \in [n]$ , let  $\mathcal{L}_i$  be the set of vertices  $u$  in  $\Psi'_i$ , such that  $6 \log n < |X_u| < n - 6 \log n$ , and  $u$  has a parent in  $\mathcal{U}_i$ . These gates are referred to as *lower leveled* gates. Observe that  $|\mathcal{L}_i| \geq \frac{|\mathcal{U}_i|}{2}$ , since the in-degree of every vertex in  $\Psi'_i$  is at most 2. The key structural property of the set  $\mathcal{L}_i$  is the following (see Proposition 5.5 in [31]).

► **Lemma 4** ([31]). *Let  $i \in [n]$ , and let  $h_1, h_2, \dots, h_\ell$  be the polynomials computed by the gates in  $\mathcal{L}_i$ . Then, there exist multilinear polynomials  $g_1, g_2, \dots, g_\ell, g$  such that*

$$f_i = \sum_{j \in [\ell]} g_j \cdot h_j + g \tag{1}$$

where

- For every  $j \in [\ell]$ ,  $h_j$  and  $g_j$  are variable disjoint.
- The degree of  $g$  is at most  $O(\log n)$ .

Observe that (1) is basically a decomposition of a potentially-hard polynomial  $f_i$  in terms of the sum of products of multilinear polynomials in an intermediate number of variables. The goal is to show that for an appropriate explicit  $f_i$ , the number of summands on the right hand side of (1) cannot be too small. A similar scenario also appears in the multilinear formula lower bounds and bounded depth multilinear formula lower bounds of [29, 28, 33] (albeit with some key differences). Hence, a natural approach at this point would be to use the tools in [29, 28, 33], namely the rank of the *partial derivative matrix*, to attempt to prove this lower bound. We refer the reader to Section 2.2 for the definitions and properties of the partial derivative matrix and proceed with the overview. For each  $j \in [\ell]$ , let the polynomial  $h_j$  in Lemma 4 depend on the variables  $S_j \subseteq X$ . The key technical step in the rest of the proof is to show that there is a partition of the set of variables  $X = \{x_1, x_2, \dots, x_n\}$  into  $Y$  and  $Z$  such that  $|Y| = |Z|$  and for every  $j \in [\ell]$ ,  $||S_j \cap Y| - |S_j \cap Z|| \geq \Omega(\log n)$ . In [31], the authors show that there is an absolute constant  $\varepsilon > 0$  such that if  $\ell \leq \varepsilon n^{1/3} / \log n$ , then there is an equipartition of  $X$  which *unbalances* all the sets  $\{S_j : j \in [\ell]\}$  by at least  $\Omega(\log n)$ . Our key technical contribution (Theorem 3) in this paper is to show that as long as  $\ell \leq \varepsilon n / \log n$ , there is an equipartition which unbalances all the  $S_j$ 's by at least  $\Omega(\log n)$ . This implies an  $\Omega(n / \log n)$  on the size of each set  $\mathcal{U}_i$ , and thus an  $\Omega(n^2 / \log^2 n)$  lower bound on the circuit size.

Before we dive into a more detailed discussion on the overview and main ideas in the proof of Theorem 3 in the next section, we would like to remark that the lower bound question in (1) seems to be a trickier question than what is encountered while proving

multilinear formula lower bounds [29, 28] or bounded depth syntactically multilinear circuit lower bounds [33]. The main differences are that in the proofs in [29, 28, 33], the sets  $S_j$  have a stronger guarantee on their size (at least  $n^{\Omega(1)}$  and at most  $n - n^{\Omega(1)}$ ), and each of the summands on the right has *many* variable disjoint factors and not just two factors as in (1). For instance, in the formula lower bound proofs the number of variable disjoint factors in each summand on the right is  $\Omega(\log n)$ , and for constant depth circuit lower bounds it is  $n^{\Omega(1)}$ . Together, these properties make it possible to show much stronger lower bounds on  $\ell$ . In particular, it is known that a *random* equipartition works for these two applications, in the sense that it unbalances sufficiently many factors in each summand, thereby implying that the rank of the partial derivative matrix of the polynomial is small. Hence, for an appropriate<sup>9</sup>  $f_i$ , the number of summands must be large. However, since a set of size  $O(\log n)$  is balanced under a random equipartition with probability  $\Omega(1/\sqrt{\log n})$  and the identity in (1) involves just two variable disjoint factors, taking a random equipartition would not enable us to prove any meaningful bounds.

### Proof sketch of Theorem 3

Recall that our task is, given a small collection of subsets of  $[n]$ , to find a balanced partition which is unbalanced on each of the sets. Equivalently, we would like to prove that if  $\mathcal{F}$  is a family of subsets such that every balanced partition balances at least one set in  $\mathcal{F}$ , then  $|\mathcal{F}|$  must be large (of course,  $\mathcal{F}$  must satisfy the conditions in Theorem 3).

We first sketch the proof of a special case (which suffices for the main application here), when  $n = 4p$  and  $p$  is a prime. For the sake of simplicity, suppose also that all subsets  $S \in \mathcal{F}$  are of even size, and assume further that for every subset  $T \subseteq [n]$  of size  $n/2$  there exists  $S \in \mathcal{F}$  such that  $T$  completely balances  $S$ , namely,  $|T \cap S| = |S|/2$ . One possible approach to obtain lower bounds on  $|\mathcal{F}|$  is via an application of the polynomial method as done, for example, in [1]. Define the following polynomial over, say, the rationals:

$$f(x_1, \dots, x_n) = \prod_{S \in \mathcal{F}} (\langle x, \mathbf{1}_S \rangle - |S|/2).$$

By the assumption on  $\mathcal{F}$ , the polynomial  $f$  evaluates to 0 over all points in  $\{0, 1\}^n$  with Hamming weight exactly  $n/2$ . We can also argue, using the assumption on the set sizes in  $\mathcal{F}$ , that  $f$  is not identically zero, and clearly  $\deg(f) \leq |\mathcal{F}|$ . Thus, a lower bound on  $\deg(f)$  translates to a lower bound on  $|\mathcal{F}|$ .

This idea, however, seems like a complete nonstarter, since there exists a degree 1 non-zero polynomial which evaluates to 0 over the middle layer of  $\{0, 1\}^n$ , namely,  $\sum_i x_i - n/2$ .

A very clever solution to this potential obstacle was found by Hegedűs [16]. Suppose  $n = 4p$  for some prime  $p$ . The main insight in [16] is to consider the polynomial  $f$  over  $\mathbb{F}_p$ , and to add the requirement that there exists some  $z \in \{0, 1\}^{4p}$ , of Hamming weight *exactly*  $3p$ , such that  $f(z) \neq 0$ . This requirement rules out the trivial example  $\sum_i x_i - n/2$ , and Hegedűs was able to show that the degree of any polynomial with these properties must be at least  $p = n/4$  (see Lemma 5 for the complete statement).

We are thus left with the task of proving that our polynomial evaluates to a non-zero value over some point  $z \in \{0, 1\}^{4p}$  of Hamming weight  $3p$ . This turns out to be not very hard to show, assuming each set is of size at least, say,  $100 \log n$  and at most  $n - 100 \log n$ ,

<sup>9</sup>  $f_i$  is chosen so that the the partial derivative matrix for  $f_i$  is of full rank for *every* equipartition.

by choosing a random such vector  $z$ . Indeed, it is not surprising that it is much easier to directly show that a highly unbalanced partition of  $[n]$  (into  $3n/4$  vs  $n/4$ ) unbalances all the sets  $\mathcal{F}$ .<sup>10</sup>

As mentioned earlier, the case  $n = 4p$  and  $\tau \geq 100 \log n$  in Theorem 3 is considerably easier to prove and suffices for the application to circuit lower bounds. Proving this theorem for every even  $n$  and every  $\tau \geq 1$  requires further technical ideas which appear in the full version of this paper [2].

Even though Lemma 5 seems to be a fundamental statement about polynomials over finite fields and could conceivably have an elementary proof, the proof in [16] uses more advanced techniques. It relies on the description of Gröbner basis for ideals of polynomials in  $\mathbb{F}[x_1, x_2, \dots, x_n]$  which vanish on all points in  $\{0, 1\}^n$  of weight equal to  $n/2$ . A complete description of the reduced Gröbner basis for such ideals was given by Hegedűs and Rónyai [17] and their proof builds up on a number of earlier partial results [4, 10] on this problem.

To the best of our knowledge, the proof in [16] is the only known proof of Lemma 5, and giving a self contained elementary proof of it seems to be an interesting question.

### Organization of the paper

In the rest of the paper, we set up some notation and discuss some preliminary notions in Section 2, prove Theorem 3 in Section 3 and complete the proof of Theorem 1 in Section 4. Throughout the paper we assume, whenever this is needed, that  $n$  is sufficiently large, and make no attempts to optimize the absolute constants.

## 2 Preliminaries

For  $n \in \mathbb{N}$ , we denote  $[n] = \{1, 2, \dots, n\}$ . For a prime  $p$ , we denote by  $\mathbb{F}_p$  the finite field with  $p$  elements. For two integers  $i, j$  with  $i \leq j$ , we denote  $[i, j] = \{a \in \mathbb{Z} : i \leq a \leq j\}$ . The characteristic vector of a set  $S \subseteq [n]$  is denoted by  $\mathbf{1}_S \in \{0, 1\}^n$ .

As is standard,  $\binom{[n]}{k}$  denotes the family  $\{S \subseteq [n] : |S| = k\}$ .

For an even  $n \in \mathbb{N}$  and  $Y \subseteq [n]$  such that  $|Y| = n/2$ , we call  $Y$  a *balanced partition* of  $[n]$ , with the implied meaning that  $Y$  partitions  $[n]$  evenly into  $Y$  and  $[n] \setminus Y$ . The *imbalance* of a set  $S \subseteq [n]$  under  $Y$  is  $d_Y(S) := ||Y \cap S| - |S|/2|$ . Observe the useful symmetry  $d_Y(S) = d_Y([n] \setminus [S])$ , which follows from the fact that  $|Y| = n/2$ . We say  $S$  is  $\tau$ -unbalanced under  $Y$  if  $d_Y(S) \geq \tau$ .

We use the following lemma from [16].

► **Lemma 5** ([16]). *Let  $p$  be a prime, and let  $f \in \mathbb{F}_p[x_1, \dots, x_{4p}]$  be a polynomial. Suppose that for all  $Y \in \binom{[4p]}{2p}$ , it holds that  $f(\mathbf{1}_Y) = 0$ , and that there exists  $T \subseteq [4p]$  such that  $|T| = 3p$  and  $f(\mathbf{1}_T) \neq 0$ . Then  $\deg(f) \geq p$ .*

### 2.1 Hypergeometric distribution

For parameters  $N, M, k$ , where  $N \geq M$ , by  $\mathcal{H}(M, N, k)$ , we denote the distribution of  $|S \cap T|$ , where  $S$  is any fixed subset of  $[N]$  of size  $M$ , and  $T$  is a uniformly random subset of  $[N]$  of

<sup>10</sup>In our case, we need to argue that the imbalance is non-zero modulo  $p$ , which adds an extra layer of complication, although again, one which is not hard to solve.

size equal to  $k$ . Clearly,

$$\Pr[|S \cap T| = i] = \frac{\binom{M}{i} \binom{N-M}{k-i}}{\binom{N}{k}}.$$

The expected value of  $|S \cap T|$  under this distribution is equal to  $kM/N$ . We need the following tail bound of hypergeometric distribution for our proof.

► **Lemma 6** ([36]). *Let  $N, M, k$ , and  $\mathcal{H}(M, N, k)$  be as defined above. Then, for every  $t$*

$$\Pr[||S \cap T| - kM/N| \geq tk] \leq e^{-2t^2k}.$$

► **Lemma 7** (Hoeffding's inequality, [3]). *Let  $X_1, X_2, \dots, X_n$  be independent random variables taking values in  $\{0, 1\}$ . Then,*

$$\Pr \left[ \left| \sum_{i=1}^n X_i - \mathbb{E} \left[ \sum_{i=1}^n X_i \right] \right| \geq t \right] \leq 2 \exp(-2t^2/n).$$

## 2.2 Partial derivative matrix

For a circuit  $\Psi$ , we denote by  $|\Psi|$  the size of  $\Psi$ , namely, the number of gates in it. For a gate  $v$ , we denote by  $X_v$  the set of variables that occur in the subcircuit rooted at  $v$ .

Let  $X = \{x_1, \dots, x_n\}$  be a set of variables,  $Y \subseteq X$  (not necessarily of size  $n/2$ ) and let  $Z = X \setminus Y$ . For a multilinear polynomial  $f(X) \in \mathbb{F}[X]$ , we define the *partial derivative matrix* of  $f$  with respect to  $Y, Z$ , denoted  $M_{Y,Z}(f)$ , as follows: the rows of  $M$  are indexed by multilinear monomials in  $Y$ . the columns of  $M$  are indexed by multilinear monomials in  $Z$ . The entry which corresponds to  $(m_1, m_2)$  is the coefficient of the monomial  $m_1 \cdot m_2$  in  $f$ . We define  $\text{rank}_{Y,Z}(f) = \text{rank}(M_{Y,Z}(f))$ .

The following properties of the partial derivative matrix are easy to prove and well-documented (see, e.g., [31]).

► **Proposition 8.** *The following properties hold:*

1. *For every multilinear polynomial  $f(X) \in \mathbb{F}[X]$ ,  $Y \subseteq X$  and  $Z = X \setminus Y$ ,  $\text{rank}_{Y,Z}(f) \leq \min \{2^{|Y|}, 2^{|Z|}\}$ .*
2. *For every two multilinear polynomials  $f_1(X), f_2(X) \in \mathbb{F}[X]$  and for every partition  $X = Y \sqcup Z$ ,  $\text{rank}_{Y,Z}(f_1 + f_2) \leq \text{rank}_{Y,Z}(f_1) + \text{rank}_{Y,Z}(f_2)$ .*
3. *Let  $f_1 \in \mathbb{F}[X_1]$  and  $f_2 \in \mathbb{F}[X_2]$  be multilinear polynomials such that  $X_1 \cap X_2 = \emptyset$ . Let  $Y_i \subseteq X_i$  and  $Z_i = X_i \setminus Y_i$  for  $i \in \{1, 2\}$ . Set  $Y = Y_1 \cup Y_2, Z = Z_1 \cup Z_2$ . Then  $\text{rank}_{Y,Z}(f_1 \cdot f_2) = \text{rank}_{Y_1,Z_1}(f_1) \cdot \text{rank}_{Y_2,Z_2}(f_2)$ .*
4. *Let  $f(X) \in \mathbb{F}[X]$  be a multilinear polynomial such that  $X = Y \sqcup Z$  and  $|Y| = |Z| = n/2$ . Suppose  $\text{rank}_{Y,Z}(f) = 2^{n/2}$ , and let  $g = \partial f / \partial x$  for some  $x \in X$ . Then  $\text{rank}_{Y,Z}(g) = 2^{n/2-1}$ .*
5. *Let  $f(X) \in \mathbb{F}[X]$  be a multilinear polynomial of total degree  $d$ . Then for every partition  $X = Y \sqcup Z$  such that  $|Y| = |Z| = n/2$ ,  $\text{rank}_{Y,Z}(f) \leq 2^{(d+1) \log(n/2)}$ .*

## 3 Unbalancing sets under a balanced partition

In this section, we prove Theorem 3. We start by proving a special case (see Theorem 9 below) when  $n$  equals  $4p$  for some prime  $p$ , and  $\tau \geq \Omega(\log n)$ . This special case already suffices for the application to the proof of Theorem 1 (for infinitely many values of  $n$ ), and has a somewhat simpler proof. We then move on to prove the case for general  $n$  and  $\tau$ , which while being similar to the proof of Theorem 9, needs some additional ideas and care.



### 3.1 Special case: $n = 4p$ and $\tau \geq \Omega(\log n)$

► **Theorem 9.** *Let  $p$  be a large enough prime, and let  $\log p \leq \tau \leq p/1000$ . Let  $S_1, \dots, S_m \subseteq [4p]$  be sets such that for all  $i \in [m]$ ,  $100\tau \leq |S_i| \leq 4p - 100\tau$ . Further, assume that for every balanced partition  $Y$  of  $[4p]$  there exists  $i \in [m]$  such that  $d_Y(S_i) < \tau$ . Then,  $m \geq \frac{1}{2} \cdot p/\tau$ .*

We start with the following lemma, which shows that a small collection of sets can be unbalanced (modulo  $p$ ) by a partition which is very unbalanced.

► **Lemma 10.** *Let  $p$  be a large enough prime, and let  $\log p \leq \tau \leq p/1000$ . Let  $S_1, \dots, S_m \subseteq [4p]$  be sets such that for all  $i \in [m]$ ,  $100\tau \leq |S_i| \leq 2p$ . Assume further  $m \leq p$ . Then, there exists  $T \subseteq [4p]$ ,  $|T| = 3p$  such that for all  $i \in [m]$  and for all  $-\tau + 1 \leq t \leq \tau$ ,  $|S_i \cap T| \not\equiv \lfloor |S_i|/2 \rfloor + t \pmod{p}$ .*

To prove Lemma 10, we use the following two technical claims. Let  $\mu_{3/4}$  denote the probability distribution on subsets of  $[4p]$  obtained by putting each  $j \in [4p]$  in  $T$  with probability  $3/4$ , independently of all other elements.

► **Claim 11.** *For a random set  $T \sim \mu_{3/4}$ ,  $\Pr[|T| = 3p] = \Theta(1/\sqrt{p})$ .*

**Proof.** The probability that  $|T| = 3p$  is given by  $\binom{4p}{3p} \cdot (3/4)^{3p} \cdot (1/4)^p$ , which is  $\Theta(1/\sqrt{p})$ , by Stirling's approximation. ◀

► **Claim 12.** *Let  $\log p \leq \tau \leq p/1000$  and let  $S \subseteq [4p]$  such that  $100\tau \leq |S| \leq 2p$ . For a random set  $T \sim \mu_{3/4}$ , the probability that for some integer  $-\tau + 1 \leq t \leq \tau$  it holds that  $|T \cap S| = \lfloor |S|/2 \rfloor + t \pmod{p}$  is at most  $1/p^5$ .*

**Proof.** Denote  $s = |S|$ . Then  $\mathbb{E}[|T \cap S|] = 3s/4$ . We say  $T$  is bad for  $S$  if  $|T \cap S| = \lfloor s/2 \rfloor + t + kp$  for some  $-\tau \leq t \leq \tau + 1$  and  $k \in \mathbb{Z}$ . We claim this in particular implies that  $||T \cap S| - 3s/4| \geq s/5$ . Indeed, since  $|T \cap S|$  is an integer in the interval  $[0, 2p]$ , and by the bounds on  $s$ , the only cases needed to be analyzed are  $k = 0, \pm 1$ .

If  $|T \cap S| = \lfloor s/2 \rfloor + t - p$ , then clearly  $|T \cap S| \leq \lfloor s/2 \rfloor$  which implies the statement.

If  $|T \cap S| = \lfloor s/2 \rfloor + t + p$ , then, as  $s \leq 2p$  and  $\tau \leq s/100$ ,

$$|T \cap S| - 3s/4 \geq -s/4 - 1 + t + p \geq p/2 + t - 1 \geq s/4 + t - 1 \geq s/5$$

(The “ $-1$ ” accounts for the fact that  $s/2$  might not be an integer).

Finally, if  $|T \cap S| = \lfloor s/2 \rfloor + t$ , it holds that

$$|T \cap S| \leq s/2 + \tau \leq s/2 + 2s/100,$$

which again implies the statement.

By Chernoff Bound (see, e.g., [3]),  $\Pr[||T \cap S| - 3s/4| \geq s/5] \leq 2^{-|S|/20} \leq 1/p^5$ , hence  $T$  is bad for  $S$  with at most that probability. ◀

The proof of Lemma 10 is now fairly immediate.

**Proof of Lemma 10.** Pick  $T \sim \mu_{3/4}$ . By Claim 11,  $|T| = 3p$  with probability  $\Theta(1/\sqrt{p})$ . Recall that  $T$  is bad for  $S_i$  if  $|T \cap S_i| = \lfloor |S_i|/2 \rfloor + t \pmod{p}$  for  $t \in \{-\tau + 1, \dots, \tau\}$ . By Claim 11, for each  $S_i$ ,  $T$  is bad for  $S_i$  with probability at most  $1/p^5$ . Hence, the probability that there exists  $i \in [m]$  such that  $T$  is bad for  $S_i$  is at most  $m/p^5 \leq 1/p^4$ .

It follows that with probability at most  $1 - \Theta(1/\sqrt{p}) + 1/p^4 < 1$ , either  $|T| \neq 3p$  or  $T$  is bad for some  $S_i$ , and hence there exists a selection of  $T$  such that  $|T| = 3p$  and  $T$  is good for all  $S_i$ 's. ◀



We are now ready to prove Theorem 9.

**Proof of Theorem 9.** Let  $S_1, \dots, S_m$  be a collection of sets as stated in the theorem. Since  $d_Y(S_j) = d_Y([n] \setminus S_j)$ , we can assume without loss of generality, by possibly replacing a set with its complement, that  $|S_j| \leq 2p$  for all  $j \in [m]$ . We may further assume  $m \leq p$  as otherwise the statement directly follows. For  $j \in [m]$ , define the following polynomials over  $\mathbb{F}_p$ :

$$B_j(x_1, \dots, x_{4p}) = \prod_{t=-\tau+1}^{\tau} (\langle x, \mathbf{1}_{S_j} \rangle - \lfloor |S_j|/2 \rfloor - t),$$

where  $x = (x_1, \dots, x_{4p})$  and  $\langle u, v \rangle = \sum u_i v_i$  is the usual inner product. Further, define

$$f(x_1, \dots, x_{4p}) = \prod_{j=1}^m B_j(x_1, \dots, x_{4p}),$$

as a polynomial over  $\mathbb{F}_p$ .

By assumption, for every  $Y \in \binom{[4p]}{2p}$ ,  $f(\mathbf{1}_Y) = 0$ . This follows because  $\langle \mathbf{1}_Y, \mathbf{1}_{S_j} \rangle = |Y \cap S_j|$ , and by assumption, for some  $j$  it holds that  $d_Y(S_j) < \tau$ , so it must be that  $|Y \cap S_j| - \lfloor |S_j|/2 \rfloor \in \{-\tau + 1, \dots, 0, \dots, \tau\}$ , so that  $B_j(\mathbf{1}_Y) = 0$ .

Furthermore, Lemma 10 guarantees the existence of a set  $T \in \binom{[4p]}{3p}$  such that  $f(\mathbf{1}_T) \neq 0$ , as the set  $T$  from Lemma 10 satisfies the property that  $(\langle \mathbf{1}_T, \mathbf{1}_{S_j} \rangle - \lfloor |S_j|/2 \rfloor - t) \neq 0 \pmod p$  for all  $-\tau + 1 \leq t \leq \tau$  and for all  $j \in [m]$ .

By Lemma 5,  $\deg(f) \geq p$ , and by construction,  $\deg(f) \leq 2\tau \cdot m$ , which implies the desired lower bound on  $m$ .  $\blacktriangleleft$

In the full version of the paper we extend Theorem 9 for a more general range of parameters, by proving the following.

**► Theorem 13.** *Let  $n$  be a large enough even natural number, and let  $\tau \in \{1, 2, \dots, n/10^6\}$  be a parameter. Let  $S_1, S_2, \dots, S_m \subseteq [n]$  be sets such that for each  $i \in [m]$ ,  $2\tau \leq |S_i| \leq n - 2\tau$ . Furthermore, assume that for every balanced partition  $Y$  of  $[n]$ , there exists an  $i$  such that  $d_Y(S_i) < \tau$ . Then,  $m \geq \frac{1}{10^5} \cdot n/\tau$ .*

The proof of Theorem 13 appears in the full version of the paper [2]. We remark that Theorem 9 suffices for the application to circuit lower bounds.

## 4 Syntactically Multilinear Arithmetic Circuits

In this section, for the sake of completeness, we review the arguments of Raz, Shpilka and Yehudayoff [31], and show how Theorem 9 implies a lower bound of  $\Omega(n^2/\log^2 n)$ . We mostly refer for [31] for the proofs.

Specifically, we will show the following.

**► Theorem 14.** *Let  $n$  be an even integer, and  $X = \{x_1, \dots, x_n\}$ . Let  $f(X) \in \mathbb{F}[X]$  be a multilinear polynomial such that for every balanced partition  $X = Y \sqcup Z$ ,  $\text{rank}_{Y,Z}(f) = 2^{n/2}$ . Let  $\Psi$  be a syntactically multilinear circuit computing  $f$ . Then  $|\Psi| = \Omega(n^2/\log^2 n)$ .*

The first step in proof of Theorem 14 is to show that if  $f$  is computed by a syntactically multilinear circuit of size  $s$ , then there exists a syntactically multilinear circuit of size  $O(s)$  that computes all the first-order partial derivatives of  $f$ , with the additional important property that for each  $i$ , the variable  $x_i$  does not appear in the subcircuit rooted at the output gate which computes  $\partial f/\partial x_i$ .

► **Theorem 15** ([31], Theorem 3.1). *Let  $\Psi$  be a syntactically multilinear circuit over a field  $\mathbb{F}$  and the set of variables  $X = \{x_1, \dots, x_n\}$ . Then, there exists a syntactically multilinear circuit  $\Psi'$ , over  $\mathbb{F}$  and  $X$ , such that:*

1.  $\Psi'$  computes all  $n$  first-order partial derivatives  $\partial f / \partial x_i$ ,  $i \in [n]$ .
2.  $|\Psi'| \leq 5|\Psi|$ .
3.  $\Psi'$  is syntactically multilinear.
4. For every  $i \in [n]$ ,  $x_i \notin X_{v_i}$ , where  $v_i$  is the gate in  $\Psi'$  computing  $\partial f / \partial x_i$ .

*In particular, if  $v$  is a gate in  $\Psi'$ , then it is connected by a directed path to at most  $n - |X_v|$  output gates.*

The proof of Theorem 15 appears in [31], and mostly follows the classical proof of Baur and Strassen [5] of the analogous result for general circuits, with additional care in order to guarantee the last two properties.

Next we define two types of gates in a syntactically multilinear arithmetic circuits.

► **Definition 16.** Let  $\Phi$  be a syntactically multilinear arithmetic circuit. Define  $\mathcal{L}(\Phi, k)$ , the set of lower-leveled gates in  $\Phi$ , by

$$\mathcal{L}(\Phi, k) = \{u : u \text{ is a gate in } \Phi, k < |X_u| < n - k, \text{ and } u \text{ has a parent } v \text{ with } |X_v| \geq n - k\}.$$

Define  $\mathcal{U}(\Phi, k)$ , the set of upper-leveled gates in  $\Phi$ , by

$$\mathcal{U}(\Phi, k) = \{v : v \text{ is a gate in } \Phi, |X_v| \geq n - k, \text{ and } v \text{ has a child } u \in \mathcal{L}(\Phi, k)\}.$$

The following lemma shows that if the set of lower-leveled gates is small, then there exists a partition  $X = Y \sqcup Z$  under which the polynomial computed by the circuit is not of full rank.

► **Lemma 17.** *Let  $\Phi$  be a syntactically multilinear arithmetic circuit over  $\mathbb{F}$  and  $X = \{x_1, \dots, x_n\}$ , for an even integer  $n$ , computing  $f$ . Let  $\tau = 3 \log n$  and  $\mathcal{L} = \mathcal{L}(\Phi, 100\tau)$ . If  $|\mathcal{L}| < n / (10^5 \tau)$ , then there exists a partition  $X = Y \sqcup Z$  such that  $\text{rank}_{Y,Z}(f) < 2^{n/2-1}$ .*

We first sketch how Theorem 14 follows from Lemma 17. The proof is identical to the proof given in [31] with slightly different parameters.

**Proof of Theorem 14 assuming Lemma 17.** Let  $\Psi'$  be the arithmetic circuit computing all  $n$  first-order partial derivatives of  $f$ , given by Theorem 15. Set  $\tau = 3 \log n$  and let  $\mathcal{L} = \mathcal{L}(\Psi', 100\tau)$  and  $\mathcal{U} = \mathcal{U}(\Psi', 100\tau)$  as in Definition 16.

Denote  $f_i = \partial f / \partial x_i$  and let  $v_i$  be the gate in  $\Psi'$  computing  $f_i$ , and  $\Psi'_i$  be the subcircuit of  $\Psi'$  rooted at  $v_i$ . Let  $\mathcal{L}_i = \mathcal{L}(\Psi'_i, 100\tau)$ . It is not hard to show (see [31]) that  $\mathcal{L}_i \subseteq \mathcal{L}$ , and by Lemma 17 and item 4 in Proposition 8, it follows that  $|\mathcal{L}_i| \geq n / (10^5 \tau)$ .

For every gate  $v$  in  $\Psi'$  define  $C_v = \{i \in [n] : v \text{ is a gate in } \Psi'_i\}$  to be the set of indices  $i$  such that there exists a directed path from  $v$  to the output gate computing  $f_i$ . For  $i \in [n]$ , let  $\mathcal{U}_i = \{u \in \mathcal{U} : u \text{ is a gate in } \Psi'_i\}$ , so that  $\sum_{u \in \mathcal{U}} C_u = \sum_{i \in [n]} |\mathcal{U}_i|$ .

Since the fan-in of each gate is at most two,  $|\mathcal{L}_i| \leq 2|\mathcal{U}_i|$ , and since every  $u \in \mathcal{U}$  satisfies  $|X_u| \geq n - 100\tau$ , it follows by Theorem 15 that  $|C_u| \leq 100\tau$ . Thus, we get

$$n \cdot \frac{n}{10^5 \tau} \leq \sum_{i \in [n]} |\mathcal{L}_i| \leq 2 \sum_{i \in [n]} |\mathcal{U}_i| = 2 \sum_{u \in \mathcal{U}} C_u \leq 2|\mathcal{U}| \cdot 100\tau.$$

By item 2 in Theorem 15, and  $\tau = 3 \log n$ ,

$$|\Psi| = \Omega(|\Psi'|) = \Omega(|\mathcal{U}|) = \Omega\left(\frac{n^2}{\log^2 n}\right). \quad \blacktriangleleft$$

It remains to prove Lemma 17. As the proof mostly appears in [31], we only sketch the main steps.

**Proof sketch of Lemma 17.** Suppose  $\mathcal{L} \leq n/(10^5\tau)$ . By applying Theorem 13 to the family of sets  $\{X_v : v \in \mathcal{L}\}$ , it follows that there exists a balanced partition  $Y \sqcup Z$  of  $X$  such that  $X_v$  is  $\tau$ -unbalanced for every gate  $v \in \mathcal{L}$  (one could get slightly improved constants in the case  $n = 4p$  by applying Theorem 9).

The proof now proceeds in the exact same manner as the proof of Lemma 5.2 in [31]. In Proposition 5.5 of [31], it is shown that one can write

$$f = \sum_{i \in [\ell]} g_i h_i + g,$$

where  $\mathcal{L} = \{v_1, \dots, v_\ell\}$ ,  $h_i$  is the polynomial computed at  $v_i$ , and the set of variables appearing in  $g_i$  is disjoint from  $X_{v_i}$ .

In Claim 5.7 of [31], it is shown that for every  $i \in [\ell]$ ,  $\text{rank}_{Y,Z}(g_i h_i) \leq 2^{n/2-\tau}$ . This uses the fact that  $X_{v_i}$  is  $\tau$ -unbalanced, the upper bound in item 1 in Proposition 8, and item 3 in the same proposition.

In Proposition 5.8 of [31], it is shown (with the necessary change of parameters) that the degree of  $g$  is at most  $200\tau$ .

Thus, by the fact that  $\tau = 3 \log n$ , item 5 and item 2 of Proposition 8, it follows that for large enough  $n$ ,

$$\text{rank}_{Y,Z}(f) \leq \ell \cdot 2^{n/2-\tau} + 2^{\tau^3} < 2^{n/2-1}. \quad \blacktriangleleft$$

## 4.1 An explicit full-rank polynomial

In this section, for the sake of completeness, we give a construction of a polynomial which is full-rank under any partition of the variables.

► **Construction 18** (Full rank polynomial, [31]). *Let  $n$  be an even integer, and let  $\mathcal{W} = \{\omega_1, \dots, \omega_n\}$  and  $X = \{x_1, \dots, x_n\}$  be sets of variables. For a set  $B \in \binom{[n]}{n/2}$ , denote by  $i_1 < \dots < i_{n/2}$  the elements of  $B$  in increasing order, and by  $j_1 < \dots < j_{n/2}$  the elements of  $[n] \setminus B$  in increasing order. Define  $r_B = \prod_{\ell \in B} \omega_\ell$ , and  $g_B = \prod_{\ell \in [n/2]} (x_{i_\ell} + x_{j_\ell})$ .*

*Finally, define*

$$f = \sum_{B \in \binom{[n]}{n/2}} r_B g_B.$$

► **Claim 19** ([31]). *For  $f$  from Construction 18, it holds that for every balanced partition of  $X = Y \sqcup Z$ ,  $\text{rank}_{Y,Z}(f) = 2^{n/2}$ , where the rank is taken over  $\mathbb{F}(\mathcal{W})$ .*

We give a proof which is shorter and simpler than the one given in [31].

**Proof of Claim 19.** Fix a balanced partition  $X = Y \sqcup Z$ , and consider the matrix  $M_{Y,Z}(f)$  where  $f$  is interpreted as a polynomial in  $f \in (\mathcal{F}[\mathcal{W}])[X]$  (that is, the rows and columns of the matrix are indexed by  $X$  variables and its entries are polynomials in  $\mathcal{W}$ ). We want to show that  $\det(M_{Y,Z}(f)) \in \mathbb{F}[\mathcal{W}]$  is a non-zero polynomial. Fix  $\omega_i = 1$  if  $i \in Y$  and  $\omega_i = 0$  otherwise. Under this restriction,  $f = g_Y$ . It is also not hard to see that  $\det(M_{Y,Z}(g_Y)) \neq 0$ , since this is a permutation matrix (this also follows from item 3 of Proposition 8). Thus,  $\det(M_{Y,Z}(f))$  evaluates to a non-zero value under this setting of the variables  $\mathcal{W}$ , which implies it a non-zero polynomial. ◀

► **Corollary 20.** *Every syntactically multilinear circuit computing the polynomial  $f$  has size at least  $\Omega(n^2/\log^2 n)$ .*

The polynomial  $f$  in Construction 18 is in the class VNP of explicit polynomials, but it is not known whether there exists a polynomial size multilinear circuit for  $f$ .

Raz and Yehudayoff [32] constructed a full-rank polynomial  $g \in \mathbb{F}[X, \mathcal{W}']$  that has a syntactically multilinear circuit of size  $O(n^3)$ . Their construction also uses a set of auxiliary variables  $\mathcal{W}'$  of size  $O(n^3)$ . Thus, if one measures the complexity as a function of  $|X| \cup |\mathcal{W}'|$ , the quadratic lower bound of Theorem 14 is meaningless, because a lower bound of  $\Omega(n^3)$  holds trivially. However, we believe that since the rank is taken over  $\mathbb{F}(\mathcal{W}')$ , it is only fair to consider computations over  $\mathbb{F}(\mathcal{W}')$ , where any rational expression in the variables of  $\mathcal{W}'$  is merely a field constant. Thus, in this setting, an input gate can be labeled by an arbitrarily complex rational function in the variables of  $\mathcal{W}'$ , and the complexity is measured as a function of  $|X|$  alone. In this model the lower bound of Theorem 14 is meaningful, and furthermore, this example shows that the partial derivative matrix technique cannot prove an  $\omega(n^3)$  lower bound.

---

## References

- 1 Noga Alon, E. E. Bergmann, Don Coppersmith, and Andrew M. Odlyzko. Balancing sets of vectors. *IEEE Trans. Information Theory*, 34(1):128–130, 1988. doi:10.1109/18.2610.
- 2 Noga Alon, Mrinal Kumar, and Ben Lee Volk. An almost quadratic lower bound for syntactically multilinear arithmetic circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:124, 2017. URL: <https://eccc.weizmann.ac.il/report/2017/124>.
- 3 Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley Publishing, 4th edition, 2016. URL: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1119061954.html>.
- 4 Richard P. Anstee, Lajos Rónyai, and Attila Sali. Shattering news. *Graphs and Combinatorics*, 18(1):59–73, 2002. doi:10.1007/s003730200003.
- 5 Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983. doi:10.1016/0304-3975(83)90110-X.
- 6 Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inf. Process. Lett.*, 18(3):147–150, 1984. doi:10.1016/0020-0190(84)90018-8.
- 7 Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity and beyond. *Foundations and Trends in Theoretical Computer Science*, 6(1-2):1–138, 2011. doi:10.1561/04000000043.
- 8 L. Csanky. Fast parallel matrix inversion algorithms. *SIAM J. Comput.*, 5(4):618–623, 1976. doi:10.1137/0205040.
- 9 H. Enomoto, Peter Frankl, N. Ito, and K. Nomura. Codes with given distances. *Graphs and Combinatorics*, 3(1):25–38, 1987. doi:10.1007/BF01788526.
- 10 Jeffrey B. Farr and Shuhong Gao. Computing gröbner bases for vanishing ideals of finite sets of points. In Marc P. C. Fossorier, Hideki Imai, Shu Lin, and Alain Poli, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 16th International Symposium, AAECC-16, Las Vegas, NV, USA, February 20-24, 2006, Proceedings*, volume 3857 of *Lecture Notes in Computer Science*, pages 118–127. Springer, 2006. doi:10.1007/11617983\_11.
- 11 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 128–135. ACM, 2014. doi:10.1145/2591796.2591824.

- 12 Peter Frankl and Vojtěch Rödl. Forbidden intersections. *Trans. Amer. Math. Soc.*, 300(1):259–286, 1987. doi:10.2307/2000598.
- 13 Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 577–582. ACM, 1998. doi:10.1145/276698.276872.
- 14 Dima Grigoriev and Alexander A. Razborov. Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Appl. Algebra Eng. Commun. Comput.*, 10(6):465–487, 2000. doi:10.1007/s002009900021.
- 15 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Satharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, 2014. doi:10.1145/2629541.
- 16 Gábor Hegedűs. Balancing sets of vectors. *Studia Sci. Math. Hungar.*, 47(3):333–349, 2010. doi:10.1556/SScMath.2009.1134.
- 17 Gábor Hegedűs and Lajos Rónyai. Gröbner bases for complete uniform families. *J. Algebraic Combin.*, 17(2):171–180, 2003. doi:10.1023/A:1022934815185.
- 18 Maurice J. Jansen. Lower bounds for syntactically multilinear algebraic branching programs. In Edward Ochmanski and Jerzy Tyszkiewicz, editors, *Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008, Torun, Poland, August 25-29, 2008, Proceedings*, volume 5162 of *Lecture Notes in Computer Science*, pages 407–418. Springer, 2008. doi:10.1007/978-3-540-85238-4\_33.
- 19 K. Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985. doi:10.1137/0214050.
- 20 Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 61–70. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.15.
- 21 Donald E. Knuth. Efficient balanced codes. *IEEE Trans. Information Theory*, 32(1):51–53, 1986. doi:10.1109/TIT.1986.1057136.
- 22 Mrinal Kumar. A quadratic lower bound for homogeneous algebraic branching programs. In Ryan O’Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPICs*, pages 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CCC.2017.19.
- 23 Mrinal Kumar and Ramprasad Satharishi. An exponential lower bound for homogeneous depth-5 circuits over finite fields. In Ryan O’Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPICs*, pages 31:1–31:30. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CCC.2017.31.
- 24 Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 364–373. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.46.
- 25 Meena Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chicago J. Theor. Comput. Sci.*, 1997, 1997. Preliminary version in the *8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1997)*. URL: <http://cjtcs.cs.uchicago.edu/articles/1997/5/contents.html>.
- 26 Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418. ACM, 1991. doi:10.1145/103418.103462.

- 27 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. doi:10.1007/BF01294256.
- 28 Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. doi:10.4086/toc.2006.v002a006.
- 29 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009. doi:10.1145/1502793.1502797.
- 30 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010. doi:10.4086/toc.2010.v006a007.
- 31 Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. Comput.*, 38(4):1624–1647, 2008. doi:10.1137/070707932.
- 32 Ran Raz and Amir Yehudayoff. Balancing syntactically multilinear arithmetic circuits. *Computational Complexity*, 17(4):515–535, 2008. doi:10.1007/s00037-008-0254-0.
- 33 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009. doi:10.1007/s00037-009-0270-8.
- 34 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, <https://github.com/dasarpmar/lowerbounds-survey/>, 2016. URL: <https://github.com/dasarpmar/lowerbounds-survey/releases/>.
- 35 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:10.1561/04000000039.
- 36 Matthew Skala. Hypergeometric tail inequalities: ending the insanity. *arXiv preprint arXiv:1311.5939*, 2013. URL: <https://arxiv.org/abs/1311.5939>.
- 37 Volker Strassen. Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, 1973. doi:10.1007/BF01436566.

# Hardness Amplification for Non-Commutative Arithmetic Circuits

Marco L. Carmosino<sup>1</sup>

Department of Computer Science, University of California San Diego, La Jolla, CA, USA  
marco@ntime.org

Russell Impagliazzo<sup>2</sup>

Department of Computer Science, University of California San Diego, La Jolla, CA, USA  
russell@cs.ucsd.edu

Shachar Lovett<sup>3</sup>

Department of Computer Science, University of California San Diego, La Jolla, CA, USA  
slovett@ucsd.edu

Ivan Mihajlin<sup>4</sup>

Department of Computer Science, University of California San Diego, La Jolla, CA, USA  
imikhail@cs.ucsd.edu

---

## Abstract

We show that proving mildly super-linear lower bounds on non-commutative arithmetic circuits implies exponential lower bounds on non-commutative circuits. That is, non-commutative circuit complexity is a threshold phenomenon: an apparently weak lower bound actually suffices to show the strongest lower bounds we could desire.

This is part of a recent line of inquiry into why arithmetic circuit complexity, despite being a heavily restricted version of Boolean complexity, still cannot prove super-linear lower bounds on general devices. One can view our work as positive news (it suffices to prove weak lower bounds to get strong ones) or negative news (it is as hard to prove weak lower bounds as it is to prove strong ones). We leave it to the reader to determine their own level of optimism.

**2012 ACM Subject Classification** Theory of computation → Algebraic complexity theory

**Keywords and phrases** arithmetic circuits, hardness amplification, circuit lower bounds, non-commutative computation

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.12

## 1 Introduction

Arithmetic circuits are a natural computational model for computing polynomials, which has been extensively studied in complexity theory. Most of the research is focused on proving lower bounds. Namely, showing that certain “hard” polynomials (such as the permanent, which is complete for an arithmetic version of NP [17]) require large arithmetic circuits. Despite much research, strong lower bounds are only known for restricted families of circuits, such as circuits of fixed depth, multi-linear circuits, or monotone circuits. For general arithmetic circuits, the best lower bound known is still the classical result of Baur-Strassen [5]

---

<sup>1</sup> Supported by the Simons Foundation

<sup>2</sup> Supported by the Simons Foundation

<sup>3</sup> Supported by NSF CAREER award 1350481 and CCF award 1614023

<sup>4</sup> Supported by the Simons Foundation



© Marco L. Carmosino, Russell Impagliazzo, Shachar Lovett, and Ivan Mihajlin; licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 12; pp. 12:1–12:16



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





who showed that to compute  $X_1^n + \dots + X_n^n$  one needs an arithmetic circuit of size  $\Omega(n \log n)$ . We refer to the recent survey [16] and the references within for details about these many works.

An interesting restriction of the arithmetic model, which is the focus on this paper, is that of *non-commutative polynomials* and correspondingly *non-commutative circuits*. A non-commutative polynomial over a field  $\mathbb{F}$  in variables  $X_1, \dots, X_n$ , is a linear combination of monomials, except that here monomials are defined as words over the variables. Otherwise put, variables do not commute, so the order of variables in a monomial is important. Despite this severe restriction, the non-commutative setting maintains complexity-theoretic structure: the permanent is complete for non-commutative arithmetic NP [10] (VNP), and natural polynomials are complete for non-commutative arithmetic P [3] (VP). The hope is that it will be easier to prove strong lower bounds against non-commutative circuits, as various cancellations that occur in standard (commutative) arithmetic circuits crucially depend on the commutativity of the variables. For example, the  $n \times n$  determinant can be computed by a  $O(n^3)$  arithmetic circuit, but to the best of our knowledge, there is no non-commutative arithmetic circuit for determinant of size  $n^{o(n)}$ . Moreover if determinant can be computed by polynomial size non-commutative circuits then  $VP = VNP$  [4]<sup>5</sup>.

If one restricts attention further to non-commutative *formulas*, then our understanding is dramatically better. A fundamental result in this area is a theorem of Nisan [14], who proved exponential lower bounds on non-commutative formulas. For example, his technique applied to the  $n \times n$  permanent (or also the  $n \times n$  determinant) shows that any non-commutative formula computing either of them requires size  $\Omega(2^n)$ . On the other hand, no lower bounds for non-commutative circuits are known which are better than these known for standard commutative circuits. This dichotomy leads to the main problem motivating this paper, posed by [11]:

Why do we have exponential lower bounds for non-commutative formulas, but only marginally super-linear lower bounds for non-commutative circuits?

The main message of the this paper is that weak lower bounds for non-commutative circuits can be “amplified” to arbitrarily large polynomial, or even exponential, lower bounds for non-commutative circuits. One can view this as positive news (it suffices to prove weak lower bounds to get strong ones) or negative news (it is as hard to prove weak lower bounds as it is to prove strong ones). We leave it to the reader to make their own choice. Below, we state the formal versions of our main results.

We recall the standard notation that  $\omega$  is the best known exponent for matrix multiplication, where the best known upper bounds on it are  $\omega \leq 2.374$  due to [13]. Our first theorem is that polynomial lower bounds better than  $\Omega(n^{\omega/2})$  for any non-commutative polynomial in  $n$  variables of polynomial degree can be lifted to arbitrary polynomial lower bounds.

► **Theorem 1.** *Let  $\varepsilon > 0$ . Assume that there exists an explicit non-commutative polynomial in  $n$  variables of degree  $\text{poly}(n)$ , such that any non-commutative circuit computing it requires size  $\Omega(n^{\omega/2+\varepsilon})$ .*

*Then, for any  $c > 1$ , there exists another explicit polynomial in  $m$  variables of degree  $\text{poly}(m)$ , such that any non-commutative circuit computing it requires size  $\Omega(m^c)$ .*

Some authors [11] had suggested that for non-commutative polynomials number of variables might be not the best parameter. In contrast with the commutative setting, one

---

<sup>5</sup> Formally, one needs to define a non-commutative determinant, by inducing some natural order on the variables in monomials of the standard commutative permanent.



can focus on polynomials with a constant number of variables, using the degree as a measure. The difference comes from the fact that there are  $2^d$  different non-commutative monomials on 2 variables of degree  $d$  versus  $d + 1$  for the commutative case. For this regime, the best known lower bounds are of the form  $\Omega(\log(d))$  where  $d$  is the degree. Theorem 1 states that if we have good enough lower bound to start with, we can give a family of polynomials of complexity  $\Omega(d)$ . We will, however, use number of variables as our measure, as we will be dealing with constant degree polynomials.

Our second theorem shows that proving lower bounds better than  $\Omega(n^{\omega/2})$  for any *constant degree* non-commutative polynomial in  $n$  variables can be lifted to exponential lower bounds. This may help to explain why no super-linear lower bound for a constant-degree non-commutative polynomial is currently known. The polynomial that we start with in this case must be *explicit*, a notion of uniformity described in section 2.

► **Theorem 2.** *Let  $\varepsilon > 0$ . Assume that there exists an explicit non-commutative polynomial in  $n$  variables of constant degree, such that any non-commutative circuit computing it requires size  $\Omega(n^{\omega/2+\varepsilon})$ .*

*Then, for some  $c > 0$ , there exists another explicit polynomial in  $m$  variables of degree  $\text{poly}(m)$ , such that any non-commutative circuit computing it requires size  $\exp(m^c)$ .*

Here is one way to interpret our results, which we find intriguing: proving any super-linear lower bound  $\Omega(n^{1+\varepsilon})$  against non-commutative circuits would imply one of two things: (i) an arbitrarily large polynomial lower bound (or even better) against non-commutative circuits; or (ii) a proof that  $\omega > 2$ , namely, a super-linear lower bound for (standard, commutative) matrix multiplication.

## 1.1 Technique

Our main technical result is a lifting theorem, which allows us to amplify lower bounds against non-commutative circuits, by reducing the number of variables without hurting the lower bound too much.

Let  $f$  be a non-commutative polynomial over variables  $X_1, \dots, X_n$ . Fix a constant integer  $r \geq 1$  and assume that  $n = m^r$ . Define new variables  $Y_{i,j}$  where  $i = 0, \dots, (r - 1)$  and  $j = 0, \dots, (m - 1)$ . We will encode each  $X_i$  as a monomial  $Y_{0,a_0} Y_{1,a_1} \dots Y_{(r-1),a_{r-1}}$ , where  $a_0 \dots a_{r-1}$  is the encoding of  $i$  in base- $m$ . Let  $E(f)$  denote the polynomial obtained by doing this replacement to each variable in  $f$ . Note that  $E(f)$  is a polynomial over the  $rm$  variables  $\{Y_{i,j}\}$  of degree  $\deg(E(f)) = r \deg(f)$ .

Our main technical lemma (lemma 4) shows that any non-commutative circuit  $C$  which computes  $E(f)$  can be transformed to another non-commutative circuit  $C'$  which computes  $f$ . We think of this as “decoding” the circuit for the encoding  $E(f)$  to a circuit for  $f$ . Moreover, the size of  $C'$  is not much larger than  $C$ . The optimal parameters are achieved by taking  $r = 3$ , using fast matrix multiplication; they give that  $\text{size}(C') \leq \text{size}(C) \cdot n^{\omega/3}$ .

Otherwise put, if  $f$  requires arithmetic circuits of size  $s$ , then  $E(f)$  requires arithmetic circuits of size  $s \cdot n^{-\omega/3}$ . However,  $E(f)$  has only marginally higher degree and many fewer variables  $m = n^{1/3}$ . Applying this idea iteratively, we make progress as long as  $s \gg n^{\omega/2}$ . This implies both of our main theorems (Theorem 1 and Theorem 2).

For our generic technique to go through, we need to “massage” non-commutative circuits for  $E(f)$  so that they can be “decoded” into non-commutative circuits for  $f$ . Basically, we want all the gates in the circuit to compute polynomials over  $\{Y_{i,j}\}$  that are encoding of polynomials over  $\{X_i\}$ . We accomplish that by several rounds of simplification of the structure of the circuit. This can be seen as an analog to the homogenization process performed on algebraic circuits, except that in our case, the process is more delicate.

## 1.2 Related Work

This work parallels that of Hrubeš, Wigderson and Yehudayoff [11]. They showed that if any *explicit* degree 4 polynomial has a strong enough super-linear lower bound on *width*, then this lower bound can be lifted to an exponential circuit lower bound for a non-commutative polynomial. We refer the reader to the original paper for the formal definition of width. To compare these results with ours, note that implicit in [11] is the relationship  $\frac{s}{n^2} \leq w(P) \leq O(ns)$ , where  $P$  is any degree 4 polynomial,  $w$  is the “width” of this polynomial, and  $s$  is the minimal size of a circuit computing  $P$ . Thus, [11] shows that any super-cubic circuit lower bound for an explicit polynomial of degree 4 implies exponential circuit lower bounds for some explicit polynomial.

We show that one can start from circuit lower bounds of the form  $n^{\frac{s}{2}}$  against *any* constant degree polynomial and lift to exponential circuit lower bounds. Moreover, even lower bounds against higher degree polynomials can be lifted.

As in [11], we give new structural properties of non-commutative circuits computing restricted polynomials. The restrictions of [11] force polynomials to form monomials by selecting each variable from some sets of variables that always appear in a fixed order of some fixed length. Our restrictions allow the sets of variables to have a *periodic* ordering, according to  $\mathbb{Z}/r$  for some  $r$ . This allows our structures to easily generalize to higher degrees.

An encoding of variables similar to our lifting was used previously in [2], as a step in randomized polynomial identity testing for *sparse* non-commutative circuits. The work of [3] uses a similar double-indexed “positional” encoding of monomials, to establish a transfer theorem from “ $f$  is complete for a non-commutative algebraic class” to “ $\text{decoded}(f)$  is complete for a commutative algebraic class.”

There has been a great deal of recent interest in understanding why it is hard to prove lower bounds in the arithmetic setting, even though it is more restricted than the Boolean setting. Analogs of the Natural Proofs barrier of [15] have been proposed in [7] and [8], and an unconditional barrier for rank-based methods was just shown by [6]. Our result is most similar to the “chasm” family of results [1, 12, 9]: they show that one “only” needs to prove depth-3 lower bounds to prove general super-polynomial lower bounds. We show that, in the non-commutative case, one “only” needs to prove mildly super-linear lower bounds to prove super-polynomial lower bounds.

## Organization

In section 2, we formally define lifting, state the key “circuit decoding” lemma, and show how the results follow. In section 3, we prove the decoding lemma by giving new structural results about non-commutative circuits.

## 2 Preliminaries

### Polynomials and Circuits

Let  $X = \{x_1, \dots, x_n\}$  be a set of variables and let  $\mathbb{F}$  be a field. We denote by  $\mathbb{F}\langle X \rangle$  the set of non-commutative polynomials over  $X$  with coefficients in  $\mathbb{F}$ . These polynomials sum over monomials that are *words* over  $X$ , because multiplication of variables does not commute. We define circuits computing polynomials from  $\mathbb{F}\langle X \rangle$  in the natural way: as directed acyclic graphs with internal nodes (gates) labeled by  $+$ ,  $\times$  and leaves labeled by  $x \in X$  or field elements. Each  $+$ ,  $\times$  gate has two children, and each  $\times$  gate has distinguished left and right children. Denote by  $AC(f)$  the arithmetic complexity of a non-commutative polynomial  $f$ , as the minimal number of gates in a non-commutative circuit computing  $f$ .

### Explicitness

It is easy to prove that some polynomials require exponential size circuits. So we restrict ourselves to the set of explicit polynomials. A polynomial  $f$  is *explicit* if and only if each of its coefficients can be computed in polynomial time in the description length of a monomial. Thus, the coefficients of an explicit constant degree polynomial can be computed in polylogarithmic time.

## 3 Lifting Polynomials

We define polynomial lifting and give basic properties. Unless otherwise stated, all our polynomials and circuits are non-commutative. We consider both singly-indexed variables  $X = \{x_i\}$  and doubly-indexed variables  $Y = \{y_{i,j}\}$ . To ease work over  $Y$ , define sets  $Y_i$  as  $\{y_{i,j}\}_{j \in \mathbb{N}}$ , the sets of all  $y$  variables with first index  $i$ . We use the notation  $\mathbb{F}\langle X \rangle = \mathbb{F}\langle x_1, \dots, x_n \rangle$  to denote non-commutative polynomials over the  $X$  variables, and analogously for polynomials over the  $Y$  variables.

Lifting takes a polynomial over the  $X$  variables to a polynomial over the  $Y$  variables. Starting with  $f \in \mathbb{F}\langle x_1, \dots, x_n \rangle$ , we replace each  $x_i$  by a product of  $y$  variables that encodes  $i$  in base  $n^{1/r}$ , rounding up to ensure that  $n^{1/r}$  is an integer. To simplify notations, we will always assume that  $n = m^r$  for some integer  $m$ , so no rounding will be necessary. Since the  $y$  variables do not commute, the resulting polynomial can easily be mapped back to  $f$  by reading “sub-words” of monomials base  $n^{1/r}$  to recover which  $x_i$  a string of  $y$  variables represents. To formalize this below, we use  $\text{digit}(t, i, j)$  to refer to the  $j$ th digit of the base- $t$  representation of  $i$ .

► **Definition 3** (Lifting). Let  $f \in \mathbb{F}\langle X \rangle$ . Define  $L_r(f) \in \mathbb{F}\langle Y \rangle$  by applying the following map to each variable of  $f$ :

$$x_i \rightarrow \prod_{j=0}^{(r-1)} y_{j, \ell_j} \quad \text{where } \ell_j = \text{digit}(n^{1/r}, i, j)$$

This means that  $L_r(f)$  will be over  $rn^{1/r}$  variables  $(y_{0,1}, \dots, y_{r-1, n^{1/r}})$ . If the degree of  $f$  is  $d$ , then the degree of  $L_r(f)$  is  $dr$ . So lifting shrinks the number of variables while increasing the degree.

Lifting preserves explicitness. Suppose we want to compute a coefficient of  $L_r(f)$ . Let’s assume there is an algorithm that takes a description of a monomial of  $f$  and outputs the coefficient on it in time  $t$ . Then one can use the same algorithm to compute coefficients of  $L_r(f)$ , as the description of a monomial and its lifted version is *exactly* the same.

Our main technical lemma, proved in Section 4, efficiently converts a circuit for the lifted polynomial  $L_3(f)$  into a circuit for  $f$ . Setting  $r = 3$  is easiest to present, and gives the best qualitative bounds that we know how to achieve with this technique. So we continue with this choice of  $r$  below.

► **Lemma 4** (Circuit Decoding). *If there exists an arithmetic circuit of size  $s$  computing  $L_3(f)$ , then there exists a circuit of size  $O(n^{\omega/3}s)$  computing  $f$ .*

The lifting operation can be iterated. Take a polynomial  $L_r(f) \in \mathbb{F}\langle Y \rangle$  and re-number the  $Y$  variables lexicographically to obtain new singly-indexed  $X$  variables, and lift the resulting polynomial again. The result of repeating this process  $k$  times on a polynomial  $f$  is denoted  $L_r^k(f)$ . Using the circuit-decoding Lemma 4 we have the following lower-bound amplification for iterated lifting.

► **Lemma 5** (Iterated Lifting Amplifies Hardness). *Let  $k \leq \gamma \log(n)$  be a positive integer, where  $\gamma > 0$  is a sufficiently small positive constant. Suppose  $f$  is a polynomial on  $N = 3^{3/2}(3^{1/2}n)^{3^k}$  variables of degree  $d$ . Then  $L_3^k(f)$  is a polynomial on  $9n$  variables of degree  $3^k d$  and the following holds:*

$$AC(L_3^k(f)) \geq \frac{AC(f)}{N^{\omega/2}}$$

If we have a small circuit for  $L_3^k(f)$  then by applying Lemma 4 iteratively  $k$  times we will end up with a small circuit for  $f$ . We require  $N$  to be in a particular form to avoid dealing with rounding. The calculations appear below.

**Proof of Lemma 5.** Let  $N_k$  denote the number of variables of  $L_3^k(f)$ , where one can verify that  $N_0 = N$  and  $N_{i+1} = 3N_i^{1/3}$ . Our choice for  $N$  guarantees that  $N_i$  is an integer for all  $i = 0, \dots, k$ . Using Lemma 4 we get

$$AC(L_3^{i+1}(f)) \geq \alpha \frac{AC(L^i)}{N_i^{\omega/3}},$$

where  $\alpha > 0$  is some absolute constant. Folding the recursion gives

$$AC(L_3^k(f)) \geq \alpha^k AC(f) \prod_{i=0}^{k-1} N_i^{-\omega/3}$$

We will need to use an explicit expression for  $N_i = 3^{3/2}(3^{-3/2}N)^{\frac{1}{3^i}}$ .

$$\begin{aligned} AC(L_3^k(f)) &\geq \alpha^k AC(f) \left( \prod_{i=0}^{k-1} 3^{3/2}(3^{-3/2}N)^{\frac{1}{3^i}} \right)^{-\omega/3} \\ &= \alpha^k AC(f) \left( 3^{\frac{3}{4}(2k-3+3^{1-k})} N^{\frac{3}{2}(1-3^{-k})} \right)^{-\omega/3} \end{aligned}$$

So:

$$AC(L_3^k(f)) \geq \frac{AC(f)}{N^{\omega/2}} \left( \frac{\alpha^{\frac{3k}{\omega}} N^{\frac{3k-1}{2}}}{3^{\frac{3}{4}(-3+3^{1-k}+2k)}} \right)^{\omega/3}$$

If we recall that  $k \leq \gamma \log(n)$  and choose  $\gamma$  small enough we can ensure that:

$$\alpha^{-\frac{3k}{\omega}} 3^{\frac{3}{4}(-3+3^{1-k}+2k)} < N^{\frac{3}{2}3^{-k}}$$

As left hand side is  $2^{\theta(k)}$  and right hand side is  $n^{\theta(1)}$ . This immediately implies:

$$AC(L_3^k(f)) \geq \frac{AC(f)}{N^{\omega/2}} \quad \blacktriangleleft$$

### 3.1 Amplifying Lower Bounds via Lifting

Theorem 1 (amplification to any fixed polynomial hardness) is straightforward, by taking  $k$  to be some large constant in Lemma 5 above:

**Proof of Theorem 1.** Let  $P = \{P_n\}$  be a family of explicit polynomials, where  $P_n$  is a polynomial on  $n$  variables, such that  $\exists \alpha, \epsilon > 0$  such that  $\forall n : P_n$  is not computable by arithmetic circuits of size  $\alpha n^{\frac{\omega}{2} + \epsilon}$ . We will define family of polynomials  $Q = \{Q_n\}$  to be

lifted version of  $P$ , where again  $Q_n$  is a polynomial on  $n$  variables. Formally,  $Q_{9n} = L_3^k(P_N)$ , where  $N = N(n) = 3^{3/2}(3^{1/2}n)^{3^k}$ . It is easy to verify that  $N$  is always an integer. For general  $n$  define  $Q_n = Q_{9\lfloor n/9 \rfloor}$  by adding dummy variables. By Lemma 5:

$$AC(Q_n) \geq \frac{AC(P_N)}{N^{\frac{\epsilon}{2}}} \geq \alpha N^\epsilon \geq \alpha 3^{3/2\epsilon} (3^{1/2}n)^{\epsilon 3^k} = n^{\Omega(3^k)}$$

For any  $c > 0$  we can take  $k$  to be sufficiently large constant and have  $AC(Q_n) > n^c$ . Furthermore, note that  $\deg(Q_{9n}) = 3^k \deg(P_N)$ . So if  $\deg(P_N) = O(N^a)$  is polynomial in  $N$ , then  $\deg(Q_n) = O(3^k n^{a3^k})$ . In particular, for any fixed  $k$ ,  $\deg(Q_n) = \text{poly}(n)$  as claimed. Also  $Q$  is explicit as it is a lifted version of  $P$ . ◀

**Proof of Theorem 2.** The proof is identical to the proof of Theorem 1, except that we take  $k = \gamma \log n$ . Note that as we assume here that  $P_n$  all have a constant degree, then  $Q_n$  will have degree  $\text{poly}(n)$  as claimed. As  $P$  is an explicit polynomial,  $Q$  is also explicit polynomial. ◀

## 4 Structuring Circuits

In this section we obtain a normal form for non-commutative circuits computing certain restricted types of polynomials. The idea is similar to homogenization: we classify monomials into “types” and efficiently re-write the circuit in terms of operations on those types. The proofs share a common structure: we define an operator that splits polynomials into well-typed monomials. We then pass this operator through the circuit  $C$  layer-by-layer, starting from the output gate. Each time we advance the operator-layer through  $C$ , we maintain:

- (i) The polynomial computed by  $C$  does not change;
- (ii) All gates above the operator-layer compute restricted polynomials;
- (iii) Not too much additional hardware is introduced;
- (iv) At leaf nodes, operators can be eliminated from  $C$ .

This process is like a glacial movement during the ice age. An operator slides over the circuit and then disappears, drastically changing the landscape behind it.

### 4.1 Monomial & Circuit Types

For non-commutative polynomials, monomials are just words over the variables. So all of our monomial types will be constraints on the ordering of variables, referring to the “place” part of a  $Y$  variable.

► **Definition 6** (Structured Monomials in  $Y$ ). For fixed  $r \in \mathbb{N}$ , we define the following subsets of all monomials over double-indexed variables  $Y$ .

**r-pinned**,  $\widetilde{\mathcal{M}}_{i,j}^r$ : monomials  $m$  that start with  $y \in Y_i$ , end with  $y' \in Y_j$ , and obey  $\mathbb{Z}/r$  ordering. That is, after each  $y \in Y_k$  appearing in  $m$  the next variable is always some  $y' \in Y_{(k+1) \bmod r}$ .

**r-aligned**,  $\widetilde{\mathcal{M}}^r$ : any  $m \in \widetilde{\mathcal{M}}_{0,(r-1)}^r$ .

We do not bound the lengths of pinned or aligned monomials. The counter  $k$  indexing sets of variables  $Y_k$  may circle around  $\mathbb{Z}/r$  many times in going from  $i$  to  $j$ . We classify circuits and polynomials in the obvious way based on these sets of monomials.

► **Definition 7** (Structured Polynomials in  $Y$ ). A polynomial  $p \in \mathbb{F}\langle Y \rangle$  is **r-pinned** if  $\exists i, j$  such that every monomial of  $p$  is in  $\widetilde{\mathcal{M}}_{i,j}^r$ , or **r-aligned** if every monomial of  $p$  is in  $\widetilde{\mathcal{M}}^r$ .

When  $r$  is clear from the context, we shorthand  $\widetilde{\mathcal{M}}_{i,j} = \widetilde{\mathcal{M}}_{i,j}^r$  and  $\widetilde{\mathcal{M}} = \widetilde{\mathcal{M}}^r$ .

► **Definition 8** (Structured Circuits in  $Y$ ). A circuit  $C$  is **r-pinned** if every gate of  $C$  computes an  $r$ -pinned polynomial. Note that each gate could have different start and end indices  $i, j$ .  $C$  is **r-aligned** if every gate of  $C$  computes an  $r$ -aligned polynomial. An  $r$ -aligned circuit has  $r$ -aligned monomials as inputs, not single variables.

Recall that our goal in this section is to build a circuit for  $f$  from a circuit for  $L_r(f)$ . If we have an  $r$ -aligned circuit of size  $s$  for  $L_r(f)$ , this is straightforward. The bottom layer of an  $r$ -aligned circuit is a set of monomials, not variables. Since these monomials are  $r$ -aligned, each one uniquely represents a sequence of natural numbers in base  $n^{1/r}$ . Simply replace each encoded number  $i$  with  $x_i$  and take their product. After this substitution, we have a circuit that computes  $f$  of size  $O(s)$ .

If some *general* circuit  $C$  computes an *aligned* polynomial  $f$ , we can obtain an *aligned* circuit  $C'$  for  $f$  of only slightly larger size. This construction proceeds in two stages: from general circuits to pinned circuits (lemma 9), then from pinned circuits to aligned circuits (lemmas 10 and 11).

The circuit decoding for lifted polynomials of Lemma 4 is then immediate, because  $L(f)$  is always an aligned polynomial. We give two constructions: the first is elementary but inefficient, the second uses fast matrix multiplication to optimize storage of “type information” about polynomials. The first stage, from general to pinned circuits, is common to both proofs.

## 4.2 Operators on Polynomials

To efficiently store polynomials, we will sometimes need to “trim off” extraneous variables from the ends of each monomial. So we give two new operators on polynomials,  $\div_{\mathcal{L}}$  and  $\div_{\mathcal{R}}$ , that “divide what they can and discard the remainder.” These operators act on the left and right of  $f$ , respectively. Formally,  $\div_{\mathcal{L}}$  and  $\div_{\mathcal{R}}$  are defined in terms of two possible decompositions of a polynomial  $f$ :

- **Right division:** Let  $f = Q \times \sigma + R$  where  $Q \times \sigma$  sums over monomials of  $f$  ending with  $\sigma$ . Define:  $f \div_{\mathcal{R}} \sigma = Q$ .
- **Left division:** Let  $f = \tau \times Q' + R$  where  $\tau \times Q'$  sums over monomials of  $f$  starting with  $\tau$ . Define:  $\tau \div_{\mathcal{L}} f = Q'$ .

Because our polynomials are non-commutative, these decompositions are unique. Notice that in left-division  $\div_{\mathcal{L}}$ , the monomial  $\tau$  is *not* the object being operated on; it appears on the left to denote which side of the monomials of  $f$  is altered by the operation. Immediately, we have:

$$p \times q = \sum_{a \in Y} (p \div_{\mathcal{R}} a) \times (a \times q) = \sum_{a \in Y} (p \times a) \times (a \div_{\mathcal{L}} q)$$

Finally, we denote by  $\mathcal{M}$  the set of all possible monomials, and by  $\text{coeff}(f, m)$  the coefficient of  $f$  on monomial  $m$ . When expanding polynomials as sums over monomials, we write the monomial  $m$  as  $x^m$  or  $y^m$ , like so:

$$f(Y) = \sum_{m \in \mathcal{M}} \text{coeff}(f, m) \times y^m$$

## 4.3 Aligning Circuits

We begin the alignment process by taking a general circuit for a pinned polynomial, and constructing a pinned circuit. This is similar to homogenization using the more complex set of monomial types introduced above.

► **Lemma 9** (General to Pinned Circuits). *Let  $C$  be a **general** arithmetic circuit of size  $s$  computing an  **$r$ -pinned** polynomial  $f(Y)$ . Then there exists an  **$r$ -pinned** arithmetic circuit  $C'$  of size  $r^3s$  computing  $f$ .*

**Proof of Lemma 9.** Define  $\Delta_{i,j}$  to transform  $f(Y)$  into a  $r$ -pinned polynomial, by discarding any coefficients on monomials outside  $\widetilde{\mathcal{M}}_{i,j}$ :

$$\Delta_{i,j}(p) = \sum_{m \in \widetilde{\mathcal{M}}_{i,j}} \text{coeff}(p, m) \times y^m$$

Let  $g_o$  be the output gate of  $C$ . By assumption,  $g_o$  computes an  $r$ -pinned polynomial. From the definition,  $\exists i, j$  such that  $\Delta_{i,j}(g_o) = g_o$ . This is our base case. Inductively, let  $g \in C$  be such that  $\exists i, j$  so  $\Delta_{i,j}(g) = g$ . We reason by cases on the type of  $g$ .

If  $g = u + v$ :

$$\begin{aligned} \Delta_{i,j}(u + v) &= \Delta_{i,j} \left( \sum_{m \in \mathcal{M}} (\text{coeff}(m, u) + \text{coeff}(m, v)) y^m \right) && \text{expand } u + v \\ &= \sum_{m \in \widetilde{\mathcal{M}}_{i,j}} (\text{coeff}(m, u) + \text{coeff}(m, v)) y^m && \text{definition of } \Delta \\ &= \sum_{m \in \widetilde{\mathcal{M}}_{i,j}} \text{coeff}(m, u) y^m + \sum_{m \in \widetilde{\mathcal{M}}_{i,j}} \text{coeff}(m, v) y^m && \text{split the sum} \\ &= \Delta_{i,j}(u) + \Delta_{i,j}(v) && \text{definition of } \Delta \end{aligned}$$

If  $g = u \times v$ :

$$\begin{aligned} \Delta_{i,j}(u \times v) &= \Delta_{i,j} \left( \sum_{m_\ell \in \mathcal{M}} \text{coeff}(m_\ell, u) y^{m_\ell} \times \sum_{m_r \in \mathcal{M}} \text{coeff}(m_r, v) y^{m_r} \right) && \text{unroll} \\ &= \Delta_{i,j} \left( \sum_{\substack{m_\ell \in \mathcal{M} \\ m_r \in \mathcal{M}}} \text{coeff}(m_\ell, u) y^{m_\ell} \text{coeff}(m_r, v) y^{m_r} \right) && \text{distribute} \\ &= \Delta_{i,j} \left( \sum_{\substack{m_\ell \in \mathcal{M} \\ m_r \in \mathcal{M}}} \text{coeff}(m_\ell, u) \text{coeff}(m_r, v) y^{m_\ell m_r} \right) && \text{commute in } \mathbb{F} \\ &= \sum_{\substack{m_\ell, m_r \in \mathcal{M} \\ \text{st. } m_\ell m_r \in \widetilde{\mathcal{M}}_{i,j}}} \text{coeff}(m_\ell, u) \text{coeff}(m_r, v) y^{m_\ell m_r} && \text{definition of } \Delta \end{aligned}$$

Because  $m_\ell m_r$  is pinned, we know (1) that  $m_\ell$  begins with some  $y \in Y_i$  and  $m_r$  ends with some  $y' \in Y_j$  and (2) that the transition from  $m_\ell$  to  $m_r$  must respect ordering in  $\mathbb{Z}/r$ . Formally, we know that  $\exists t$  such that  $m_\ell \in Y_i \dots Y_t$  and  $m_r \in Y_{(t+1) \bmod r} \dots Y_j$ . So let's split the above summation on this index, which is bounded by  $r$  because we assumed the polynomial is  $r$ -pinned. To ease legibility below, all indexing arithmetic for monomial sets  $\widetilde{\mathcal{M}}$  and for the operator  $\Delta$  is *implicitly* carried out in  $\mathbb{Z}/r$ .



## 12:10 Hardness Amplification for Non-Commutative Arithmetic Circuits

$$\begin{aligned}
g &= \sum_{t \in \mathbb{Z}/r} \sum_{\substack{m_\ell \in \widetilde{\mathcal{M}}_{i,t} \\ m_r \in \widetilde{\mathcal{M}}_{t+1,j}}} \text{coeff}(m_\ell, u) \text{coeff}(m_r, v) y^{m_\ell m_r} \\
&= \sum_{t \in \mathbb{Z}/r} \left( \sum_{m_\ell \in \widetilde{\mathcal{M}}_{i,t}} \text{coeff}(m_\ell, u) y^{m_\ell} \right) \left( \sum_{m_r \in \widetilde{\mathcal{M}}_{(t+1),j}} \text{coeff}(m_r, v) y^{m_r} \right) && \text{distribute} \\
&= \sum_{t \in \mathbb{Z}/r} \Delta_{i,t}(u) \times \Delta_{t+1,j}(v) && \text{definition of } \Delta
\end{aligned}$$

The circuit  $C'$  contains, for every gate  $g$  in  $C$ , the  $r^2$  gates computing  $\Delta_{i,j}(g)$  for all  $i, j \in \mathbb{Z}/r$ . Addition gates do not require additional gates; multiplication gates require an addition a factor of  $r$  more gates to compute. So in total if  $C$  has  $s$  gates then  $C'$  has at most  $r^3 s$  gates.  $\blacktriangleleft$

The pinning lemma proved above enforces an ordering on variables that respects  $\mathbb{Z}/r$ . But for circuit decoding, monomials that are *aligned* and thus represent complete numbers are required.

We partition pinned monomials into a prefix, body and suffix. The body of a monomial is the substring between the first variable from  $Y_0$  and the last variable from  $Y_{(r-1)}$  (it can be empty). By definition, the length of the body is a multiple of  $r$ . This means that the body uniquely represents a string of natural numbers, which can easily be mapped back to  $x$ -variables.

Then the prefix of a monomial is everything to the left of the body, and the suffix is everything to the right of the body. We also need to consider monomials of small length, for which the body is undefined. These parts of a monomial do not *yet* represent even a single natural number. But, because the circuit computes an aligned polynomial, we know that these monomials will eventually become part of the body via subsequent multiplication operations.

### 4.3.1 Simple Circuit Alignment

The construction below anticipates and brute-forces these possible “completions” of non-body monomials at each gate of the circuit.

► **Lemma 10** (Pinned to Aligned Circuits, Simply). *Let  $C$  be an  $r$ -pinned arithmetic circuit of size  $s$  computing an  $r$ -aligned polynomial  $f(Y)$ . Then there exists a  $r$ -aligned arithmetic circuit  $C'$  of size  $O(sn^{3r-2})$  computing  $f$ . If  $C$  was a monotone circuit, then  $C'$  is also monotone.*

**Proof.** First, we define the undesirable sets of monomials. These monomials are all possible obstructions to alignment that must be computed in terms of aligned polynomials.

Incomplete :  $I = \{\rho \mid \rho \in \mathcal{M} \text{ of length } < r\}$

Prefix :  $P = \left( \cup_{i=1}^{r-1} \widetilde{\mathcal{M}}_{i,(r-1)} \right) \cap I$

Suffix :  $S = \left( \cup_{i=0}^{r-2} \widetilde{\mathcal{M}}_{0,i} \right) \cap I$



We use these monomial sets to separate the body of a monomial from the prefix and suffix, which are not perfectly aligned:

$$\mathcal{W}_{\sigma,\tau} = \{(w, m) \mid w = \sigma m \tau \text{ where } m \in \widetilde{\mathcal{M}}, \sigma \in \text{Prefix}, \text{ and } \tau \in \text{Suffix}\}$$

We want all the polynomials computed by  $C'$  to be aligned, so we can only have monomials with empty prefix, suffix, and incomplete monomial sets at each gate. But we need the coefficients associated with these “flawed” polynomials to compute with. This suggests an operator  $\Gamma$  that will take only parts of the polynomial with a particular suffix and prefix, multiplying the coefficient on  $\sigma m \tau$  by the monomial  $m$  only, where  $m$  is a body monomial. We will also need to recover coefficients on incomplete monomials, so we let a unary  $\Gamma$  extract specific coefficients:  $\Gamma_\rho(f) = \text{coeff}(\rho, f)$ . We could also use the “division” operators above to express  $\Gamma$ :

$$\begin{aligned} \Gamma_{\sigma,\tau}(f) &= \sum_{(w,m) \in \mathcal{W}_{\sigma,\tau}} \text{coeff}(w, f) \times y^m \\ &= \sigma \div_{\mathcal{L}} f \div_{\mathcal{R}} \tau \end{aligned}$$

If a polynomial  $f$  is aligned, then  $\Gamma_{1,1}(f) = f$  and all other operators are 0. That means that if  $g_o$  is the output gate of the original circuit  $C$ , then  $\Gamma_{1,1}(g_o) = g_o$ . Inductively, let  $C$  be the pinned circuit computing an aligned polynomial  $f$  and suppose  $g \in C$ . We are going to show how to push  $\Gamma$  operators one level deeper into the circuit, reasoning by cases on the form of  $f$ .

Suppose  $g = u + v$ . Addition does not change the collection of monomials except by cancellation, so we have the following easy identities, which follow from the same kind of monomial partitioning used to prove the pinning Lemma 9 above:

$$\begin{aligned} \forall a, b : \Gamma_{a,b}(g) &= \Gamma_{a,b}(u) + \Gamma_{a,b}(v) \\ \forall c : \Gamma_c(g) &= \Gamma_c(u) + \Gamma_c(v) \end{aligned}$$

Now suppose  $g = u \times v$ . First consider how some incomplete monomial  $c$  could have a nonzero coefficient in  $g$ ; it would have to be the case that two incomplete monomials of  $u$  and  $v$  were multiplied together to form  $c$ . Therefore:

$$\forall c, \Gamma_c(g) = \sum_{\{d,e \in I \mid de=c\}} \Gamma_d(u) \Gamma_e(v).$$

Similarly, we reason by cases on how the monomials of  $\Gamma_{a,b}(g)$  could have been formed by multiplying the monomials of  $u$  and  $v$ :

$$\begin{aligned} \Gamma_{a,b}(g) &= \sum_{\{c \in S, d \in P : |cd|=r\}} \Gamma_{a,c}(u) y^{cd} \Gamma_{d,b}(v) \quad // \text{suffix}(u) \times \text{prefix}(v) \text{ becomes aligned} \\ &+ \sum_{\{c \in S, d \in I : cd=b\}} \Gamma_{a,c}(u) \Gamma_d(v) \quad // \text{suffix}(u) \times \text{incomplete}(v) \text{ becomes } b \\ &+ \sum_{\{c \in I, d \in P : cd=a\}} \Gamma_c(u) \Gamma_{d,b}(v) \quad // \text{incomplete}(u) \times \text{prefix}(v) \text{ becomes } a \end{aligned}$$

The above formula completely enumerates how the polynomials  $u$  and  $v$  could multiply to produce coefficients on monomials with prefix and suffix  $a, b$  in  $g$ , in terms of  $\Gamma$  applied to  $u$  and  $v$ . Thus we have successfully expressed  $\Gamma_{a,b}(g)$  in terms of earlier gates.

Using the formulas above we can push the  $\Gamma$ -operators down one level. Clearly, all gates above the operator level compute aligned polynomials: we are keeping track of undesirable monomials in the labels on gates. Finally, observe that all the operators applied to a single variable or constant are constants. This means we can replace every operator applied to the input of the circuit by a constant. So after pushing the operators down to the leaves we get an aligned circuit that computes  $\Gamma_{1,1}(g_o) = f$ .

All that remains is to estimate the size of the resulting circuit. Each addition gate of the old circuit was substituted with a circuit of size  $n^{2r-2}$ . Each multiplication gate of the old circuit was substituted with a circuit of size  $n^{3r-2}$ . So the size of our aligned circuit computing  $f$  is at most  $O(sn^{3r-2})$ . ◀

### 4.3.2 Efficient Circuit Alignment

We can get smaller aligned circuits using a more sophisticated technique. Notice that the construction above enumerates all possible “completions” of non-aligned polynomials to aligned polynomials at each level. We assigned a gate to each such completion, which fails to exploit the fact that it is not possible to obtain non-aligned polynomials by arithmetic operations on aligned polynomials. The construction below *does* take advantage of these restrictions to do much less brute-force enumeration of intermediate non-aligned polynomials, by *implicitly* representing future completions at each gate. We use matrix multiplication to organize this more efficient combination of polynomial types, which is why  $\omega$  appears in the complexity of the resulting circuit.

We restrict our attention from now on to  $r = 3$ . It will simplify the proof and it turns out that it gives almost optimal results.

► **Lemma 11** (Pinned to Aligned Circuits, Efficiently). *If there exists a **3-pinned** arithmetic circuit  $C$  of size  $s$  computing a **3-aligned** polynomial  $f(Y)$ , then there exists a **3-aligned** circuit of size  $O(sn^\omega)$  computing  $f(Y)$ .*

The high level idea of the proof of Lemma 11 is as follows. Let  $f_M$  denote a matrix of size  $n \times n$  that has  $f$  as its  $[1, 1]$  entry and 0 elsewhere. One can measure the arithmetic circuit complexity of  $f_M$  in a model where matrices are on the wires of the circuit instead of scalars. We use this observation to prove the above lemma in two steps:

1. Convert the circuit for  $f$  into a circuit for  $f_M$  over the ring of matrices. (Lemma 13)
2. Convert the circuit for  $f_M$  back into a circuit for  $f$  by replacing each gate with circuits for matrix addition and matrix multiplication. The resulting circuit is aligned, and has hardware proportional to the original number of gates times the cost of matrix multiplication.

The key step is converting a circuit for  $f$  into a circuit for  $f_M$ . As before, we introduce a mapping  $\Phi$  to transform the original circuit layer-by-layer. This time, however, it is not an operator on polynomials: it maps polynomials to matrices. By propagating this  $\Phi$  through  $C$ , we obtain a circuit for  $f_M$ . Lemma 12 below states the properties of  $\Phi$ . We give the full proof of correctness for our efficient construction of aligned circuits (Lemma 11) at the end of this section, because it is straightforward once we have  $\Phi$ .

► **Lemma 12** (Polynomial to Matrix). *There exists a map  $\Phi$  that takes a polynomial on  $3n$  variables to an  $n \times n$  matrix with polynomial entries satisfying the following conditions:*

- (i) *For all 3-pinned polynomials  $g$  all entries of  $\Phi(g)$  are aligned polynomials.*
- (ii) *If  $g$  is a 3-aligned polynomial, then  $\Phi(g)[1, 1] = g$  and all other entries of  $\Phi(g)$  are zero.*
- (iii) *If  $g$  is a variable or a constant, then the degree of each entry of  $\Phi(g)$  is at most 3.*

(iv) For all 3-pinned polynomials  $g, u, v$ , and arithmetic  $+, \times$  over the ring of matrices:

$$\begin{aligned} g = u + v &\Rightarrow \Phi(g) = \Phi(u) + \Phi(v) \\ g = u \times v &\Rightarrow \Phi(g) = \Phi(u) \times \Phi(v) \end{aligned}$$

One new trick that we are going to use is that we will sometimes not store the suffix or prefix of the monomial if it is too long. Instead we will store what it can become after we complete it to an aligned monomial. For example, consider the following polynomial:  $y_{0,i}y_{2,j} + y_{1,i'}y_{2,j'}$ . Instead of memorizing it this way, one can remember that it will become  $y_{0,i}y_{2,j}y_{3,k} + y_{1,i'}y_{2,j'}y_{3,k}$  after we multiply it by  $y_{3,k}$ . By contrast, the simple alignment procedure stores these completions on *both* sides of the multiplication, duplicating information and wasting gates.

**Proof of Lemma 12.** We need only define the operator  $\Phi$  for 3-pinned polynomials. Every 3-pinned polynomial  $g(Y)$  is one of 9 types  $(a, b) \in \{0, 1, 2\} \times \{0, 1, 2\}$ , based on which  $Y$ -variables start and end all the monomials of  $f$ . Denote by  $\mathbb{F}_{a,b}(Y)$  the set of 3-pinned polynomials of type  $(a, b)$ . Each entry  $[i, j]$  of the matrix  $\Phi(g)$  will be an arithmetic expression in terms of  $g$  that depends on the “pinning type” of  $g$  and the indices  $[i, j]$ . Below, we define functions  $\lambda$  and  $\rho$  which select how to transform  $g$  from the left and the right, respectively, in terms of pinning type of  $g$  and index of  $\Phi(g)$ . We use below the notation  $\delta(i) = 1$  if  $i = 1$  and  $\delta(i) = 0$  otherwise.

For  $g \in \mathbb{F}_{a,b}(Y)$  define  $\Phi(g)[i, j] = \lambda(a, i) g \rho(b, j)$  where:

$$\lambda(a, i) = \begin{cases} \delta(i) \times & \text{if } a = 0, \\ y_{0,i} \times & \text{if } a = 1, \\ y_{2,i} \div_{\mathcal{L}} & \text{if } a = 2 \end{cases} \quad \text{and} \quad \rho(b, j) = \begin{cases} \times \delta(j) & \text{if } b = 2, \\ \times y_{2,j} & \text{if } b = 1, \\ \div_{\mathcal{R}} y_{0,j} & \text{if } b = 0 \end{cases}$$

We expand the definition of  $\Phi$  concretely below. This matrix is the outer product of the  $\lambda$  and  $\rho$  operation selection functions “around”  $g$ .

$$\Phi(g)[i, j] \leftarrow \text{entry } (a, b) \text{ of } \begin{bmatrix} \delta(i) \times g \div_{\mathcal{R}} y_{0,j} & \delta(i) \times g \times y_{2,j} & \delta(i) \times g \times \delta(j) \\ y_{0,i} \times g \div_{\mathcal{R}} y_{0,j} & y_{0,i} \times g \times y_{2,j} & y_{0,i} \times g \times \delta(j) \\ y_{2,i} \div_{\mathcal{L}} g \div_{\mathcal{R}} y_{0,j} & y_{2,i} \div_{\mathcal{L}} g \times y_{2,j} & y_{2,i} \div_{\mathcal{L}} g \times \delta(j) \end{bmatrix}$$

Inspecting the expansion above, properties (i), (ii), and (iii) claimed for  $\Phi$  are clear. It remains to show property (iv): that  $\Phi$  maps arithmetic over 3-pinned polynomials to arithmetic over the ring of matrices. Reasoning from the definitions of  $\Phi$  and the division operators we have the following:

$$\forall c, d \in \{0, 1, 2\} \text{ such that } d = (c + 1) \bmod 3 : p \times q = \sum_{i \in [n]} p \rho(c, i) \times \lambda(d, i) q$$

If  $g, u, v$  are 3-pinned polynomials then, by additivity of the matrix ring,  $g = u + v \Rightarrow \Phi(g) = \Phi(u) + \Phi(v)$ . We also need  $g = u \times v \Rightarrow \Phi(g) = \Phi(u) \times \Phi(v)$  which we prove directly. Let  $a, b, c \in \mathbb{Z}/3$  be such that  $u \in \mathbb{F}_{a,b}(Y)$  and  $v \in \mathbb{F}_{b+1,c}(Y)$ . These numbers must exist, since  $u$

## 12:14 Hardness Amplification for Non-Commutative Arithmetic Circuits

and  $v$  multiply to give the 3-pinned polynomial  $g \in \mathbb{F}_{a,c}\langle Y \rangle$ . So

$$\begin{aligned} (\Phi(u) \times \Phi(v))[i, j] &= \sum_k \Phi(u)[i, k] \Phi(v)[k, j] \\ &= \lambda(a, i) \left( \sum_k u \rho(b, k) \lambda(b+1, k) v \right) \rho(c, j) \\ &= \lambda(a, i) u \times v \rho(c, j) \\ &= \lambda(a, i) g \rho(c, j) = \Phi(g)[i, j] \end{aligned}$$

The key observation for the derivation above is that  $\rho(b, k)\lambda(b+1, k)$  “cancels out” for any  $b \in \mathbb{Z}/3$ . This is what saves hardware compared to the simple construction: there is no “garbage” in the middle of the representation to enumerate over. ◀

We can now push  $\Phi$  “down” through a pinned circuit to obtain an aligned circuit. We first need the following lemma, to convert a circuit for  $f$  into a circuit for  $f_M$  over the ring of matrices.

► **Lemma 13** (Pinned Circuit to Matrix Circuit). *If there exists a 3-pinned arithmetic circuit of size  $s$  that computes a 3-pinned polynomial  $f$ , then there exists a circuit of size  $O(s)$  that computes  $f_M$ . This circuit uses matrix addition and multiplication as gates, and has matrices with aligned monomials of degree at most 3 in entries as inputs.*

**Proof.** Suppose that we are given a 3-pinned circuit  $C$  for 3-aligned polynomial  $f$ . Then, as  $\Phi(f) = f_M$  we can apply the operator  $\Phi$  to the output of  $C$  and get a circuit for  $f_M$ . Recall the properties of  $\Phi$  guaranteed in Lemma 12. We use property (iv) to push  $\Phi$  down one level of  $C$ . We will measure the size of this circuit as the number of gates that perform arithmetic operations, both over polynomials and matrices, which is the same as counting all except  $\Phi$ -gates. It is easy to see that when we apply rule (iv) we are not increasing size of the circuit measured this way.

Eventually we will sink all the  $\Phi$ -gates to the very bottom. We will have a circuit with only matrix addition, matrix multiplication and  $\Phi$  gates, and the last are only applied to the inputs. By property (iii) we know that  $\Phi$  applied to the input computes a matrix whose entries are aligned polynomials of degree at most 3. That means that we can just claim the outputs of  $\Phi$  as our new inputs – we are allowed to have matrices with degree 3 aligned polynomials as inputs in the model of matrix circuits. This removes all the  $\Phi$  from  $C$ , and the only types of gates left are matrix multiplication and addition. Then our measure of size is now the same as the number of gates, so we have a new matrix circuit with size exactly matching that of  $C$ . ◀

Note that it is impossible to obtain non-aligned polynomials by arithmetic operations on aligned polynomials. Therefore, all matrices computed by the gates in such a circuit would have aligned polynomials in all entries. We conclude by mapping pinned circuits to aligned circuits, efficiently.

**Proof of Lemma 11.** Take a circuit for  $f$ , and construct a circuit for  $f_M$ , using Lemma 13. Replace each matrix with  $n^2$  gates each representing one entry. Replace each matrix addition and multiplication gate with a circuit on  $2n^2$  inputs that perform the same operations. This will leave us with a circuit of size  $O(sn^\omega)$  over aligned monomials of degree at most 3 as inputs. ◀

---

**References**

---

- 1 Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25–28, 2008, Philadelphia, PA, USA*, pages 67–75. IEEE Computer Society, 2008.
- 2 Vikraman Arvind, Pushkar S. Joglekar, Partha Mukhopadhyay, and S. Raja. Randomized polynomial time identity testing for noncommutative circuits. 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 831–841. ACM, 2017. doi:10.1145/3055399.3055442.
- 3 Vikraman Arvind, Pushkar S. Joglekar, and S. Raja. Noncommutative valiant’s classes: Structure and complete problems. *TOCT*, 9(1):3:1–3:29, 2016. doi:10.1145/2956230.
- 4 Vikraman Arvind and Srikanth Srinivasan. On the hardness of the noncommutative determinant. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 677–686. ACM, 2010.
- 5 Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983.
- 6 Klim Efremenko, Ankit Garg, Rafael Mendes de Oliveira, and Avi Wigderson. Barriers for rank methods in arithmetic complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:27, 2017.
- 7 Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving algebraic circuits lower bounds. 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 653–664. ACM, 2017.
- 8 Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:9, 2017.
- 9 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016.
- 10 Pavel Hrubes, Avi Wigderson, and Amir Yehudayoff. Relationless completeness and separations. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:40, 2010. URL: <http://eccc.hpi-web.de/report/2010/040>.
- 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.
- 12 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012.
- 13 François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303. ACM, 2014.
- 14 Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 410–418. ACM, 1991.
- 15 Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.
- 16 Amir Shpilka, Amir Yehudayoff, et al. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010.
- 17 Leslie G. Valiant. Completeness classes in algebra. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings*

of the 11th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

**A** Infinitely often vs almost everywhere hardness

In this section we carry out our hardness amplification for polynomials that are sometimes hard, as opposed to hard everywhere. In the proof of 1 we used the assumption that:

$$\exists \alpha, \epsilon > 0 \text{ such that } \forall n : P_n \text{ is not computable by arithmetic circuits of size } \alpha n^{c+\epsilon}$$

A more natural way to say that some polynomial  $P$  requires circuits of size larger than  $n^c$  would be:

$$P \notin \text{ASize}[n^c],$$

where  $\text{ASize}[f(n)]$  is set of all sequences of polynomials that can be computed by circuits of size  $O(f(n))$ . The difference between these two definitions is that the first means that the polynomial is not computable by small circuits everywhere, and the second means that the polynomial is not computable by small circuits for infinitely many  $n$ . While in the proof of Theorem 1 we used the first definition, we actually only needed that the polynomial is hard on infinitely many points of the form  $3^{-3/2}(3^{\frac{1}{2}}n)^{3^k}$  for some  $n$  and fixed  $k$ . This motivates the notion of infinitely often hardness on a subset, described below:

► **Definition 14.**  $\text{ASize}[f(n)]$  is set of all sequences of polynomials that can be computed by circuits of size  $O(f(n))$

Now we will tweak this definition to describe hardness on subset of integers:

► **Definition 15.** Let  $S$  be a infinite size subset of natural numbers.  $\text{ASize}_S[f(n)]$  is set of all sequences of polynomials that can be computed by circuits of size  $O(f(n))$  for all  $n \in S$ .

► **Lemma 16.** Let  $A$  be a subset of even natural numbers, such that  $\frac{\log A_{n+1}}{\log A_n} = 1 + o(1)$ , where  $A_n$  is  $n$ -th smallest element of  $A$  is  $\leq 2^{n^\gamma}$  for some  $\gamma$  and  $P$  is an explicit sequence of polynomials that is i.o.  $n^c$  hard for some  $c$ . Then for every  $\epsilon > 0$  there is an explicit sequence of polynomials  $Q$ , such that  $Q$  is i.o.  $n^{c-\epsilon}$  hard on  $A$ .

**Proof.** We construct  $Q$  as by setting:

$$Q_{2n+1} = Q_{2n} = \sum_{k=1}^n x_{n+k} P_k(x_1, x_2, \dots, x_k)$$

It is easy to see that  $Q_{2n}(x_1, x_2, \dots, x_n, 0, 0, \dots, 0, 1, 0, \dots, 0) = P_k(x_1, \dots, x_k)$  if 1 is set in the  $n + k$ -th position. This means that:

$$\forall n : AC(Q_{2n+1}) = AC(Q_{2n}) \geq \max_{k \in [n]} AC(P_k)$$

Then suppose that  $AC(P_n) > \alpha n^c$  and let  $i$  be the smallest number, such that  $A_i$  is bigger than  $2n$ . Then  $AC(Q_{A_i}) > AC(P_n)$ . This implies that  $AC(Q_{A_i}) > \alpha n^c > \alpha A_{i-1}^c > \alpha A_i^{c \frac{\log A_{i-1}}{\log A_i}} > \alpha A_i^{c-o(1)}$ . This means that for any  $\epsilon > 0$  there would be infinitely many  $n \in A$ , such that  $AC(Q_n) > \alpha n^{c-\epsilon}$  ◀

Now we just need to observe that the set  $A = \{x | \exists n : x = 3^{-3/2}(3^{\frac{1}{2}}n)^{3^k}\}$  satisfies the property  $\frac{\log A_{n+1}}{\log A_n} = 1 + o(1)$ . It is true even if we allow  $k$  to be a monotone function of  $n$  if  $k = O(\log(n))$ , which covers all the range of parameters that we are currently using.

# Hardness vs Randomness for Bounded Depth Arithmetic Circuits

**Chi-Ning Chou**

School of Engineering and Applied Sciences, Harvard University,  
Cambridge, MA 02138, USA  
chiningchou@g.harvard.edu

**Mrinal Kumar**

Center for Mathematical Sciences and Applications, Harvard University  
Cambridge, MA 02138, USA  
mrinalkumar08@gmail.com

**Noam Solomon**

Center for Mathematical Sciences and Applications, Harvard University  
Cambridge, MA 02138, USA  
noam.solom@gmail.com

---

## Abstract

In this paper, we study the question of hardness-randomness tradeoffs for bounded depth arithmetic circuits. We show that if there is a family of explicit polynomials  $\{f_n\}$ , where  $f_n$  is of degree  $O(\log^2 n / \log^2 \log n)$  in  $n$  variables such that  $f_n$  cannot be computed by a depth  $\Delta$  arithmetic circuits of size  $\text{poly}(n)$ , then there is a deterministic sub-exponential time algorithm for polynomial identity testing of arithmetic circuits of depth  $\Delta - 5$ .

This is incomparable to a beautiful result of Dvir et al. [SICOMP, 2009], where they showed that super-polynomial lower bounds for depth  $\Delta$  circuits for any explicit family of polynomials (of potentially high degree) implies sub-exponential time deterministic PIT for depth  $\Delta - 5$  circuits of *bounded individual degree*. Thus, we remove the “bounded individual degree” condition in the work of Dvir et al. at the cost of strengthening the hardness assumption to hold for polynomials of *low degree*.

The key technical ingredient of our proof is the following property of roots of polynomials computable by a bounded depth arithmetic circuit : if  $f(x_1, x_2, \dots, x_n)$  and  $P(x_1, x_2, \dots, x_n, y)$  are polynomials of degree  $d$  and  $r$  respectively, such that  $P$  can be computed by a circuit of size  $s$  and depth  $\Delta$  and  $P(x_1, x_2, \dots, x_n, f) \equiv 0$ , then,  $f$  can be computed by a circuit of size  $\text{poly}(n, s, r, d^{O(\sqrt{d})})$  and depth  $\Delta + 3$ . In comparison, Dvir et al. showed that  $f$  can be computed by a circuit of depth  $\Delta + 3$  and size  $\text{poly}(n, s, r, d^t)$ , where  $t$  is the degree of  $P$  in  $y$ . Thus, the size upper bound in the work of Dvir et al. is non-trivial when  $t$  is small but  $d$  could be large, whereas our size upper bound is non-trivial when  $d$  is small, but  $t$  could be large.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Algebraic complexity theory

**Keywords and phrases** Algebraic Complexity, Polynomial Factorization Circuit Lower Bounds, Polynomial Identity Testing

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.13

**Related Version** <https://ecc.weizmann.ac.il/report/2018/052/>

**Acknowledgements** We are thankful to Rafael Oliveira and Guy Moshkovitz for helpful discussions.



© Chi-Ning Chou, Mrinal Kumar, and Noam Solomon;  
licensed under Creative Commons License CC-BY  
33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 13; pp. 13:1–13:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





## 1 Introduction

Arithmetic circuits are one of the most natural and fundamental models of algebraic computation. Formally, an arithmetic circuit  $\Psi$  over a field  $\mathbb{F}$  and variables  $\vec{x} = (x_1, x_2, \dots, x_n)$  is a directed acyclic graph, with the gates of in-degree zero (called leaves) being labeled by elements in  $\mathbb{F}$  and variables in  $\vec{x}$ , and the internal nodes being labeled by  $+$  (sum gates) or  $\times$  (product gates). The vertices of out-degree zero in  $\Psi$  are called output gates. The circuit  $\Psi$  computes a polynomial in  $\mathbb{F}[\vec{x}]$  in a natural way : the leaves compute the polynomial equal to its label. A sum gate computes the polynomial equal to the sum of the polynomials computed at its children, while a product gate computes the polynomial equal to the product of the polynomials computed at its children. Arithmetic circuits can be thought of as algebraic analog of Boolean circuits, and provide a succinct representation of multivariate polynomials, and are natural objects of study in Algebraic Complexity theory. Two of the most fundamental problems of interest in this area of research are the following.

- **Lower Bounds.** To show that there are *explicit* polynomial families which are hard, i.e. they cannot be computed by arithmetic circuits whose size is polynomial in the number of variables.
- **Polynomial Identity Testing (PIT).** To design an efficient deterministic algorithm which takes as input an arithmetic circuit  $C$ , and outputs if it is identically zero or not. It is easy to show by an appropriate counting argument that a random polynomial of degree  $d$  in  $n$  variables cannot be computed by an arithmetic circuit of size  $\text{poly}(n, d)$ , but no such *explicit*<sup>1</sup> polynomial families are known. Similarly, a randomized algorithm for the PIT question immediately follows from the classical Schwartz-Zippel lemma (see Lemma 15). The key challenge is to accomplish this task without using randomness.

The progress on these questions for general arithmetic circuits has been painfully slow. To date, there are no non-trivial<sup>2</sup> algorithms for PIT for general arithmetic circuits, while the best known lower bound, due to Bauer and Strassen [2], is a slightly superlinear lower bound  $\Omega(n \log n)$ , established over three decades ago. In fact, even for the class of bounded depth arithmetic circuits, no non-trivial deterministic PIT algorithms are known, and the best lower bounds known are just slightly superlinear [22].

In a very influential work, Kabanets and Impagliazzo [10] showed that the questions of derandomizing PIT and that of proving lower bounds for arithmetic circuits are equivalent in some sense. Their result adapts the Hardness vs Randomness framework of Nisan and Wigderson [18] to the algebraic setting. In their proof, Kabanets and Impagliazzo combine the use of Nisan-Wigderson generator with Kaltofen's result that all factors of a low degree (degree  $\text{poly}(n)$ ) polynomial with  $\text{poly}(n)$  sized circuit are computable by size  $\text{poly}(n)$  circuits [12]. They showed that given an explicit family of *hard* polynomials, one can obtain a *non-trivial*<sup>3</sup> deterministic algorithm for PIT.

The extremely slow progress on the lower bound and PIT questions for general circuits has led to a lot of attention on understanding these questions for more structured sub-classes of arithmetic circuits. Arithmetic formula [11], algebraic branching programs [15], multilinear circuits [21, 25, 24], and constant depth arithmetic circuits [19, 22, 9, 7, 17] are some examples of such circuit classes. A natural question is to ask if the equivalence of PIT and lower bounds

<sup>1</sup> See Definition 10 for a formal definition.

<sup>2</sup> Here, non-trivial means anything which is better than the brute force algorithm for general arithmetic circuits given by the Schwartz-Zippel lemma.

<sup>3</sup> Here, non-trivial means subexponential time, or quasipolynomial time, based on the hardness assumption.



also carries over to these more structured circuit classes. For example, does super-polynomial lower bounds for arithmetic formulas imply non-trivial deterministic algorithms for PIT for arithmetic formulas, and vice-versa?

The answers to these questions do not follow directly from the results in [10]; unlike general arithmetic circuits, none of these sub-classes are known to be *closed* under factoring, i.e., given a polynomial  $P$  which has a small formula (or bounded depth circuit), it is not known whether the factors of  $P$  also have small formulas (or bounded depth circuits). Recently, there has been some progress on these questions (see [20, 4]), but in general, these questions of being closed under factoring for arithmetic formulas and bounded depth circuits continue to remain open.

## 1.1 Bounded Depth Circuits

Dvir, Shpilka and Yehudayoff [5] initiated the study of this question of equivalence of PIT and lower bounds for bounded depth circuits. Dvir et al. observed that a part of the proof in [10] can be generalized to show that non-trivial PIT for bounded depth circuits implies lower bounds for such circuits. For the converse, the authors only showed a weaker statement; they proved that super-polynomial lower bounds for depth  $\Delta$  arithmetic circuit implies non-trivial PIT for depth  $\Delta - 5$  arithmetic circuits with *bounded individual degree*. The bounded individual degree condition is a bit unsatisfying, and so, the following question is of fundamental interest.

► **Question 1.** *Does a super-polynomial lower bound for depth  $\Delta$  arithmetic circuits imply non-trivial deterministic PIT for depth  $\Delta'$  arithmetic circuits<sup>4</sup>? In particular, can we get rid of the “bounded individual degree” condition from the results in [5]?*

In this paper, we partially answer Question 1 in the affirmative. Informally, we prove the following theorem.

► **Theorem 2 (Informal).** *A super-polynomial lower bound for depth  $\Delta$  arithmetic circuits for an explicit family of low degree polynomials implies non-trivial deterministic PIT for depth  $\Delta - 5$  arithmetic circuits.*

Here, by low degree polynomials, we mean polynomials in  $n$  variables and degree at most  $O(\log^2 n / \log^2 \log n)$ . Thus, by strengthening the hardness hypothesis in [5], we remove the bounded individual degree restriction from the implication. We now formally state our results and elaborate further how they compare with prior work.

## 1.2 Our Results

We start by stating our main theorem, which is a formal restatement of Theorem 2.

► **Theorem 3.** *Let  $\Delta \geq 6$  be a positive integer, and let  $\varepsilon > 0$  be any real number. Let  $\{f_m\}$  be a family of explicit polynomials such that  $f_m$  is an  $m$ -variate multilinear polynomial of degree  $d = O(\log^2 m / \log^2 \log m)$  which cannot be computed by an arithmetic circuit of depth  $\Delta$  and size  $\text{poly}(m)$ . Then, there is a deterministic algorithm, which, given as input a circuit  $C \in \mathbb{C}[\bar{x}]$  of size  $s$ , depth  $\Delta - 5$  and degree  $D$  on  $n$  variables, runs in time  $(snD)^{O(n^{2\varepsilon})}$  and determines if the polynomial computed by  $C$  is identically zero.*

<sup>4</sup> Here, we think of  $\Delta'$  as  $\Delta - O(1)$ .

Some remarks on the above theorem statement.

► **Remark.** Our algorithm works as long as the characteristic of the underlying field is sufficiently large or zero, but for simplicity, the presentation in this paper just focuses on the field  $\mathbb{Q}$  of rational numbers.

► **Remark.** The bound  $d \leq \log^2 m / \log^2 \log m$  can be relaxed to  $d \leq \log^k m / \log^k \log m$  for any positive integer  $k$ , but we would need lower bounds for depth  $\Delta + 2k + 2$  to be able to do PIT for depth  $\Delta$  circuits. We point this difference out in the proof of Theorem 5, but do not dwell further on this.

► **Remark.** The running time of the PIT algorithm gets better as the lower bound gets stronger. Also, the constraint on the degree of the hard polynomial family can be further relaxed a bit, at the cost of strengthening the hardness assumption, and increasing the running time of the resulting PIT algorithm<sup>5</sup>. We leave it to the interested reader to work out these details.

► **Remark.** In general, explicit polynomial families do not have to be multilinear. But, if we have a hard polynomial which is not multilinear, and has a polynomial degree in each variable, we can derive from it an explicit hard multilinear polynomial with only a polynomial deterioration in the hardness parameters. More precisely, replacing  $x_i^r$ , for  $r > 1$  with  $y_{i_0}^{r_0} \cdots y_{i_s}^{r_s}$ , where  $(r_0 \dots r_s)$  is the binary representation of  $r$ , gives a new multilinear polynomial in a slightly larger number of variables. This polynomial is at least as hard as the original polynomial which can be recovered from it by the substitution  $y_{i_j} = x_i^{2^j}$ .

As discussed earlier, Theorem 3 is closely related to the main result in [5]. We now discuss their similarities and differences.

### Comparison with [5]

- **Degree constraint on the hard polynomial.** While Theorem 3 requires that the hard polynomial on  $m$  variables has degree at most  $O(\log^2 m / \log^2 \log m)$ , Dvir et al. [5] did not have a similar constraint.
- **Individual degree constraint for PIT.** In [5], the authors get PIT for low depth circuits with bounded individual degree, whereas our Theorem 3 does not make any assumptions on individual degrees in this context.

As we alluded to earlier, the key technical challenge for extending the known hardness-randomness tradeoffs for general circuits [10] to restricted circuit classes like formulas or bounded depth circuits comes from the absence of an analog of Kaltofen's result [12] about closure under factoring for these restricted classes. More specifically, understanding the following questions seems necessary for adapting the proof strategy in [10] to other restricted classes of circuits.

► **Question 4.** *Let  $P(\vec{x}, y) \in \mathbb{F}[\vec{x}, y]$  be a polynomial of degree  $r$  and let  $f \in \mathbb{F}[\vec{x}]$  be a polynomial of degree  $d$  such that  $P(\vec{x}, f) \equiv 0$ . Assuming  $P$  can be computed by a low depth circuit (or arithmetic formula) of size at most  $s$ , can  $f$  be computed by a low depth circuit (or arithmetic formula) of size at most  $\text{poly}(s, n, d, r)$ ?*

In [5], the authors partially answer this question by showing that under the hypothesis of Question 4, the polynomial  $f$  can be computed by a low depth circuit of size at most

---

<sup>5</sup> If we assume a sub-exponential lower bound, then we can get a quasi-polynomial time PIT. Note that this is the parameter region used in [5]

$\text{poly}(s, r, d^{\deg_y(P)})$ . Thus, for the case of polynomials  $P$  which have small individual degree with respect to  $y$ , they answer the question in affirmative.

Our main technical observation is the following result, which gives an upper bound on the *low depth* circuit complexity of roots of low degree of a multivariate polynomial which has a small low depth circuit.

► **Theorem 5.** *Let  $P \in \mathbb{F}[\vec{x}, y]$  be a polynomial of degree at most  $r$  in  $n + 1$  variables that can be computed by an arithmetic circuit of size  $s$  of depth at most  $\Delta$ . Let  $f \in \mathbb{F}[\vec{x}]$  be a polynomial of degree at most  $d$  such that*

$$P(\vec{x}, f) = 0.$$

*Then,  $f$  can be computed by a circuit of depth at most  $\Delta + 3$  and size at most  $O((srn)^{10}d^{O(\sqrt{d})})$ .*

### 1.3 Proof Overview

The proof of Theorem 3 is very much along the lines of the proofs of similar results in [10] and [5]. In particular, all our technical contributions are confined to the proof of Theorem 5, which when combined with the standard machinery of Nisan-Wigderson designs yields Theorem 3. Our proof of Theorem 5 also mirrors the proof of the analogous theorem about the structure of roots in [5]. We now outline the main steps, and point out the differences between the proofs.

The first step in the proof is to show that one can use the standard Hensel Lifting to iteratively obtain better approximations of the root  $f$  given a circuit for  $P(\vec{x}, y)$ . More formally, in the  $k^{\text{th}}$  step, we start with a polynomial  $h_k$  which agrees with  $f$  on all monomials of degree at most  $k$ , and use it to obtain a polynomial  $h_{k+1}$  which agrees with  $f$  on all monomials of degree at most  $k + 1$ . Moreover, the proof shows that if  $h_k$  has a small circuit, then  $h_{k+1}$  has a circuit which is only slightly larger than that of  $h_k$ . This iterative process starts with the constant term of  $f$ , which trivially has a small circuit. Thus, after  $d$  iterations, we have a polynomial  $h_d$  such that the root  $f$  is the sum of the homogeneous components of  $h_d$  of degree at most  $d$ . This lifting step is exactly the same as that in [5] or in some of the earlier works on polynomial factorization [3], and is formally stated in Lemma 16.

The key insight of Dvir et al. [5] was that if  $\deg_y(P) = t$ , and  $C_0(\vec{x}), C_1(\vec{x}), \dots, C_t(\vec{x})$  are polynomials such that  $P(\vec{x}, y) = \sum_{i=0}^t C_i(\vec{x})y^i$ , then for every  $k \in \{0, 1, \dots, d\}$ , we have a polynomial  $B_k$  of degree at most  $k$  such that

$$h_k(\vec{x}) = B_k(C_0(\vec{x}), C_1(\vec{x}), \dots, C_t(\vec{x})).$$

Now, consider the case when  $t \ll n$  (for instance  $t = O(1)$ ). It follows from standard interpolation results for low depth circuits (see Lemma 12) that each of the polynomials  $C_i(\vec{x})$  has a circuit of size  $O(sr)$  and depth  $\Delta$  since  $P$  has a polynomial of size  $s$  and depth  $\Delta$ . Thus,  $h_d(\vec{x})$  can be written as a sum of at most  $\binom{d+t}{t} = O(d^t)$  monomials if we treat each  $C_i$  as a formal variable. Plugging in the small depth  $\Delta$  circuits for each  $C_i$ , and standard interpolation (Lemma 12), it follows that  $f$  has a circuit of size  $\text{poly}(s, n, d^t)$  of depth  $\Delta + O(1)$ .

Observe that this size bound of  $\text{poly}(s, n, d^t)$  is small only when  $t$  is small. For instance, when  $t > n$ , this bound becomes trivial. Our key observation is that independently of  $t$ , there is a set of  $d + 1$  polynomials  $g_0(\vec{x}), g_1(\vec{x}), \dots, g_d(\vec{x})$  of degree at most  $d$ , and polynomials  $A_0, A_1, \dots, A_k$  on  $d + 1$  variables such that for every  $k \in \{0, 1, \dots, d\}$ ,

$$h_k(\vec{x}) = A_k(g_0(\vec{x}), g_1(\vec{x}), \dots, g_d(\vec{x})).$$

Moreover, for every  $k$ ,  $A_k$  has degree at most  $k$  and is computable by a circuit of size at most  $O(d^3)$ . This observation essentially decouples the number of *generators* from the individual degree of  $P$  in  $y$ , and is formally stated as Lemma 18. Also, each of these generators  $g_i$  can be computed by a circuit of size  $\text{poly}(s, r)$  and depth  $\Delta$ . Thus, expressing  $A_d(z_0, z_1, \dots, z_d)$  as a sum of monomials, and then composing this representation with the circuits for  $g_0, g_1, \dots, g_d$  would give us a circuit of size  $\text{poly}(s, n, r, d, 4^d)$  of depth  $\Delta + O(1)$ . To get a sub-exponential dependence on  $d$  in the size, we do not write  $A_d(z_0, z_1, \dots, z_d)$  as  $\sum \prod$  circuit of size  $O(4^d)$ , but instead express it as a  $\sum \prod \sum$  circuit of size at most  $d^{O(\sqrt{d})}$ , using the depth reduction result of [8]<sup>6</sup>.

One point to note is that just from Kaltofen's result [12], it follows that  $f$  has an arithmetic circuit<sup>7</sup> of size  $\text{poly}(n)$ . Thus, from Theorem 9, it follows that  $f$  has a circuit of depth-3 of size at most  $n^{O(\sqrt{d})}$ . The key advantage of Theorem 5 over this bound is that the exponential term is  $d^{O(\sqrt{d})}$  and not of the form  $n^{d^\epsilon}$ . For  $d \leq \log^2 n / \log^2 \log n$ ,  $d^{O(\sqrt{d})}$  is bounded by a polynomial in  $n$  and so the final bound is meaningful.

We end this section with a short discussion on the *low degree* condition in the hypothesis of Theorem 3.

## 1.4 The Low Degree Condition

An intriguing question is to understand how restrictive the “low degree” condition in the hardness assumption of Theorem 3 is. More formally, is the question of proving super-polynomial lower bounds for constant depth circuits for an explicit polynomial family of low degree much harder than the question of proving super-polynomial lower bound for constant depth circuits for an explicit polynomial family of potentially larger degree<sup>8</sup>? Currently, we do not even know quadratic lower bounds for arithmetic circuits of constant depth, and so, perhaps we are quite far from understanding this question.

It is, however, easy to see that some of the known lower bounds for low depth circuits carries over to the low degree regime. For instance, the proofs of super-polynomial lower bounds for homogeneous depth-3 circuits by Nisan and Wigderson [19], super-polynomial lower bounds for homogeneous depth-4 circuits based on the idea of shifted partial derivatives (see for example, [9, 13, 7, 17]) and super-linear lower bound due to Raz [22] do not require the degree of the hard function to be large.

There are some known exceptions to this. For instance, lower bounds for homogeneous depth-5 circuits over finite fields due to Kumar and Saptharishi [16] are of the form  $2^{\Omega(\sqrt{d})}$  and become trivial if  $d < \log^2 n$ . Another result which distinguishes the low degree and high degree regime is a separation between homogeneous depth-5 and homogeneous depth-4 circuit [16] which is only known to be true in the low degree regime (degree less than  $\log^2 n$ ).

Another result of relevance is a result of Raz [23], which shows that constructing an explicit family of tensors  $T_n : [n]^d \rightarrow \mathbb{F}$ , of rank at least  $n^{d(1-o(1))}$  implies super-polynomial lower bound for arithmetic formulas, provided  $d \leq O(\log n / \log \log n)$ . As far as we know, we do not know of such connections in the regime of high degree.

One prominent family of lower bound results which do not seem to generalize to this low degree regime are the super-polynomial lower bounds for multilinear formulas [21], and multilinear constant depth circuits [25]. In fact, the results in [23] show that super-polynomial

<sup>6</sup> See Theorem 9 for a formal statement of this result.

<sup>7</sup> Of potentially very large depth.

<sup>8</sup> In general, the degree only has to be upper bounded by a polynomial function in the number of variables.

lower bounds for set multilinear formulas for polynomials of degree at most  $O(\log n / \log \log n)$  implies super-polynomial lower bounds for general arithmetic formulas.

In the context of polynomial factorization, low degree factors of polynomials with small circuits have been considered before. For instance, Forbes [6] gave a quasi-polynomial time deterministic algorithm to test if a given polynomial of constant degree divides a given sparse polynomial. Extending this result to even testing if a given sparse polynomial divides another given sparse polynomial remains an open problem.

## 2 Preliminaries

We start by setting up some notation and stating some basic definitions and results from prior work which will be used in our proofs.

### 2.1 Notations

- We use boldface letters  $\vec{x}, \vec{y}, \vec{z}$  to denote tuples of variables.
- For a polynomial  $P$ ,  $\deg(P)$  denotes the total degree of  $P$  and  $\deg_y(P)$  denotes the total degree of  $P$  with respect to the variable  $y$ .
- Throughout this paper, we state and prove our results when the underlying field  $\mathbb{F}$  is the field of rational numbers  $\mathbb{Q}$ , even though all our results hold as long as the field is of sufficiently large or zero characteristic.
- Let  $P \in \mathbb{F}[\vec{x}]$  be a polynomial of degree equal to  $d$ . For every  $k \in \mathbb{N}$ ,  $\mathcal{H}_k[P]$  denotes the homogeneous component of  $P$  of degree  $k$ . Similarly,  $\mathcal{H}_{\leq k}[P]$  is defined to be equal  $\sum_{i=0}^k \mathcal{H}_i[P]$ .
- For an arithmetic circuit  $C$ , we use  $\text{size}(C)$  to denote the number of wires in  $C$ . The depth of  $C$  is the length of the longest path from any output gate to any input gate.
- Throughout this paper, we assume that all our circuits are layered with alternating layers of addition and multiplication gates. Moreover, we always assume that the top layer is of addition gates. For instance, a depth-3 circuit is of the form  $\Sigma \Pi \Sigma$  and a depth-4 circuit is of the form  $\Sigma \Pi \Sigma \Pi$ .

### 2.2 Derivatives

We start by defining derivatives of a polynomial. For the ease of presentation, we work with the notion of the slightly non-standard notion of *Hasse* derivatives even though we work with fields of characteristic zero.

► **Definition 6** (Derivatives). Let  $\mathbb{F}$  be any field and let  $P(y) \in \mathbb{F}[y]$  be a polynomial. Then for every  $k \in \mathbb{N}$ , the partial derivative of  $P$  of order  $k$  with respect to  $y$  denoted by  $\frac{\partial^k P(y)}{\partial y^k}$  or  $P^{(k)}(y)$  is defined as the coefficient of  $z^k$  in the polynomial  $P(y+z)$ .

We also use  $P'(y)$  and  $P''(y)$  to denote the first and second order derivatives of  $P$  respectively. An immediate consequence of this definition is the following lemma.

► **Lemma 7** (Taylor's expansion). Let  $P(y) \in \mathbb{F}[y]$  be a polynomial of degree  $d$ . Then,

$$P(y+z) = P(y) + z \cdot P'(y) + z^2 \cdot P^{(2)}(y) + \dots + z^d \cdot P^{(d)}(y).$$

## 2.3 Depth Reductions

We will use the following depth reduction theorems as a blackbox for our proofs.

► **Theorem 8** (Depth reduction to depth- $2k$  [1, 14, 28]). *Let  $k$  be a positive integer and  $\mathbb{F}$  be any field. If  $P(\vec{x}) \in \mathbb{F}[\vec{x}]$  is an  $n$ -variate polynomial of degree  $d$  that be computed by an arithmetic circuit  $\Psi$  of size at most  $s$ , then  $P$  can be computed by a depth  $2k$  circuit of size at most  $(snd)^{O(d^{1/k})}$ .*

Invoked with  $k = 2$  the above theorem gives a circuit of depth 4 for the polynomial  $P$  of size  $s^{O(\sqrt{d})}$ . The next depth reduction result gives a further reduction to depth-3, as long as the field is of characteristic zero, and will be useful for our proof.

► **Theorem 9** (Depth reduction to depth-3 [8]). *Let  $P(\vec{x}) \in \mathbb{Q}[\vec{x}]$  be an  $n$ -variate polynomial of degree  $d$  that can be computed by an arithmetic circuit  $\Psi$  of size at most  $s$ . Then,  $P$  can be computed by a  $\sum \prod \sum$  circuit of size at most  $(snd)^{O(\sqrt{d})}$ .*

## 2.4 Explicit Polynomials

► **Definition 10** ([5]). Let  $\{f_m\}$  be a family of multilinear polynomials such that  $f_m \in \mathbb{F}[x_1, x_2, \dots, x_m]$  for every  $m$ . Then, the family  $\{f_m\}$  is said to be explicit if the following two conditions hold.

- All the coefficients of  $f_m$  have bit complexity polynomial in  $m$ .
- There is an algorithm which on input  $m$  outputs the list of all  $2^m$  coefficients of  $f_m$  in time  $2^{O(m)}$ .

## 2.5 Extracting Homogeneous Components

For our proofs, we will also rely on the following classical result of Strassen, which shows that if a polynomial  $P$  has a small circuit, then all its low degree homogeneous components also have small circuits.

► **Theorem 11** (Homogenization). *Let  $\mathbb{F}$  be any field, and let  $\Psi \in \mathbb{F}[\vec{x}]$  be an arithmetic circuit of size at most  $s$ . Then, for every  $k \in \mathbb{N}$ , there is a homogeneous circuit  $\Psi_k$  of formal degree at most  $k$  and size at most  $O(k^2 s)$ , such that*

$$\Psi_k = \mathcal{H}_k[\Psi].$$

Theorem 11 gives us a way of extracting homogeneous components of the polynomial computed by a given circuit. One drawback of Theorem 11 is that the depth of  $\Psi_k$  could be much larger than the depth of  $\Psi$ . Thus, given a low depth circuit (and hence, unbounded in-degree circuit) for a polynomial  $P$ , it is not clear if the homogeneous components of  $P$  also have small low depth circuits. The following standard trick implies this observation, and would be useful for our proof.

► **Lemma 12** (Interpolation). *Let  $\mathbb{F}$  be any field with at least  $d + 1$  elements. Let  $P(\vec{x}, y) \in \mathbb{F}[\vec{x}, y]$  be a polynomial of degree at most  $d$ . Let  $C_0(\vec{x}), C_1(\vec{x}), \dots, C_d(\vec{x}) \in \mathbb{F}[\vec{x}]$  be polynomials such that  $P(\vec{x}, y) = \sum_{j=0}^d y^j \cdot C_j(\vec{x})$ . Then, if  $P(\vec{x}, y)$  has a circuit of size at most  $s$  and depth at most  $\Delta$ , then for every  $j \in \{0, 1, \dots, d\}$ ,  $C_j(\vec{x})$  has a circuit of size at most  $O(sd)$  and depth  $\Delta$ .*

We refer the reader to excellent surveys of Shpilka and Yehudayoff [27] and Saptharishi [26] for a proof of these results.

## 2.6 Hitting Sets

► **Definition 13.** A set of points  $\mathcal{P}$  is said to be a hitting set for a class  $\mathcal{C}$  of circuits, if for every  $C \in \mathcal{C}$  which is not identically zero, there is an  $\vec{a} \in \mathcal{P}$  such that  $C(\vec{a}) \neq 0$ .

Clearly, deterministic and efficient construction of a hitting set of small size for a class  $\mathcal{C}$  of circuits immediately implies a deterministic PIT algorithm for  $\mathcal{C}$ . PIT algorithms designed in this way are also *blackbox*, in the sense that they do not have to look inside into the wiring of the circuit to decide if it computes a polynomial which is identically zero. The PIT algorithms in this paper are all blackbox in this sense.

## 2.7 Nisan-Wigderson Designs

We state the following well known result of Nisan and Wigderson [18] on the explicit construction of combinatorial designs.

► **Theorem 14** ([18]). *Let  $n, m$  be positive integers such that  $n < 2^m$ . Then, there is a family of subsets  $S_1, S_2, \dots, S_n \subseteq [\ell]$  with the following properties.*

- For each  $i \in [n]$ ,  $|S_i| = m$ .
- For each  $i, j \in [n]$ , such that  $i \neq j$ ,  $|S_i \cap S_j| \leq \log n$ .
- $\ell = O\left(\frac{m^2}{\log n}\right)$ .

Moreover, such a family of sets can be constructed via a deterministic algorithm in time  $\text{poly}(n, 2^\ell)$ .

## 2.8 Schwartz-Zippel Lemma

We now state the well known Schwartz-Zippel lemma.

► **Lemma 15** (Schwartz-Zippel). *Let  $\mathbb{F}$  be a field, and let  $P \in \mathbb{F}[\vec{x}]$  be a non-zero polynomial of degree (at most)  $d$  in  $n$  variables. Then, for any finite set  $S \subset \mathbb{F}$  we have*

$$|\{\vec{a} \in S^n : P(\vec{a}) = 0\}| \leq d|S|^{n-1}.$$

In particular, if  $|S| \geq d + 1$ , then there exists some  $\vec{a} \in S^n$  satisfying  $P(\vec{a}) \neq 0$ . This gives us a brute force deterministic algorithm, running in time  $(d + 1)^n$ , to test if an arithmetic circuit computing a polynomial of degree at most  $d$  in  $n$  variables is identically zero.

## 3 Low Degree Roots of Polynomials with Shallow Circuits

In this section, we prove Theorem 5, which is also our main technical result. We start with the following lemma, which gives us a way of *approximating* the root of a polynomial to higher and higher accuracy, in an iterative manner. The lemma is a standard example of Hensel Lifting (in fact, sloppy Hensel Lifting), which appears in many of prior works in this area including [5]. The statement and the proof below, are from the work of Dvir et al [5].

► **Lemma 16** (Hensel Lifting [5]). *Let  $P \in \mathbb{F}[\vec{x}, y]$  and  $f \in \mathbb{F}[\vec{x}]$  be polynomials such that  $P(\vec{x}, f) = 0$  and  $\mathcal{H}_0 \left[ \frac{\partial P}{\partial y}(\vec{x}, f(\vec{x})) \right] = \delta \neq 0$ . Let  $i \in \{1, 2, \dots, \deg(f)\}$  be any number. If  $h \in \mathbb{F}[\vec{x}]$  is a polynomial such that  $\mathcal{H}_{<i-1}[f] = \mathcal{H}_{<i-1}[h]$ , then*

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i} \left[ h - \frac{P(\vec{x}, h)}{\delta} \right].$$



### 13:10 Hardness vs Randomness for Bounded Depth Arithmetic Circuits

**Proof.** For the rest of the proof, we think of  $P(\vec{x}, y)$  as an element of  $\mathbb{F}[\vec{x}][y]$ . Henceforth, we drop the variables  $\vec{x}$  everywhere, and think of  $P$  as a univariate in  $y$ . Thus,  $P(y) = P(\vec{x}, y)$ . For brevity, we denote  $\mathcal{H}_j[f]$  by  $f_j$  for every  $j \in \mathbb{N}$ .

From the hypothesis, we know that  $P(f) = 0$ . Therefore,  $\mathcal{H}_{\leq i}(P(f)) = \mathcal{H}_{\leq i-1}[P(f)] = 0$ . Moreover, since  $\mathcal{H}_{\leq i-1}[h] = \mathcal{H}_{\leq i-1}[f]$ , we get that  $\mathcal{H}_{\leq i-1}[P(f)] = \mathcal{H}_{\leq i-1}[P(h)] = 0$ . So, we have

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i}[P(f)] \\ &= \mathcal{H}_{\leq i}[P(h + (f_i - h_i))] \end{aligned}$$

Now, by using Lemma 7, we get the following equality.

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i} \left[ P(h) + P'(h) \cdot (f_i - h_i) + P''(h) \cdot (f_i - h_i)^2 + \dots + P^{(r)}(h) \cdot (f_i - h_i)^r \right] \\ &= \mathcal{H}_{\leq i}[P(h)] + \mathcal{H}_{\leq i}[P'(h) \cdot (f_i - h_i)] + \dots + \mathcal{H}_{\leq i}[P^{(r)}(h) \cdot (f_i - h_i)^r] \end{aligned}$$

Here,  $r$  denotes the degree of  $P$ . Since every monomial in  $f_i - h_i$  has degree equal to  $i$ , any term in the above summand which is divisible by  $(f_i - h_i)^2$  does not contribute any monomial of degree at most  $i$ . Thus, we have the following.

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i}[P(h)] + \mathcal{H}_{\leq i}[P'(h) \cdot (f_i - h_i)] \\ &= \mathcal{H}_{\leq i}[P(h)] + \mathcal{H}_0[P'(h)] \cdot (f_i - h_i). \end{aligned}$$

Now, we know that  $\mathcal{H}_0[P'(h)] = \mathcal{H}_0[P'(f)] = \delta \neq 0$ . Thus,

$$f_i = h_i - \frac{\mathcal{H}_i[P(h)]}{\delta}.$$

Since  $\mathcal{H}_{\leq i-1}[P(h)]$  is identically zero, we get,

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i} \left[ h - \frac{P(h)}{\delta} \right]. \quad \blacktriangleleft$$

For our proof, we shall look at the structure of the outcome of the lifting operation in Lemma 16 more closely. Before proceeding further, we need the following crucial lemma.

► **Lemma 17.** *Let  $P(\vec{x}, y) \in \mathbb{F}[\vec{x}, y]$  be a polynomial of degree at most  $r$ , let  $\alpha \in \mathbb{F}$  be a field element and  $d \in \mathbb{N}$  be a positive integer. Let  $\mathcal{G}'(P, \alpha, d)$  be the set of polynomials defined as follows.*

$$\mathcal{G}'(P, \alpha, d) = \left\{ \mathcal{H}_{\leq d} \left[ \frac{\partial^j P}{\partial y^j}(\vec{x}, \alpha) \right] - \mathcal{H}_0 \left[ \frac{\partial^j P}{\partial y^j}(\vec{x}, \alpha) \right] : j \in \{0, 1, 2, \dots, d\} \right\}.$$

Let  $\mathcal{G}(P, \alpha, d)$  be the subset of  $\mathcal{G}'(P, \alpha, d)$  consisting of all non-zero polynomials. Then, the following statements are true.

- For every  $g \in \mathcal{G}(P, \alpha, d)$ , the degree of every non-zero monomial in  $g$  is at least 1 and at most  $d$ .
- $|\mathcal{G}| \leq d + 1$ .
- If  $P$  has a circuit of size at most  $s$  and depth  $\Delta$ , then every  $g \in \mathcal{G}(P, \alpha, d)$  has a circuit of size at most  $O(sr^3d^2)$  and depth  $\Delta$ .



► **Remark.**  $\mathcal{G}'$  contains the non-constant part of the partial derivatives of  $P$  at  $\alpha$  up to order  $d$ . Note that  $\mathcal{G}'$  may contain the zero polynomial, but  $\mathcal{G}$  is the subset of  $\mathcal{G}'$  without the zero polynomial.

**Proof.** The first two items follow immediately from the definition of  $\mathcal{G}(P, \alpha, d)$ . We focus on the proof of the third item. Let  $C_0(\vec{x}), C_1(\vec{x}), \dots, C_r(\vec{x})$  be polynomials such that

$$P(\vec{x}, y) = \sum_{i=0}^r C_i(\vec{x}) \cdot y^i .$$

Now, for any  $j \in \{0, 1, 2, \dots, d\}$ , by Definition 6,  $\frac{\partial^j P}{\partial y^j}(\vec{x}, y)$  is the coefficient of  $z^j$  in  $P(\vec{x}, y+z)$ . Moreover,

$$\begin{aligned} P(\vec{x}, y+z) &= \sum_{i=0}^r C_i(\vec{x}) \cdot (y+z)^i , \\ &= \sum_{i=0}^r C_i(\vec{x}) \cdot \left( \sum_{j=0}^i \binom{i}{j} z^j y^{i-j} \right) , \\ &= \sum_{j=0}^r \left( \sum_{i=j}^r \binom{i}{j} C_i(\vec{x}) \cdot y^{i-j} \right) \cdot z^j . \end{aligned}$$

Thus, for every  $j \in \{0, 1, \dots, d\}$ , the coefficient of  $z^j$  in  $P(\vec{x}, y+z)$  is given by  $\sum_{i=j}^r \binom{i}{j} C_i(\vec{x}) \cdot y^{i-j}$ . From Lemma 12, we know that each  $C_i(\vec{x})$  has a circuit of depth  $\Delta$  and size at most  $O(sr)$ . Thus, we can obtain a circuit for  $\binom{i}{j} C_i(\vec{x}) \cdot y^{i-j}$  by adding an additional layer of  $\times$  gates on top of the circuit for  $C_i(\vec{x})$ . This increases the size by a multiplicative factor of  $r$ , and the depth by 1. However, observe that this increase in depth is not necessary. Since, an expression of the form  $y^i \cdot (\sum_a \prod_b Q_{a,b})$  can be simplified to  $\sum_a y^i \cdot (\prod_b Q_{a,b})$ . Thus, the multiplication by  $y^i$  can be absorbed in the product layer below the topmost layer of the circuits for  $C_i(\vec{x})$ , and this does not incur any additional increase in size. Thus, the polynomials  $\frac{\partial^j P}{\partial y^j}(\vec{x}, y)$ , and hence  $\frac{\partial^j P}{\partial y^j}(\vec{x}, \alpha)$  have a circuit of size at most  $O(sr^3)$  and depth at most  $\Delta$ . To compute the homogeneous components of these polynomials, which are essentially the elements of  $\mathcal{G}(P, \alpha, d)$ , we just use Lemma 12. This increases the size by a factor of at most  $O(d^2)$  while keeping the depth the same. ◀

We now state our key technical observation.

► **Lemma 18.** *Let  $P \in \mathbb{F}[\vec{x}, y]$  and  $f \in \mathbb{F}[\vec{x}]$  be polynomials of degree  $r$  and  $d$  respectively such that  $P(\vec{x}, f) = 0$  and  $\mathcal{H}_0 \left[ \frac{\partial P}{\partial y}(\vec{x}, f(\vec{x})) \right] = \delta \neq 0$ . Let the polynomials in the set  $\mathcal{G}(P, \mathcal{H}_0[f], d)$  be denoted by  $g_0, g_1, \dots, g_d$ . Then, for every  $i \in \{1, 2, \dots, d\}$ , there is a polynomial  $A_i(\vec{z})$  in  $d+1$  variables such that the following are true.*

- $\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \dots, g_d)]$ , and
- $A_i(\vec{z})$  is computable by a circuit of size at most  $10d^2i$ .

This is the analog of the main technical lemma in [5], which we state below.

► **Lemma 19 ([5]).** *Let  $P \in \mathbb{F}[\vec{x}, y]$  and  $f \in \mathbb{F}[\vec{x}]$  be polynomials of degree  $r$  and  $d$  respectively such that  $P(\vec{x}, f) = 0$  and  $\mathcal{H}_0 \left[ \frac{\partial P}{\partial y}(\vec{x}, f(\vec{x})) \right] = \delta \neq 0$ . Let  $P(\vec{x}, y) = \sum_{i=0}^k C_i(\vec{x}) \cdot y^i$ . Then, for every  $i \in \{1, 2, \dots, \deg(f)\}$ , there is a polynomial  $A_i(\vec{z})$  in  $k+1$  variables such that,*

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \dots, C_k)] .$$

## 13:12 Hardness vs Randomness for Bounded Depth Arithmetic Circuits

The difference between these lemmas is that in [5], it is shown that there is a set of polynomials of size at most  $\deg_y(P) + 1$  which *generate* every homogeneous component of the root  $f$ . Thus, in the regime of bounded individual degree, the size of this generating set is very small. However, when  $\deg_y(P) \geq n$ , Lemma 19 does not say anything non-trivial since  $f$  can be trivially written as a polynomial in the  $n$  original variables. In contrast, Lemma 18 continues to say something non-trivial, as long as  $d \ll n$ , regardless of the value of  $\deg_y(P)$ . We now proceed with the proof.

**Proof of Lemma 18.** For the rest of the proof, we think of  $P(\vec{x}, y)$  as an element of  $\mathbb{F}[\vec{x}][y]$ . So, we drop the variables  $\vec{x}$  everywhere, and think of  $P$  as a univariate in  $y$ . Thus,  $P(y) = P(\vec{x}, y)$ . For brevity, we denote  $\mathcal{H}_j[f]$  by  $f_j$  for every  $j \in \mathbb{N}$ . We also use  $\mathcal{G}$  for  $\mathcal{G}(P, f_0, d)$ . The proof will be by induction on  $i$  and crucially use Lemma 16.

- **Base case.** We first prove the lemma for  $i = 1$ . We invoke Lemma 16 with  $i = 1$  and  $h = f_0$ . We get that

$$\mathcal{H}_{\leq 1}[f] = \mathcal{H}_{\leq 1} \left[ f_0 - \frac{P(f_0)}{\delta} \right].$$

The proof follows by observing that  $f_0, \delta$  are constants and  $\mathcal{H}_1[P(f_0)] = \mathcal{H}_1[g_0]$  where  $g_0 = \mathcal{H}_{\leq d}[P(f_0)] - \mathcal{H}_0[P(f_0)] \in \mathcal{G}$ .

- **Induction step.** We assume that the claim in the lemma holds up to homogeneous components of degree at most  $i - 1$ , and argue that it holds for  $\mathcal{H}_{\leq i}[f]$ . We invoke Lemma 16 with  $h = A_{i-1}(g_0, g_1, \dots, g_d)$ , which exists by the induction hypothesis.

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i} \left[ h - \frac{P(h)}{\delta} \right].$$

Recall that  $\mathcal{H}_0(h) = \mathcal{H}_0(f)$ . Thus,  $h = f_0 + \tilde{h}$ , where every monomial in  $\tilde{h}$  has degree at least 1. By Lemma 7,

$$P(f_0 + \tilde{h}) = P(f_0) + P'(f_0) \cdot \tilde{h} + \dots + P^{(r)}(f_0) \cdot \tilde{h}^r.$$

Thus, as  $\tilde{h}$  has degree at least 1, we have

$$\begin{aligned} \mathcal{H}_{\leq i}[f] &= \mathcal{H}_{\leq i} \left[ h - \frac{1}{\delta} \cdot \left( P(f_0) + P'(f_0) \cdot \tilde{h} + \dots + P^{(r)}(f_0) \cdot \tilde{h}^r \right) \right], \\ &= \mathcal{H}_{\leq i} \left[ h - \frac{1}{\delta} \cdot \left( P(f_0) + P'(f_0) \cdot \tilde{h} + \dots + P^{(i)}(f_0) \cdot \tilde{h}^i \right) \right]. \end{aligned}$$

Since we are only interested in  $i \leq d$ , the following equality is also true.

$$\begin{aligned} \mathcal{H}_{\leq i}[f] &= \mathcal{H}_{\leq i} \left[ h - \frac{1}{\delta} \cdot \left( \mathcal{H}_{\leq d}[P(f_0)] + \mathcal{H}_{\leq d}[P'(f_0)] \cdot \tilde{h} + \dots + \mathcal{H}_{\leq d}[P^{(i)}(f_0)] \cdot \tilde{h}^i \right) \right]. \end{aligned}$$

Observe that for every  $j \in \{0, 1, \dots, d\}$ ,  $\mathcal{H}_{\leq d}[P^{(j)}(f_0)]$  is an affine form in the elements of  $\mathcal{G}$ . For every  $j \in \{0, 1, 2, \dots, i\}$ , let  $\ell_j(\vec{z})$  be an affine form such that  $\ell_j(g_0, g_1, \dots, g_d) =$

---

<sup>9</sup> In fact, they are an affine form in one variable.

$\mathcal{H}_{\leq d} [P^{(j)}(f_0)]$ . Now, we define  $A_i(\vec{z})$  as

$$A_i(\vec{z}) \equiv A_{i-1}(\vec{z}) - \frac{1}{\delta} (\ell_0(\vec{z}) + \ell_1(\vec{z}) \cdot (A_{i-1}(\vec{z}) - f_0) + \cdots + \ell_i(\vec{z}) \cdot (A_{i-1}(\vec{z}) - f_0)^i) .$$

The first item in the statement of the lemma is true, just by the definition of  $A_i(\vec{z})$  above. We now argue about the circuit size of  $A_i(\vec{z})$ . Each affine form  $\ell_i(\vec{z})$  can be computed by a circuit of size at most  $O(d)$ . Thus, given a circuit of  $A_{i-1}(\vec{z})$ , we can obtain a circuit for  $A_i(\vec{z})$  by adding at most  $10d^2$  additional gates. Thus,  $A_i(\vec{z})$  can be computed by a circuit of size at most  $10d^2(i-1) + 10d^2 = 10d^2i$  gates. ◀

We are now ready to complete the proof of Theorem 5.

**Proof of Theorem 5.** The first step is to massage the circuit for  $P$  so that the hypothesis of Lemma 18 holds. We will have to keep track of the size and depth blow ups incurred in the process. We begin by ensuring that  $f$  is a root of multiplicity 1 of some polynomial related to  $P$ .

### Reducing multiplicity of the root $f$

Let  $P(\vec{x}, y) = \sum_{i=0}^r y^i C_i(\vec{x})$ . Let  $m \geq 1$  be the multiplicity of  $f$  as a root of  $P(\vec{x}, y)$ . Thus,  $\frac{\partial^j P}{\partial y^j}(\vec{x}, f) = 0$  for  $j \in \{0, 1, 2, \dots, m-1\}$ , but  $\frac{\partial^m P}{\partial y^m}(\vec{x}, f) \neq 0$ . The idea is to just work with the polynomial  $\tilde{P} = \frac{\partial^{m-1} P}{\partial y^{m-1}}(\vec{x}, y)$  for the rest of the proof. Clearly,  $f$  is a root of multiplicity exactly 1 of  $\tilde{P}$ . We only need to ensure that  $\tilde{P}$  can also be computed by a small low depth circuit. This follows from the proof of the third item in Lemma 17, where we argued that  $\frac{\partial^j P}{\partial y^j}(\vec{x}, y)$  has a depth  $\Delta$  circuit of size  $O(sr^3)$ .

### Translating the origin

From the step above, we can assume without loss of generality that  $\frac{\partial P}{\partial y}(\vec{x}, f) \neq 0$ . Thus, there is a point  $\vec{a} \in \mathbb{F}^n$  such that  $\frac{\partial P}{\partial y}(\vec{a}, f(\vec{a})) \neq 0$ . By translating the origin, we will assume that  $\frac{\partial P}{\partial y}(0, f(0)) \neq 0$ . This increases the depth of the circuit by at most 1, as it could involve replacing every variable  $x_i$  by  $x_i + a_i$ , and the size by at most a factor  $n$ .

### Degree of $A_d$

From Lemma 18, we know that the polynomial  $A_d(\vec{z})$  has a circuit of size at most  $O(d^3)$ . To obtain a circuit for  $f$ , we first prune away all the homogeneous components of  $A_d(\vec{z})$  of degree larger than  $d$ . Recall that by definition, every polynomial  $g_i \in \mathcal{G}$  has degree at least 1, and that  $f = \mathcal{H}_{\leq d} [A_d(g_1, g_2, \dots, g_d)]$ . Thus, any monomial of degree strictly greater than  $d$  in  $A_d(\vec{z})$  contributes no monomial of degree at most  $d$  in the variables  $\vec{x}$  in the composed polynomial  $A_d(g_1, g_2, \dots, g_d)$ , and hence does not contribute anything to the computation of  $f$ . So, we can confine ourselves to working with the homogeneous components of  $A_d(\vec{z})$  of degree at most  $d$ .

By Theorem 11, we know that given a circuit for  $A_d(\vec{z})$ , we can construct a circuit for  $\mathcal{H}_i [A_d(\vec{z})]$  by increasing the size of the circuit by a multiplicative factor of at most  $O(i^2)$ . Thus,  $\mathcal{H}_{\leq d} [A_d(\vec{z})]$  can be computed by a circuit of size at most  $O(d^3) \times \text{size}(A_d(\vec{z}))$ . Thus, for the rest of this argument, we will assume that  $A_d(\vec{z})$  has a circuit of size at most  $O(d^6)$  and degree at most  $d$ , and

$$f = \mathcal{H}_{\leq d} [A_d(g_1, g_2, \dots, g_d)] .$$

**Circuit for  $A_d(\vec{z})$  of small depth**

Given that  $A_d(\vec{z})$  has a circuit of size  $O(d^6)$  and degree at most  $d$ , by Theorem 9, we know that  $A_d(\vec{z})$  can be computed by a  $\sum \prod \sum$  circuit  $\Psi$  of size at most  $d^{O(\sqrt{d})}$ <sup>10</sup>. Similar results follow from the application of Theorem 8.

**Circuit for  $f$  of small depth**

Composing the  $\sum \prod \sum$  circuit  $\Psi$  for  $A_d(\vec{z})$  with the circuits of  $g_1, g_2, \dots, g_d \in \mathcal{G}$ , we get a circuit  $\Psi'$  with the following properties.

- The size of  $\Psi'$  is at most  $(srn)^{10} \cdot d^{O(\sqrt{d})}$ .
- The depth of  $\Psi'$  is at most  $\Delta + 3$ . This follows by combining the bottom  $\sum$  layer of the  $\sum \prod \sum$  circuit for  $A_d(\vec{z})$  with the top  $\sum$  layer of the circuits for  $g_i \in \mathcal{G}$ .
- The degree of  $\Psi'$  is at most  $d^2$ . This is true because the degree of  $A_d(\vec{z})$  is at most  $d$  (as argued earlier in this proof), and the degree of every polynomial in  $\mathcal{G}$  is at most  $d$  (first item in Lemma 17).
- $f = \mathcal{H}_{<d}[\Psi'(\vec{x})]$ .

To obtain a circuit for  $f$ , we apply Lemma 12 to  $\Psi'$ . This increases the size of  $\Psi'$  by a multiplicative factor of at most  $O(d^2)$ , while the depth remains the same. This completes the proof of the theorem. ◀

**4 Deterministic Identity Testing using Hard Polynomials**

In this section, we use Theorem 5 to show that given a family of polynomials which are hard for depth  $\Delta$  circuits, we can do deterministic identity testing for  $\Delta - 5$  circuits in subexponential time. Since the content of this part are very similar to the proofs of similar statements in [10] and [5], we only outline the differences in the proofs (if any), and refer the reader to [5] for details. We start with the following lemma, which is the analog of Lemma 4.1 in [5].

► **Lemma 20** (Analog of Lemma 4.1 in [5]). *Let  $q(\vec{x}) \in \mathbb{F}[\vec{x}]$  be a (non-zero) polynomial of degree  $D$  in  $n$  variables, which can be computed by a circuit of size  $s$  and depth  $\Delta$ . Let  $m > \log n$  be an integer and let  $S_1, S_2, \dots, S_n \subseteq [\ell]$  be given by Theorem 14, so that  $\ell = O(m^2/\log n)$ ,  $|S_i| = m$ , and  $|S_i \cap S_j| \leq \log n$ . For a multilinear polynomial  $f \in \mathbb{F}[z_1, z_2, \dots, z_m]$  of degree  $d$ , put*

$$Q(\vec{y}) = Q(y_1, y_2, \dots, y_\ell) := q(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \dots, f(\vec{y}|_{S_n})) .$$

*If  $Q(\vec{y}) \equiv 0$ , then  $f(\vec{z})$  can be computed by an arithmetic circuit of size  $O((snD)^{12}d^{O(\sqrt{d})})$  and depth at most  $\Delta + 5$ .*

Note that the bound on the size of  $f$  remains non-trivial as long as  $d \ll m$ , while the individual degree of  $q$  is allowed to be unbounded, whereas the bound in [5] becomes trivial once  $\deg_y(q)$  is larger than  $m$ .

<sup>10</sup> Instead of Theorem 9, one could use Theorem 8 to get a better size bound than  $d^{O(\sqrt{d})}$  at the cost of increasing its depth appropriately. Also, see Remark 1.2. Also, this is one place where the underlying field plays a role, since Theorem 9 is not known to be true over general fields.

**Proof Sketch.** The proof is along the lines of the proof of Lemma 4.1 in [5]. We now give a sketch of the details. We first define the hybrid polynomials  $Q_0(\vec{x}, \vec{y}), Q_1(\vec{x}, \vec{y}), \dots, Q_n(\vec{x}, \vec{y})$  as follows.

$$Q_j(\vec{x}, \vec{y}) = q(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \dots, f(\vec{y}|_{S_j}), x_{j+1}, x_{j+2}, \dots, x_n) .$$

We know that  $Q_0(\vec{x}, \vec{y})$  is non-zero, whereas  $Q_n(\vec{x}, \vec{y})$  is identically zero. Thus, there is an  $i \in \{0, 1, \dots, n\}$  such that  $Q_i(\vec{x}, \vec{y}) \not\equiv 0$  and  $Q_{i+1}(\vec{x}, \vec{y}) \equiv 0$ . We now fix the variables  $x_{i+2}, x_{i+3}, \dots, x_n$  and the variables  $\{y_j : j \notin S_{i+1}\}$  to field constants while maintaining the non-zerosness of  $Q_i$ . This can be done via Lemma 15. Thus, we have a polynomial  $\tilde{q}$  by fixing the aforementioned variables such that the following two conditions hold.

$$\tilde{q}(f(\vec{y}|_{S_1 \cap S_{i+1}}), f(\vec{y}|_{S_2 \cap S_{i+1}}), \dots, f(\vec{y}|_{S_i \cap S_{i+1}}), x_{i+1}) \not\equiv 0 .$$

$$\tilde{q}(f(\vec{y}|_{S_1 \cap S_{i+1}}), f(\vec{y}|_{S_2 \cap S_{i+1}}), \dots, f(\vec{y}|_{S_i \cap S_{i+1}}), f(\vec{y}|_{S_{i+1}})) \equiv 0 .$$

Let  $A_0(\vec{y}|_{S_{i+1}}, x_{i+1})$  denote the polynomial

$$\tilde{q}(f(\vec{y}|_{S_1 \cap S_{i+1}}), f(\vec{y}|_{S_2 \cap S_{i+1}}), \dots, f(\vec{y}|_{S_i \cap S_{i+1}}), x_{i+1}) .$$

The above two conditions imply that  $f(\vec{y}|_{S_{i+1}})$  is a root of the polynomial  $A_0(\vec{y}|_{S_{i+1}}, x_{i+1}) \in \mathbb{F}[\vec{y}|_{S_{i+1}}][x_{i+1}]$ , viewed as a polynomial in  $x_{i+1}$ . Moreover,  $A_0(\vec{y}|_{S_{i+1}}, x_{i+1})$  has a circuit of size at most  $O(sn)$  and depth at most  $\Delta + 2$ . This follows from the fact that  $f(\vec{y}|_{S_1 \cap S_{i+1}})$  is a *multilinear* polynomial in  $\log n$  variables, and can thus be computed by a  $\sum \prod$  circuit of size at most  $n$ . We simply replace the variables  $x_1, x_2, \dots, x_i$  in the circuit for  $q$  by these  $\sum \prod$  circuits to obtain a circuit for  $A_0$ . The degree of  $A_0$  is at most  $D \log n$ . Finally, Theorem 5 implies that  $f(\vec{y}|_{S_{i+1}})$  can be computed by a circuit of size at most  $O(\text{poly}(s, n, D)d^{O(\sqrt{d})})$  and depth at most  $\Delta + 5$ , thus completing the proof.  $\blacktriangleleft$

We now sketch the proof of Theorem 3.

**Proof Sketch.** Once again, the proof follows the proof of Theorems 1 and 2 in [5]. Let  $\{f_m\}$  be a family of explicit multilinear polynomials such that  $f_m$  has  $m$  variables, degree  $d \leq O\left(\left(\frac{\log m}{\log \log m}\right)^2\right)$ , such that  $f_m$  cannot be computed by a circuit of depth  $\Delta$  and size  $\text{poly}(m)$ . Let  $\varepsilon \in (0, 0.49)$  be an arbitrary constant, and set  $m := n^\varepsilon$ , and  $f = f_m$ .

Given as input a circuit  $C \in \mathbb{F}[\vec{x}]$  of size  $s$ , depth  $\Delta - 5$  and degree  $D$  on  $n$  variables, let  $q \in \mathbb{F}[\vec{x}]$  be the polynomial computed by  $C$ . The goal here is to determine whether  $q$  is nonzero. From the equivalence of black-box PIT and hitting set, it suffices to construct hitting set for circuit class of the above properties.

- We construct a design  $S_1, S_2, \dots, S_n \subseteq [\ell]$  using Theorem 14 where each set  $S_i$  has size  $m$ ,  $\ell = O(m^2 / \log n) \leq n^{2\varepsilon} < n^{0.98}$  and  $|S_i \cap S_j| \leq \log n$ . This can be done in deterministic time  $2^{O(n^{2\varepsilon})}$ .
- We pick a subset  $T$  of the field  $\mathbb{F}$  of size  $Dd + 1$  and evaluate the polynomial  $q(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \dots, f(\vec{y}|_{S_n}))$  on all points of  $T^\ell$ .  $H = \{(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \dots, f(\vec{y}|_{S_n})) \mid \vec{y} \in T^\ell\}$  is then our candidate hitting set of size  $(Dd + 1)^\ell = n^{O(n^{2\varepsilon})} < n^{O(n^{0.98})}$ . Note that the set can be constructed deterministically in time  $m^d \cdot n^{O(n^{2\varepsilon})} = n^{O(n^{2\varepsilon})}$ .

We now argue about the correctness, *i.e.*,  $q$  does not vanish on the hitting set if and only if  $q$  is not identically zero. Observe that if the polynomial  $q(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \dots, f(\vec{y}|_{S_n}))$  is not identically zero, then it has degree at most  $Dd$  and hence by Lemma 15,  $q$  does not vanish on the set  $H$ . Else,  $q(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \dots, f(\vec{y}|_{S_n})) \equiv 0$ . But then, by Lemma 20, we get that  $f$  can be computed by a circuit of depth  $\Delta$  and size at most  $O(\text{poly}(s, n, D)d^{O(\sqrt{d})})$ .

If  $s, D$  are  $\text{poly}(n)$ , then this bound is  $\text{poly}(m)$  which contradicts the assumed hardness of  $f = f_m$  for circuits of depth  $\Delta$ . This shows that  $H$  is a hitting set for the desired circuit class and completes the proof. ◀

---

### References

- 1 Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 67–75. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.32.
- 2 Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983. doi:10.1016/0304-3975(83)90110-X.
- 3 Peter Bürgisser. The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics*, 4(4):369–396, 2004. doi:10.1007/s10208-002-0059-5.
- 4 Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. *CoRR*, abs/1710.03214, 2017. URL: <http://arxiv.org/abs/1710.03214>.
- 5 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.
- 6 Michael A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 451–465. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.35.
- 7 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 128–135. ACM, 2014. doi:10.1145/2591796.2591824.
- 8 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Satharishi. Arithmetic circuits: A chasm at depth three. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 578–587. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.68.
- 9 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Satharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, 2014. doi:10.1145/2629541.
- 10 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.
- 11 K. Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985. doi:10.1137/0214050.
- 12 Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989.
- 13 Neeraj Kayal, Chandan Saha, and Ramprasad Satharishi. A super-polynomial lower bound for regular arithmetic formulas. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 146–153. ACM, 2014. doi:10.1145/2591796.2591847.
- 14 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. doi:10.1016/j.tcs.2012.03.041.
- 15 Mrinal Kumar. A quadratic lower bound for homogeneous algebraic branching programs. In Ryan O’Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPICs*, pages 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CCC.2017.19.

- 16 Mrinal Kumar and Ramprasad Saptharishi. An exponential lower bound for homogeneous depth-5 circuits over finite fields. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPICs*, pages 31:1–31:30. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CCC.2017.31.
- 17 Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 364–373. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.46.
- 18 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.
- 19 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. doi:10.1007/BF01294256.
- 20 Rafael Oliveira. Factors of low individual degree polynomials. *Computational Complexity*, 25(2):507–561, 2016. doi:10.1007/s00037-016-0130-2.
- 21 Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. doi:10.4086/toc.2006.v002a006.
- 22 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010. doi:10.4086/toc.2010.v006a007.
- 23 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013. doi:10.1145/2535928.
- 24 Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. Comput.*, 38(4):1624–1647, 2008. doi:10.1137/070707932.
- 25 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009. doi:10.1007/s00037-009-0270-8.
- 26 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, <https://github.com/dasarpmar/lowerbounds-survey/>, 2016. URL: <https://github.com/dasarpmar/lowerbounds-survey/releases/>.
- 27 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:10.1561/04000000039.
- 28 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. doi:10.1016/j.ic.2014.09.004.





# On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product

Lijie Chen<sup>1</sup>

Massachusetts Institute of Technology, USA

lijieche@mit.edu

---

## Abstract

In this paper we study the (Bichromatic) Maximum Inner Product Problem (Max-IP), in which we are given sets  $A$  and  $B$  of vectors, and the goal is to find  $a \in A$  and  $b \in B$  maximizing inner product  $a \cdot b$ . Max-IP is very basic and serves as the base problem in the recent breakthrough of [Abboud et al., FOCS 2017] on hardness of approximation for polynomial-time problems. It is also used (implicitly) in the argument for hardness of exact  $\ell_2$ -Furthest Pair (and other important problems in computational geometry) in poly-log-log dimensions in [Williams, SODA 2018]. We have three main results regarding this problem.

- **Characterization of Multiplicative Approximation.** First, we study the best multiplicative approximation ratio for Boolean Max-IP in sub-quadratic time. We show that, for Max-IP with two sets of  $n$  vectors from  $\{0, 1\}^d$ , there is an  $n^{2-\Omega(1)}$  time  $(d/\log n)^{\Omega(1)}$ -multiplicative-approximating algorithm, and we show this is conditionally optimal, as such a  $(d/\log n)^{o(1)}$ -approximating algorithm would refute SETH. Similar characterization is also achieved for additive approximation for Max-IP.
- **$2^{O(\log^* n)}$ -dimensional Hardness for Exact Max-IP Over The Integers.** Second, we revisit the hardness of solving Max-IP exactly for vectors with integer entries. We show that, under SETH, for Max-IP with sets of  $n$  vectors from  $\mathbb{Z}^d$  for some  $d = 2^{O(\log^* n)}$ , every exact algorithm requires  $n^{2-o(1)}$  time. With the reduction from [Williams, SODA 2018], it follows that  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair in  $2^{O(\log^* n)}$  dimensions require  $n^{2-o(1)}$  time.
- **Connection with NP·UPP Communication Protocols.** Last, We establish a connection between conditional lower bounds for exact Max-IP with integer entries and NP·UPP communication protocols for Set-Disjointness, parallel to the connection between conditional lower bounds for approximating Max-IP and MA communication protocols for Set-Disjointness.

The lower bound in our first result is a direct corollary of the new MA protocol for Set-Disjointness introduced in [Rubinfeld, STOC 2018], and our algorithms utilize the polynomial method and simple random sampling. Our second result follows from a new dimensionality self reduction from the Orthogonal Vectors problem for  $n$  vectors from  $\{0, 1\}^d$  to  $n$  vectors from  $\mathbb{Z}^\ell$  where  $\ell = 2^{O(\log^* d)}$ , dramatically improving the previous reduction in [Williams, SODA 2018]. The key technical ingredient is a recursive application of *Chinese Remainder Theorem*.

As a side product, we obtain an MA communication protocol for Set-Disjointness with complexity  $O(\sqrt{n \log n \log \log n})$ , slightly improving the  $O(\sqrt{n} \log n)$  bound [Aaronson and Wigderson, TOCT 2009], and approaching the  $\Omega(\sqrt{n})$  lower bound [Klauck, CCC 2003].

Moreover, we show that (under SETH) one can apply the  $O(\sqrt{n})$  BQP communication protocol for Set-Disjointness to prove near-optimal hardness for approximation to Max-IP with vectors in  $\{-1, 1\}^d$ . This answers a question from [Abboud et al., FOCS 2017] in the affirmative.

**2012 ACM Subject Classification** Theory of computation → Problems, reductions and completeness

---

<sup>1</sup> Supported by an Akamai Fellowship



**Keywords and phrases** Maximum Inner Product, SETH, Hardness of Approximation in P, Fined-Grained Complexity, Hopcroft’s Problem, Chinese Remainder Theorem

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.14

**Acknowledgements** I would like to thank Ryan Williams for introducing the problem to me, countless encouragement and helpful discussions during this work, and also many comments on a draft of this paper. In particular, the idea of improving OV dimensionality self-reduction using CRT (the direct CRT based approach) is introduced to me by Ryan Williams.

I am grateful to Virginia Vassilevska Williams, Kaifeng Lyu, Peilin Zhong for helpful discussions and suggestions. I would like to thank Aviad Rubinfeld for sharing a manuscript of his paper, and pointing out that the  $O(\sqrt{n \log n \log \log n})$  MA protocol also works for Inner Product.

## 1 Introduction

We study the following fundamental problem from similarity search and statistics, which asks to find the most correlated pair in a dataset:

► **Definition 1.1** (Bichromatic Maximum Inner Product (Max-IP)). For  $n, d \in \mathbb{N}$ , the  $\text{Max-IP}_{n,d}$  problem is defined as: *given two sets  $A, B$  of vectors from  $\{0, 1\}^d$  compute*

$$\text{OPT}(A, B) := \max_{a \in A, b \in B} a \cdot b.$$

We use  $\mathbb{Z}\text{-Max-IP}_{n,d}$  ( $\mathbb{R}\text{-Max-IP}_{n,d}$ ) to denote the same problem, but with  $A, B$  being sets of vectors from  $\mathbb{Z}^d$  ( $\mathbb{R}^d$ ).

### Hardness of Approximation Max-IP

A natural brute-force algorithm solves Max-IP in  $O(n^2 \cdot d)$ -time. Assuming  $\text{SETH}^2$ , there is no  $n^{2-\Omega(1)}$ -time algorithm for  $\text{Max-IP}_{n,d}$  when  $d = \omega(\log n)$  [70].

Despite being one of the most central problems in similarity search and having numerous applications [47, 15, 61, 62, 65, 17, 16, 18, 57, 66, 68, 14, 49, 12, 67, 32, 31], until recently it was unclear whether there could be a near-linear time, 1.1-approximating algorithm, before the recent breakthrough of Abboud, Rubinfeld and Williams [5].<sup>3</sup>

In [5], a framework for proving inapproximability results for problems in P is established (the distributed PCP framework), from which it follows:

► **Theorem 1.2** ([5]). *Assuming SETH, there is no  $2^{(\log n)^{1-o(1)}}$ -multiplicative-approximating  $n^{2-\Omega(1)}$ -time algorithm for  $\text{Max-IP}_{n,n^{o(1)}}$ .*

Theorem 1.2 is an exciting breakthrough for hardness of approximation in P, implying other important inapproximability results for a host of problems including Bichromatic LCS Closest Pair Over Permutations, Approximate Regular Expression Matching, and Diameter in Product Metrics [5]. However, we still do not have a complete understanding of the approximation hardness of Max-IP yet. For instance, consider the following two concrete questions:

<sup>2</sup> SETH (Strong Exponential Time Hypothesis) states that for every  $\varepsilon > 0$  there is a  $k$  such that  $k$ -SAT cannot be solved in  $O((2 - \varepsilon)^n)$  time [46].

<sup>3</sup> see [5] for a thorough discussion on the state of affairs on hardness of approximation in P before their work

► **Question 1.** *Is there a  $(\log n)$ -multiplicative-approximating  $n^{2-\Omega(1)}$ -time algorithm for  $\text{Max-IP}_{n, \log^2 n}$ ? What about a 2-multiplicative-approximating algorithm for  $\text{Max-IP}_{n, \log^2 n}$ ?*

► **Question 2.** *Is there a  $(d/\log n)$ -additive-approximating  $n^{2-\Omega(1)}$ -time algorithm for  $\text{Max-IP}_{n,d}$ ?*

We note that the lower bound from [5] cannot answer Question 1. Tracing the details of their proofs, one can see that it only shows approximation hardness for dimension  $d = \log^{\omega(1)} n$ . Question 2 concerning additive approximation is not addressed at all by [5]. Given the importance of  $\text{Max-IP}$ , it is interesting to ask:

*For what ratios  $r$  do  $n^{2-\Omega(1)}$ -time  $r$ -approximation algorithms exist for  $\text{Max-IP}$ ?*

Does the best-possible approximation ratio (in  $n^{2-\Omega(1)}$  time) relate to the dimensionality, in some way?

In an important recent work, Rubinfeld [64] improved the distributed PCP construction in a very crucial way, from which one can derive more refined lower bounds on approximating  $\text{Max-IP}$ . Building on its technique, in this paper we provide full *characterizations*, determining essentially optimal multiplicative approximations and additive approximations to  $\text{Max-IP}$ , under  $\text{SETH}$ .

## Hardness of Exact $\mathbb{Z}$ -Max-IP

Recall that from [70], there is no  $n^{2-\Omega(1)}$ -time algorithm for exact Boolean  $\text{Max-IP}_{n, \omega(\log n)}$ . Since in real life applications of similarity search, one often deals with real-valued data instead of just Boolean data, it is natural to ask about  $\mathbb{Z}$ -Max-IP (which is certainly a special case of  $\mathbb{R}$ -Max-IP): what is the maximum  $d$  such that  $\mathbb{Z}$ -Max-IP $_{n,d}$  can be solved exactly in  $n^{2-\Omega(1)}$  time?

Besides being interesting in its own right, there are also reductions from  $\mathbb{Z}$ -Max-IP to  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair. Hence, lower bounds for  $\mathbb{Z}$ -Max-IP imply lower bounds for these two famous problems in computational geometry (see [72] for a discussion on this topic).

Prior to our work, it was implicitly shown in [72] that:

► **Theorem 1.3** ([72]). *There is no  $n^{2-\Omega(1)}$ -time algorithm for  $\mathbb{Z}$ -Max-IP $_{n, \omega((\log \log n)^2)}$  with vectors of  $O(\log n)$ -bit entries, assuming  $\text{SETH}$ .*

However, the best known algorithm for  $\mathbb{Z}$ -Max-IP runs in  $n^{2-\Theta(1/d)}$  time [55, 11, 74]<sup>4</sup>, hence there is still a gap between the lower bound and the best known upper bounds. To confirm these algorithms are in fact optimal, we would like to prove a lower bound with  $\omega(1)$  dimensions.

In this paper, we significantly strength the previous lower bound from  $\omega((\log \log n)^2)$  dimensions to  $2^{O(\log^* n)}$  dimensions ( $2^{O(\log^* n)}$  is an *extremely slow-growing* function, see preliminaries for its formal definition).

<sup>4</sup> [11, 74] are for  $\ell_2$ -Furthest Pair or Bichromatic  $\ell_2$ -Closest Pair. They also work for  $\mathbb{Z}$ -Max-IP as there are reductions from  $\mathbb{Z}$ -Max-IP to these two problems, see [72] or Lemma 4.5 and Lemma 4.6.

## Fine-Grained Complexity and Communication Complexity

One intriguing aspect of the distributed PCP framework is that it makes use of the  $\tilde{O}(\sqrt{n})$  MA communication protocol for Set-Disjointness [1]. Several follow-up works [50, 64] explored this connection further, and settled the hardness of approximation to several fundamental problems (under SETH).

Given the success of the interplay between these two seemingly unrelated fields, it is natural to seek more results from it. In particular, it is asked in [5] whether the  $O(\sqrt{n})$  BQP communication protocol for Set-Disjointness can be utilized.

In this paper, we answer the question affirmatively by showing that BQP communication protocol implies hardness for approximation to  $\{-1, 1\}$ -Max-IP<sup>5</sup>. Moreover, we also establish a connection between  $\mathbb{Z}$ -Max-IP lower bounds and NP · UPP communication protocols for Set-Disjointness, which suggests a new perspective on our results on  $\mathbb{Z}$ -Max-IP.

### 1.1 Our Results

We use  $\text{OV}_{n,d}$  to denote the Orthogonal Vectors problem: given two sets of vectors  $A, B$  each consisting of  $n$  vectors from  $\{0, 1\}^d$ , determine whether there are  $a \in A$  and  $b \in B$  such that  $a \cdot b = 0$ .<sup>6</sup> Similarly, we use  $\mathbb{Z}\text{-OV}_{n,d}$  to denote the same problem except for that  $A, B$  consists of vectors from  $\mathbb{Z}^d$  (which is also called Hopcroft's problem).

All our results are based on the following widely used conjecture about OV:

► **Conjecture 1.4** (Orthogonal Vectors Conjecture (OVC) [70, 7]). *For every  $\varepsilon > 0$ , there exists a  $c \geq 1$  such that  $\text{OV}_{n,d}$  requires  $n^{2-\varepsilon}$  time when  $d = c \log n$ .*

OVC is a plausible conjecture as it is implied by the popular Strong Exponential Time Hypothesis [46, 29] on the time complexity of solving  $k$ -SAT [70, 73].

## Characterizations of Hardness of Approximate Max-IP

The first main result of our paper characterizes when there is a truly sub-quadratic time ( $n^{2-\Omega(1)}$ ) time, for some universal constant hidden in the big- $\Omega$ )  $t$ -multiplicative-approximating algorithm for Max-IP, and characterizes the best-possible additive approximations as well. We begin with formal definitions of these two standard types of approximation:

- We say an algorithm  $\mathbb{A}$  for Max-IP $_{n,d}$  ( $\mathbb{Z}$ -Max-IP $_{n,d}$ ) is  $t$ -multiplicative-approximating, if for all  $A, B$ ,  $\mathbb{A}$  outputs a value  $\widetilde{\text{OPT}}(A, B) \in [\text{OPT}(A, B), \text{OPT}(A, B) \cdot t]$ .
- We say an algorithm  $\mathbb{A}$  for Max-IP $_{n,d}$  ( $\mathbb{Z}$ -Max-IP $_{n,d}$ ) is  $t$ -additive-approximating, if for all  $A, B$ ,  $\mathbb{A}$  outputs a value  $\widetilde{\text{OPT}}(A, B)$  such that  $|\widetilde{\text{OPT}}(A, B) - \text{OPT}(A, B)| \leq t$ .
- To avoid ambiguity, we call an algorithm computing  $\text{OPT}(A, B)$  exactly an *exact* algorithm for Max-IP $_{n,d}$  ( $\mathbb{Z}$ -Max-IP $_{n,d}$ ).

**Multiplicative Approximations for Max-IP.** In the multiplicative case, our characterization (formally stated below) basically says that there is a  $t$ -multiplicative-approximating  $n^{2-\Omega(1)}$ -time algorithm for Max-IP $_{n,d}$  if and only if  $t = (d/\log n)^{\Omega(1)}$ . Note that in the following theorem we require  $d = \omega(\log n)$ , since in the case of  $d = O(\log n)$ , there are  $n^{2-\varepsilon}$ -time algorithms for exact Max-IP $_{n,d}$  [14, 13].

<sup>5</sup> That is, Max-IP with sets  $A$  and  $B$  being  $n$  vectors from  $\{-1, 1\}^d$ .

<sup>6</sup> Here we use the bichromatic version of OV instead of the monochromatic one for convenience, as they are equivalent.

► **Theorem 1.5.** Letting  $\omega(\log n) < d < n^{o(1)}$  and  $t \geq 2$ ,<sup>7</sup> the following holds:

1. There is an  $n^{2-\Omega(1)}$ -time  $t$ -multiplicative-approximating algorithm for  $\text{Max-IP}_{n,d}$  if

$$t = (d/\log n)^{\Omega(1)},$$

and under *SETH* (or *OVC*), there is no  $n^{2-\Omega(1)}$ -time  $t$ -multiplicative-approximating algorithm for  $\text{Max-IP}_{n,d}$  if

$$t = (d/\log n)^{o(1)}.$$

2. Moreover, let  $\varepsilon = \min\left(\frac{\log t}{\log(d/\log n)}, 1\right)$ . There are  $t$ -multiplicative-approximating deterministic algorithms for  $\text{Max-IP}_{n,d}$  running in time

$$O\left(n^{2+o(1)-0.31 \cdot \frac{1}{\varepsilon-1+\frac{0.31}{2}}}\right) = O\left(n^{2+o(1)-\Omega(\varepsilon)}\right)$$

or time

$$O\left(n^{2-0.17 \cdot \frac{1}{\varepsilon-1+\frac{0.17}{2}}} \cdot \text{polylog}(n)\right) = O\left(n^{2-\Omega(\varepsilon)} \cdot \text{polylog}(n)\right).$$

► **Remark 1.6.** The first algorithm is slightly faster, but only sub-quadratic when  $\varepsilon = \Omega(1)$ , while the second algorithm still gets a non-trivial speed up over the brute force algorithm as long as  $\varepsilon = \omega(\log \log n / \log n)$ .

We remark here that the above algorithms indeed work for the case where the sets consisting of non-negative reals (i.e.,  $\mathbb{R}^+$ -Max-IP):

► **Corollary 1.7.** Assume  $\omega(\log n) < d < n^{o(1)}$  and let  $\varepsilon = \min\left(\frac{\log t}{\log(d/\log n)}, 1\right)$ . There is a  $t$ -multiplicative-approximating deterministic algorithm for  $\mathbb{R}^+$ -Max-IP $_{n,d}$  running in time

$$O\left(n^{2-\Omega(\varepsilon)} \cdot \text{polylog}(n)\right).$$

The lower bound is a direct corollary of the new improved MA protocols for Set-Disjointness from [64], which is based on Algebraic Geometry codes. Together with the framework of [5], that MA-protocol implies a reduction from OV to approximating Max-IP.

Our upper bounds are application of the polynomial method [71, 9]: defining appropriate sparse polynomials for approximating Max-IP on small groups of vectors, and use fast matrix multiplication to speed up the evaluation of these polynomials on many pairs of points.

Via the known reduction from Max-IP to LCS-Pair in [5], we also obtain a more refined lower bound for approximating the LCS Closest Pair problem (defined below).

► **Definition 1.8** (LCS Closest Pair). The LCS-Closest-Pair $_{n,d}$  problem is: given two sets  $A, B$  of  $n$  strings from  $\Sigma^d$  ( $\Sigma$  is a finite alphabet), determine

$$\max_{a \in A, b \in B} \text{LCS}(a, b),$$

where  $\text{LCS}(a, b)$  is the length of the longest common subsequence of strings  $a$  and  $b$ .

► **Corollary 1.9** (Improved Inapproximability for LCS-Closest-Pair). Assuming *SETH* (or *OVC*), for every  $t \geq 2$ ,  $t$ -multiplicative-approximating LCS-Closest-Pair $_{n,d}$  requires  $n^{2-o(1)}$  time, if  $d = t^{\omega(1)} \cdot \log^5 n$ .

<sup>7</sup> Note that  $t$  and  $d$  are both functions of  $n$ , we assume they are computable in  $n^{o(1)}$  time throughout this paper for simplicity.

**Additive Approximations for Max-IP.** Our characterization for additive approximations to Max-IP says that there is a  $t$ -additive-approximating  $n^{2-\Omega(1)}$ -time algorithm for  $\text{Max-IP}_{n,d}$  if and only if  $t = \Omega(d)$ .

► **Theorem 1.10.** *Letting  $\omega(\log n) < d < n^{o(1)}$  and  $0 \leq t \leq d$ , the following holds:*

1. *There is an  $n^{2-\Omega(1)}$ -time  $t$ -additive-approximating algorithm for  $\text{Max-IP}_{n,d}$  if*

$$t = \Omega(d),$$

*and under SETH (or OVC), there is no  $n^{2-\Omega(1)}$ -time  $t$ -additive-approximating algorithm for  $\text{Max-IP}_{n,d}$  if*

$$t = o(d).$$

2. *Moreover, letting  $\varepsilon = \frac{t}{d}$ , there is a randomized*

$$O\left(n^{2-\Omega(\varepsilon^{1/3}/\log \varepsilon^{-1})}\right)$$

*time,  $t$ -additive-approximating algorithm for  $\text{Max-IP}_{n,d}$  when  $\varepsilon \gg \log^6 \log n / \log^3 n$ .*

The lower bound above is already established in [64], while the upper bound works by reducing the problem to the  $d = O(\log n)$  case via random-sampling coordinates, and solving the reduced problem via known methods [14, 13].

► **Remark 1.11.** We want to remark here that the lower bounds for approximating Max-IP are direct corollaries of the new MA protocols for Set-Disjointness in [64]. Our main contribution is providing the complementary *upper bounds* to show that these lower bounds are indeed *tight* assuming SETH.

**All-Pair-Max-IP.** Finally, we remark that our algorithms (with slight adaptations) also work for the following stronger problem<sup>8</sup>: All-Pair-Max-IP $_{n,d}$ , in which we are given two sets  $A$  and  $B$  of  $n$  vectors from  $\{0, 1\}^d$ , and for each  $x \in A$  we must compute  $\text{OPT}(x, B) := \max_{y \in B} x \cdot y$ . An algorithm is  $t$ -multiplicative-approximating (additive-approximating) for All-Pair-Max-IP if for all  $\text{OPT}(x, B)$ 's, it computes corresponding approximating answers.

► **Corollary 1.12.** *Suppose  $\omega(\log n) < d < n^{o(1)}$ , and let*

$$\varepsilon_M := \min\left(\frac{\log t}{\log(d/\log n)}, 1\right) \text{ and } \varepsilon_A := \frac{\min(t, d)}{d}.$$

*There is an  $n^{2-\Omega(\varepsilon_M)}$  polylog( $n$ ) time  $t$ -multiplicative-approximating algorithm and an  $n^{2-\Omega(\varepsilon_A^{1/3}/\log \varepsilon_A^{-1})}$  time  $t$ -additive-approximating algorithm for All-Pair-Max-IP $_{n,d}$ , when  $\varepsilon_A \gg \log^6 \log n / \log^3 n$ .*

<sup>8</sup> Since All-Pair-Max-IP is stronger than Max-IP, lower bounds for Max-IP automatically apply for All-Pair-Max-IP.

## BQP Communication Protocols and Approximate $\{-1,1\}$ -Max-IP.

Making use of the  $O(\sqrt{n})$ -degree approximate polynomial for OR [27, 36], we also give a completely different proof for the hardness of multiplicative approximation to  $\{-1, 1\}$ -Max-IP. Lower bound from that approach is inferior to Theorem 1.5: in particular, *it cannot achieve a characterization.*

It is asked in [5] that whether we can make use of the  $O(\sqrt{n})$  BQP communication protocol for Set-Disjointness [28] to prove conditional lower bounds. Indeed, that quantum communication protocol is based on the  $O(\sqrt{n})$ -time quantum query algorithm for OR (Grover's algorithm [42]), which induces the needed approximate polynomial for OR. Hence, the following theorem in some sense answers their question in the affirmative:

► **Theorem 1.13 (Informal).** *Assuming SETH (or OVC), there is no  $n^{2-\Omega(1)}$  time  $n^{o(1)}$ -multiplicative-approximating algorithm for  $\{-1, 1\}$ -Max-IP $_{n, n^{o(1)}}$ .*

The full statement can be found in Theorem C.1 and Theorem C.2.

## Hardness of Exact $\mathbb{Z}$ -Max-IP in $2^{O(\log^* n)}$ Dimensions

Now we turn to discuss our results on  $\mathbb{Z}$ -Max-IP. We show that  $\mathbb{Z}$ -Max-IP is hard to solve in  $n^{2-\Omega(1)}$  time, even with  $2^{O(\log^* n)}$ -dimensional vectors:

► **Theorem 1.14.** *Assuming SETH (or OVC), there is a constant  $c$  such that any exact algorithm for  $\mathbb{Z}$ -Max-IP $_{n,d}$  for  $d = c^{\log^* n}$  dimensions requires  $n^{2-o(1)}$  time, with vectors of  $O(\log n)$ -bit entries.*

As direct corollaries of the above theorem, using reductions implicit in [72], we also conclude hardness for  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair under SETH (or OVC) in  $2^{O(\log^* n)}$  dimensions.

► **Theorem 1.15 (Hardness of  $\ell_2$ -Furthest Pair in  $c^{\log^* n}$  Dimensions).** *Assuming SETH (or OVC), there is a constant  $c$  such that  $\ell_2$ -Furthest Pair in  $c^{\log^* n}$  dimensions requires  $n^{2-o(1)}$  time, with vectors of  $O(\log n)$ -bit entries.*

► **Theorem 1.16 (Hardness of Bichromatic  $\ell_2$ -Closest Pair in  $c^{\log^* n}$  Dimensions).** *Assuming SETH (or OVC), there is a constant  $c$  such that Bichromatic  $\ell_2$ -Closest Pair in  $c^{\log^* n}$  dimensions requires  $n^{2-o(1)}$  time, with vectors of  $O(\log n)$ -bit entries.*

The above lower bounds on  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair are in sharp contrast with the case of  $\ell_2$ -Closest Pair, which can be solved in  $2^{O(d)} \cdot n \log^{O(1)} n$  time [23, 51, 37].

## Improved Dimensionality Reduction for OV and Hopcroft's Problem

Our hardness of  $\mathbb{Z}$ -Max-IP is established by a reduction from Hopcroft's problem, whose hardness is in turn derived from the following significantly improved dimensionality reduction for OV.

► **Lemma 1.17 (Improved Dimensionality Reduction for OV).** *Let  $1 \leq \ell \leq d$ . There is an*

$$O\left(n \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))} \cdot \text{poly}(d)\right) \text{-time}$$

*reduction from  $OV_{n,d}$  to  $\ell^{O(6^{\log^* d} \cdot (d/\ell))}$  instances of  $\mathbb{Z}$ -OV $_{n,\ell+1}$ , with vectors of entries with bit-length  $O\left(d/\ell \cdot \log \ell \cdot 6^{\log^* d}\right)$ .*



**Comparison with [72].** Comparing to the old construction in [72], our reduction here is more efficient when  $\ell$  is much smaller than  $d$  (which is the case we care about). That is, in [72],  $\text{OV}_{n,d}$  can be reduced to  $d^{d/\ell}$  instances of  $\mathbb{Z}\text{-OV}_{n,\ell+1}$ , while we get  $\left\{\ell^{6^{\log^* d}}\right\}^{d/\ell}$  instances in our improved one. So, for example, when  $\ell = 7^{\log^* d}$ , the old reduction yields  $d^{d/7^{\log^* d}} = n^{\omega(1)}$  instances (recall that  $d = c \log n$  for an arbitrary constant  $c$ ), while our improved one yields only  $n^{o(1)}$  instances, each with  $2^{O(\log^* n)}$  dimensions.

From Lemma 1.17, the following theorem follows in the same way as in [72].

► **Theorem 1.18** (Hardness of Hopcroft's Problem in  $c^{\log^* n}$  Dimensions). *Assuming SETH (or OVC), there is a constant  $c$  such that  $\mathbb{Z}\text{-OV}_{n,c^{\log^* n}}$  with vectors of  $O(\log n)$ -bit entries requires  $n^{2-o(1)}$  time.*

### Connection between $\mathbb{Z}$ -Max-IP Lower Bounds and NP · UPP Communication Protocols

We also show a new connection between  $\mathbb{Z}$ -Max-IP and a special type of communication protocol. Let us first recall the Set-Disjointness problem:

► **Definition 1.19** (Set-Disjointness). Let  $n \in \mathbb{N}$ , in Set-Disjointness ( $\text{DISJ}_n$ ), Alice holds a vector  $X \in \{0,1\}^n$ , Bob holds a vector  $Y \in \{0,1\}^n$ , and they want to determine whether  $X \cdot Y = 0$ .

In [5], the hardness of approximating Max-IP is established via a connection to MA communication protocols (in particular, a fast MA communication protocol for Set-Disjointness). Our lower bound for (exact)  $\mathbb{Z}$ -Max-IP can also be connected to similar NP · UPP protocols (note that  $\text{MA} = \text{NP} \cdot \text{promiseBPP}$ ).

Formally, we define NP · UPP protocols as follows:

► **Definition 1.20.** For a problem  $\Pi$  with inputs  $x, y$  of length  $n$  (Alice holds  $x$  and Bob holds  $y$ ), we say a communication protocol is an  $(m, \ell)$ -efficient NP · UPP communication protocol if the following holds:

- There are three parties Alice, Bob and Merlin in the protocol.
- Merlin sends Alice and Bob an advice string  $z$  of length  $m$ , which is a function of  $x$  and  $y$ .
- Given  $y$  and  $z$ , Bob sends Alice  $\ell$  bits, and Alice decides to accept or not.<sup>9</sup> They have an unlimited supply of private random coins (not public, which is important) during their conversation. The following conditions hold:
  - If  $\Pi(x, y) = 1$ , then there is an advice  $z$  from Merlin such that Alice accepts with probability  $\geq 1/2$ .
  - Otherwise, for all possible advice strings from Merlin, Alice accepts with probability  $< 1/2$ .

Moreover, we say the protocol is  $(m, \ell)$ -computational-efficient, if in addition the probability distributions of both Alice and Bob's behavior can be computed in  $\text{poly}(n)$  time given their input and the advice.

Our new reduction from OV to Max-IP actually implies a super-efficient NP · UPP protocol for Set-Disjointness.

---

<sup>9</sup> In UPP, actually one-way communication is equivalent to the seemingly more powerful one in which they communicate [60].



► **Theorem 1.21.** *For all  $1 \leq \alpha \leq n$ , there is an*

$$\left(\alpha \cdot 6^{\log^* n} \cdot (n/2^\alpha), O(\alpha)\right)\text{-computational-efficient}$$

NP · UPP communication protocol for  $\text{DISJ}_n$ .

For example, when  $\alpha = 3 \log^* n$ , Theorem 1.21 implies there is an  $O(o(n), O(\log^* n))$ -computational-efficient NP · UPP communication protocol for  $\text{DISJ}_n$ . Moreover, we show that if the protocol of Theorem 1.21 can be improved a little bit (like removing the  $6^{\log^* n}$  term), we would obtain the desired hardness for  $\mathbb{Z}$ -Max-IP in  $\omega(1)$ -dimensions.

► **Theorem 1.22.** *Assuming SETH (or OVC), if there is an increasing and unbounded function  $f$  such that for all  $1 \leq \alpha \leq n$ , there is an*

$$(n/f(\alpha), \alpha)\text{-computational-efficient}$$

NP · UPP communication protocol for  $\text{DISJ}_n$ , then  $\mathbb{Z}$ -Max-IP $_{n, \omega(1)}$  requires  $n^{2-o(1)}$  time with vectors of  $\text{polylog}(n)$ -bit entries. The same holds for  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair.

## Improved MA Protocols for Set-Disjointness

Finally, we also obtain a new MA protocol for Set-Disjointness, which improves on the previous  $O(\sqrt{n} \log n)$  protocol in [1], and is closer to the  $\Omega(\sqrt{n})$  lower bound by [52]. Like the protocol in [1], our new protocol also works for the following slightly harder problem Inner Product.

► **Definition 1.23** (Inner Product). Let  $n \in \mathbb{N}$ , in Inner Product ( $\text{IP}_n$ ), Alice holds a vector  $X \in \{0, 1\}^n$ , Bob holds a vector  $Y \in \{0, 1\}^n$ , and they want to compute  $X \cdot Y$ .

► **Theorem 1.24.** *There is an MA protocol for  $\text{DISJ}_n$  and  $\text{IP}_n$  with communication complexity*

$$O\left(\sqrt{n \log n \log \log n}\right).$$

In [64], the author asked whether the MA communication complexity of DISJ (IP) is  $\Theta(\sqrt{n})$  or  $\Theta(\sqrt{n} \log n)$ , and suggested that  $\Omega(n \log n)$  may be necessary for IP. Our result makes progress on that question by showing that the true complexity lies between  $\Theta(\sqrt{n})$  and  $\Theta(\sqrt{n \log n \log \log n})$ .

## 1.2 Intuition for Dimensionality Self Reduction for OV

The  $2^{O(\log^* n)}$  factor in Lemma 1.17 is not common in theoretical computer science<sup>10</sup>, and our new reduction for OV is considerably more complicated than the polynomial-based construction from [72]. Hence, it is worth discussing the intuition behind Lemma 1.17, and the reason why we get a factor of  $2^{O(\log^* n)}$ .

<sup>10</sup>Other examples include an  $O(2^{O(\log^* n)} n^{4/3})$  time algorithm for  $\mathbb{Z}$ -OV $_{n,3}$  [56],  $O(2^{O(\log^* n)} n \log n)$  time algorithms (Fürer's algorithm with its modifications) for Fast Integer Multiplication [38, 34, 43] and an old  $O(n^{d/2} 2^{O(\log^* n)})$  time algorithm for Klee's measure problem [30].

**A Direct Chinese Remainder Theorem Based Approach.** We first discuss a direct reduction based on the *Chinese Remainder Theorem* (CRT) (see Theorem 2.5 for a formal definition). CRT says that given a collection of primes  $q_1, \dots, q_b$ , and a collection of integers  $r_1, \dots, r_b$ , there exists a unique integer  $t = \text{CRR}(\{r_i\}; \{q_i\})$  such that  $t \equiv r_i \pmod{q_i}$  for each  $i \in [b]$  (CRR stands for *Chinese Remainder Representation*).

Now, let  $b, \ell \in \mathbb{N}$ , suppose we would like to have a dimensionality reduction  $\varphi$  from  $\{0, 1\}^{b \cdot \ell}$  to  $\mathbb{Z}^\ell$ . We can partition an input  $x \in \{0, 1\}^{b \cdot \ell}$  into  $\ell$  blocks, each of length  $b$ , and represent each block via CRT: that is, for a block  $z \in \{0, 1\}^b$ , we map it into a single integer  $\varphi_{\text{block}}(z) := \text{CRR}(\{z_i\}; \{q_i\})$ , and the concatenations of  $\varphi_{\text{block}}$  over all blocks of  $x$  is  $\varphi(x) \in \mathbb{Z}^\ell$ .

The key idea here is that, for  $z, z' \in \{0, 1\}^b$ ,  $\varphi_{\text{block}}(z) \cdot \varphi_{\text{block}}(z') \pmod{q_i}$  is simply  $z_i \cdot z'_i$ . That is, the multiplication between two integers  $\varphi_{\text{block}}(z) \cdot \varphi_{\text{block}}(z')$  simulates the coordinate-wise multiplication between two vectors  $z$  and  $z'$ !

Therefore, if we make all primes  $q_i$  larger than  $\ell$ , we can in fact determine  $x \cdot y$  from  $\varphi(x) \cdot \varphi(y)$ , by looking at  $\varphi(x) \cdot \varphi(y) \pmod{q_i}$  for each  $i$ . That is,

$$x \cdot y = 0 \Leftrightarrow \varphi(x) \cdot \varphi(y) \equiv 0 \pmod{q_i} \text{ for all } i.$$

Hence, let  $V$  be the set of all integer  $0 \leq v \leq \ell \cdot \left(\prod_{i=1}^b q_i\right)^2$  that  $v \equiv 0 \pmod{q_i}$  for all  $i \in [b]$ , we have

$$x \cdot y = 0 \Leftrightarrow \varphi(x) \cdot \varphi(y) \in V.$$

The reduction is completed by enumerating all integers  $v \in V$ , and appending corresponding values to make  $\varphi_A(x) = [\varphi(x), -1]$  and  $\varphi_B(y) = [\varphi(y), v]$  (this step is from [72]).

Note that a nice property for  $\varphi$  is that each  $\varphi(x)_i$  only depends on the  $i$ -th block of  $x$ , and the mapping is the same on each block ( $\varphi_{\text{block}}$ ); we call this the *block mapping property*.

**Analysis of the Direct Reduction.** To continue building intuition, let us analyze the above reduction. The size of  $V$  is the number of  $\mathbb{Z}$ -OV $_{n, \ell+1}$  instances we create, and  $|V| \geq \prod_{i=1}^b q_i$ .

These primes  $q_i$  have to be all distinct, and it follows that  $\prod_{i=1}^b q_i$  is  $b^{\Theta(b)}$ . Since we want to create at most  $n^{o(1)}$  instances (or  $n^\varepsilon$  for arbitrarily small  $\varepsilon$ ), we need to set  $b \leq \log n / \log \log n$ . Moreover, to base our hardness on OVC which deals with  $c \log n$ -dimensional vectors, we need to set  $b \cdot \ell = d = c \cdot \log n$  for an arbitrary constant  $c$ . Therefore, we must have  $\ell \geq \log \log n$ , and the above reduction only obtains the same hardness result as [72].

**Key Observation: “Most Space Modulo  $q_i$ ” is Actually Wasted.** To improve the above reduction, we need to make  $|V|$  smaller. Our key observation about  $\varphi$  is that, for the primes  $q_i$ 's, they are mostly larger than  $b \gg \ell$ , but  $\varphi(x) \cdot \varphi(y) \in \{0, 1, \dots, \ell\} \pmod{q_i}$  for all these  $q_i$ 's. Hence, “most space modulo  $q_i$ ” is actually wasted.

**Make More “Efficient” Use of the “Space”: Recursive Reduction.** Based on the previous observation, we want to use the “space modulo  $q_i$ ” more efficiently. It is natural to consider a *recursive reduction*. We will require all our primes  $q_i$ 's to be larger than  $b$ . Let  $b_{\text{micro}}$  be a very small integer compared to  $b$ , and let  $\psi : \{0, 1\}^{b_{\text{micro}} \cdot \ell} \rightarrow \mathbb{Z}^\ell$  with a set  $V_\psi$  and a block mapping  $\psi_{\text{block}}$  be a similar reduction on a much smaller input: for  $x, y \in \{0, 1\}^{b_{\text{micro}} \cdot \ell}$ ,  $x \cdot y = 0 \Leftrightarrow \psi(x) \cdot \psi(y) \in V_\psi$ . We also require here that  $\psi(x) \cdot \psi(y) \leq b$  for all  $x$  and  $y$ .

For an input  $x \in \{0, 1\}^{b \cdot \ell}$  and a block  $z \in \{0, 1\}^b$  of  $x$ , our key idea is to partition  $z$  again into  $b/b_{\text{micro}}$  “micro” blocks each of size  $b_{\text{micro}}$ . And for a block  $z$  in  $x$ , let  $z^1, \dots, z^{b/b_{\text{micro}}}$  be its  $b/b_{\text{micro}}$  micro blocks, we map  $z$  into an integer  $\varphi_{\text{block}}(z) := \text{CRR}(\{\psi_{\text{block}}(z_i)\}_{i=1}^{b/b_{\text{micro}}}; \{q_i\}_{i=1}^{b/b_{\text{micro}}})$ .

Now, given two blocks  $z, z' \in \{0, 1\}^b$ , we can see that

$$\varphi_{\text{block}}(z) \cdot \varphi_{\text{block}}(z') \equiv \psi_{\text{block}}(z) \cdot \psi_{\text{block}}(z') \pmod{q_i}.$$

That is,  $\varphi(x) \cdot \varphi(y) \pmod{q_i}$  in fact is equal to  $\psi(x^{[i]}) \cdot \psi(y^{[i]})$ , where  $x^{[i]}$  is the concatenation of the  $i$ -th micro blocks of  $x$  in each block, and  $y^{[i]}$  is defined similarly. Hence, we can determine whether  $x^{[i]} \cdot y^{[i]} = 0$  from  $\varphi(x) \cdot \varphi(y) \pmod{q_i}$  for all  $i$ , and therefore also determine whether  $x \cdot y = 0$  from  $\varphi(x) \cdot \varphi(y)$ .

We can now observe that  $|V| \leq b^{\Theta(b/b_{\text{micro}})}$ , smaller than before; thus we get an improvement, depending on how large can  $b_{\text{micro}}$  be. Clearly, the reduction  $\psi$  can also be constructed from even smaller reductions, and after recursing  $\Theta(\log^* n)$  times, we can switch to the direct construction discussed before. By a straightforward (but tedious) calculation, we can derive Lemma 1.17.

**High-Level Explanation on the  $2^{O(\log^* n)}$  Factor.** Ideally, we want to have a reduction from OV to  $\mathbb{Z}$ -OV with only  $\ell^{O(b)}$  instances, in other words, we want  $|V| = \ell^{O(b)}$ . The reason we need to pay an extra  $2^{O(\log^* n)}$  factor in the exponent is as follows:

In our reduction,  $|V|$  is at least  $\prod_{i=1}^{b/b_{\text{micro}}} q_i$ , which is also the bound on each coordinate of the reduction:  $\psi(x)_i$  equals to a CRR encoding of a vector with  $\{q_i\}_{i=1}^{b/b_{\text{micro}}}$ , whose value can be as large as  $\prod_{i=1}^{b/b_{\text{micro}}} q_i - 1$ . That is, all we want is to control the upper bound on the coordinates of the reduction.

Suppose we are constructing an “outer” reduction  $\varphi : \{0, 1\}^{b \cdot \ell} \rightarrow \mathbb{Z}^\ell$  from the “micro” reduction  $\psi : \{0, 1\}^{b_{\text{micro}} \cdot \ell} \rightarrow \mathbb{Z}^\ell$  with coordinate upper bound  $L_\psi$  ( $\psi(x)_i \leq L_\psi$ ), and let  $L_\psi = \ell^{\kappa \cdot b_{\text{micro}}}$  (that is,  $\kappa$  is the extra factor comparing to the ideal case). Recall that we have to ensure  $q_i > \psi(x) \cdot \psi(y)$  to make our construction work, and therefore we have to set  $q_i$  larger than  $L_\psi^2$ .

Then the coordinate upper bound for  $\varphi$  becomes  $L_\varphi = \prod_{i=1}^{b/b_{\text{micro}}} q_i \geq (L_\psi)^{2 \cdot b/b_{\text{micro}}} = \ell^{2\kappa \cdot b}$ .

Therefore, we can see that after one recursion, the “extra factor”  $\kappa$  at least doubles. Since our recursion proceeds in  $\Theta(\log^* n)$  rounds, we have to pay an extra  $2^{O(\log^* n)}$  factor on the exponent.

### 1.3 Related Works

**SETH-based Conditional Lower Bound.** SETH is one of the most fruitful conjectures in the Fine-Grained Complexity. There are numerous conditional lower bounds based on it for problems in P among different areas, including: dynamic data structures [58, 6, 10, 44, 53, 3, 45, 41], computational geometry [25, 72, 35], pattern matching [7, 21, 22, 26, 24], graph algorithms [63, 40, 8, 54]. See [69] for a very recent survey on SETH-based lower bounds (and more).

**Communication Complexity and Conditional Hardness.** The connection between communication protocols (in various model) for Set-Disjointness and SETH dates back at least

to [59], in which it is shown that a sub-linear, computational efficient protocol for 3-party Number-On-Forehead Set-Disjointness problem would refute SETH. And it is worth mentioning that [4]’s result builds on the  $\tilde{O}(\log n)$  IP communication protocol for Set-Disjointness in [1].

**Distributed PCP.** Using Algebraic Geometry codes, [64] obtains a better MA protocol, which in turn improves the efficiency of the previous distributed PCP construction of [5]. He then shows the  $n^{2-o(1)}$  time hardness for  $1 + o(1)$ -approximation to Bichromatic Closest Pair and  $o(d)$ -additive approximation to Max-IP $_{n,d}$  with this new technique.

[50] use the Distributed PCP framework to derive inapproximability results for  $k$ -Dominating Set under various assumptions. In particular, building on the techniques of [64], it is shown that under SETH,  $k$ -Dominating Set has no  $(\log n)^{1/\text{poly}(k,e(\varepsilon))}$  approximation in  $n^{k-\varepsilon}$  time<sup>11</sup>.

**Hardness of Approximation in P.** Making use of Chebychev embeddings, [12] prove a  $2^{\Omega\left(\frac{\sqrt{\log n}}{\log \log n}\right)}$  inapproximability lower bound on  $\{-1, 1\}$ -Max-IP.<sup>12</sup> [2] take an approach different from Distributed PCP, and shows that under certain complexity assumptions, LCS does not have a *deterministic*  $1 + o(1)$ -approximation in  $n^{2-\varepsilon}$  time. They also establish a connection with circuit lower bounds and show that the existence of such a *deterministic* algorithm implies  $\text{E}^{\text{NP}}$  does not have non-uniform linear-size Valiant Series Parallel circuits. In [4], it is improved to that any constant factor approximation deterministic algorithm for LCS in  $n^{2-\varepsilon}$  time implies that  $\text{E}^{\text{NP}}$  does not have non-uniform linear-size  $\text{NC}^1$  circuits. See [5] for more related results in hardness of approximation in P.

## Organization of the Paper

In Section 2, we introduce the needed preliminaries for this paper. In Section 3, we prove our characterizations for approximating Max-IP and other related results. In Section 4, we prove  $2^{O(\log^+ n)}$  dimensional hardness for  $\mathbb{Z}$ -Max-IP and other related problems. In Section 5, we establish the connection between NP · UPP communication protocols and SETH-based lower bounds for exact  $\mathbb{Z}$ -Max-IP. In Section 6, we present the  $O\left(\sqrt{n \log n \log \log n}\right)$  MA protocol for Set-Disjointness.

## 2 Preliminaries

We begin by introducing some notation. For an integer  $d$ , we use  $[d]$  to denote the set of integers from 1 to  $d$ . For a vector  $u$ , we use  $u_i$  to denote the  $i$ -th element of  $u$ .

We use  $\log(x)$  to denote the logarithm of  $x$  with respect to base 2 with ceiling as appropriate, and  $\ln(x)$  to denote the natural logarithm of  $x$ .

In our arguments, we use the iterated logarithm function  $\log^*(n)$ , which is defined recursively as follows:

$$\log^*(n) := \begin{cases} 0 & n \leq 1; \\ \log^*(\log n) + 1 & n > 1. \end{cases}$$

<sup>11</sup> where  $e : \mathbb{R}^+ \rightarrow \mathbb{N}$  is some function

<sup>12</sup> which is improved by Theorem 1.13

## 2.1 Fast Rectangular Matrix Multiplication

Similar to previous algorithms using the polynomial method, our algorithms make use of the algorithms for fast rectangular matrix multiplication.

► **Theorem 2.1** ([39]). *There is an  $N^{2+o(1)}$  time algorithm for multiplying two matrices  $A$  and  $B$  with size  $N \times N^\alpha$  and  $N^\alpha \times N$ , where  $\alpha > 0.31389$ .*

► **Theorem 2.2** ([33]). *There is an  $N^2 \cdot \text{polylog}(N)$  time algorithm for multiplying two matrices  $A$  and  $B$  with size  $N \times N^\alpha$  and  $N^\alpha \times N$ , where  $\alpha > 0.172$ .*

## 2.2 Number Theory

Here we recall some facts from number theory. In our reduction from OV to  $\mathbb{Z}$ -OV, we will apply the famous prime number theorem, which supplies a good estimate of the number of primes smaller than a certain number. See e.g. [19] for a reference on this.

► **Theorem 2.3** (Prime Number Theorem). *Let  $\pi(n)$  be the number of primes  $\leq n$ , then we have*

$$\lim_{n \rightarrow \infty} \frac{\pi(n)}{n / \ln n} = 1.$$

From a simple calculation, we obtain:

► **Lemma 2.4.** *There are  $10n$  distinct primes in  $[n + 1, n^2]$  for a large enough  $n$ .*

**Proof.** For a large enough  $n$ , from the prime number theorem, the number of primes in  $[n + 1, n^2]$  is equal to

$$\pi(n^2) - \pi(n) \sim n^2 / 2 \ln n - n / \ln n \gg 10n. \quad \blacktriangleleft$$

Next we recall the Chinese remainder theorem, and Chinese remainder representation.

► **Theorem 2.5.** *Given  $d$  pairwise co-prime integers  $q_1, q_2, \dots, q_d$ , and  $d$  integers  $r_1, r_2, \dots, r_d$ , there is exactly one integer  $0 \leq t < \prod_{i=1}^d q_i$  such that*

$$t \equiv r_i \pmod{q_i} \quad \text{for all } i \in [d].$$

*We call this  $t$  the Chinese remainder representation (or the CRR encoding) of the  $r_i$ 's (with respect to these  $q_i$ 's). We also denote*

$$t = \text{CRR}(\{r_i\}; \{q_i\})$$

*for convenience. We sometimes omit the sequence  $\{q_i\}$  for simplicity, when it is clear from the context.*

*Moreover,  $t$  can be computed in polynomial time with respect to the total bits of all the given integers.*

## 2.3 Communication Complexity

In our paper we will make use of a certain kind of MA protocol, we call them  $(m, r, \ell, s)$ -efficient protocols<sup>13</sup>.

<sup>13</sup>Our notations here are adopted from [50]. They also defined similar  $k$ -party communication protocols, while we only discuss 2-party protocols in this paper.

► **Definition 2.6.** We say an MA Protocol is  $(m, r, \ell, s)$ -efficient for a communication problem, if in the protocol:

- There are three parties Alice, Bob and Merlin in the protocol, Alice holds input  $x$  and Bob holds input  $y$ .
- Merlin sends an advice string  $z$  of length  $m$  to Alice, which is a function of  $x$  and  $y$ .
- Alice and Bob jointly toss  $r$  coins to obtain a random string  $w$  of length  $r$ .
- Given  $y$  and  $w$ , Bob sends Alice a message of length  $\ell$ .
- After that, Alice decides whether to accept or not.
  - When the answer is yes, Merlin has exactly one advice such that Alice always accept.
  - When the answer is no, or Merlin sends the wrong advice, Alice accepts with probability at most  $s$ .

## 2.4 Derandomization

We make use of expander graphs to reduce the amount of random coins needed in one of our communication protocols. We abstract the following result for our use here.

► **Theorem 2.7** (see e.g. Theorem 21.12 and Theorem 21.19 in [20]). *Let  $m$  be an integer, and set  $B \subseteq [m]$ . Suppose  $|B| \geq m/2$ . There is a universal constant  $c_1$  such that for all  $\varepsilon < 1/2$ , there is a  $\text{poly}(\log m, \log \varepsilon^{-1})$ -time computable function  $\mathcal{F} : \{0, 1\}^{\log m + c_1 \cdot \log \varepsilon^{-1}} \rightarrow [m]^{c_1 \cdot \log \varepsilon^{-1}}$ , such that*

$$\Pr_{w \in \{0,1\}^{\log m + c_1 \cdot \log \varepsilon^{-1}}} [a \notin B \text{ for all } a \in \mathcal{F}(w)] \leq \varepsilon,$$

here  $a \in \mathcal{F}(w)$  means  $a$  is one of the element in the sequence  $\mathcal{F}(w)$ .

## 3 Hardness of Approximate Max-IP

In this section we prove our characterizations of approximating Max-IP.

### 3.1 The Multiplicative Case

We begin with the proof of Theorem 1.5. We recap it here for convenience.

**Reminder of Theorem 1.5** *Letting  $\omega(\log n) < d < n^{o(1)}$  and  $t \geq 2$ , the following holds:*

1. *There is an  $n^{2-\Omega(1)}$ -time  $t$ -multiplicative-approximating algorithm for  $\text{Max-IP}_{n,d}$  if*

$$t = (d/\log n)^{\Omega(1)},$$

*and under SETH (or OVC), there is no  $n^{2-\Omega(1)}$ -time  $t$ -multiplicative-approximating algorithm for  $\text{Max-IP}_{n,d}$  if*

$$t = (d/\log n)^{o(1)}.$$

2. *Moreover, let  $\varepsilon = \min\left(\frac{\log t}{\log(d/\log n)}, 1\right)$ . There are  $t$ -multiplicative-approximating deterministic algorithms for  $\text{Max-IP}_{n,d}$  running in time*

$$O\left(n^{2+o(1)-0.31 \cdot \frac{1}{\varepsilon^{-1} + \frac{0.31}{2}}}\right) = O\left(n^{2+o(1)-\Omega(\varepsilon)}\right)$$

*or time*

$$O\left(n^{2-0.17 \cdot \frac{1}{\varepsilon^{-1} + \frac{0.17}{2}} \cdot \text{polylog}(n)}\right) = O\left(n^{2-\Omega(\varepsilon)} \cdot \text{polylog}(n)\right).$$

In Lemma 3.2, we construct the desired approximate algorithm and in Corollary 3.4 we prove the lower bound.

## The Algorithm

First we need the following simple lemma, which says that the  $k$ -th root of the sum of the  $k$ -th powers of non-negative reals gives a good approximation to their maximum.

► **Lemma 3.1.** *Let  $S$  be a set of non-negative real numbers,  $k$  be an integer, and  $x_{max} := \max_{x \in S} x$ . We have*

$$\left( \sum_{x \in S} x^k \right)^{1/k} \in [x_{max}, x_{max} \cdot |S|^{1/k}].$$

**Proof.** Since

$$\left( \sum_{x \in S} x^k \right) \in [x_{max}^k, |S| \cdot x_{max}^k],$$

the lemma follows directly by taking the  $k$ -th root of both sides. ◀

► **Lemma 3.2.** *Assuming  $\omega(\log n) < d < n^{o(1)}$  and letting  $\varepsilon = \min\left(\frac{\log t}{\log(d/\log n)}, 1\right)$ , there are  $t$ -multiplicative-approximating deterministic algorithms for  $\text{Max-IP}_{n,d}$  running in time*

$$O\left(n^{2+o(1)-0.31 \cdot \frac{1}{\varepsilon^{-1} + \frac{0.31}{2}}}\right) = O\left(n^{2+o(1)-\Omega(\varepsilon)}\right)$$

or time

$$O\left(n^{2-0.17 \cdot \frac{1}{\varepsilon^{-1} + \frac{0.17}{2}}} \cdot \text{polylog}(n)\right) = O\left(n^{2-\Omega(\varepsilon)} \cdot \text{polylog}(n)\right).$$

**Proof.** Let  $d = c \cdot \log n$ . From the assumption, we have  $c = \omega(1)$ , and  $\varepsilon = \min\left(\frac{\log t}{\log c}, 1\right)$ . When  $\log t > \log c$ , we simply use a  $c$ -multiplicative-approximating algorithm instead, hence in the following we assume  $\log t \leq \log c$ . We begin with the first algorithm here.

**Construction and Analysis of the Power of Sum Polynomial  $P_r(z)$ .** Let  $r$  be a parameter to be specified later and  $z$  be a vector from  $\{0, 1\}^d$ , consider the following polynomial

$$P_r(z) := \left( \sum_{i=1}^d z_i \right)^r.$$

Observe that since each  $z_i$  takes value in  $\{0, 1\}$ , we have  $z_i^k = z_i$  for  $k \geq 2$ . Therefore, by expanding out the polynomial and replacing all  $z_i^k$  with  $k \geq 2$  by  $z_i$ , we can write  $P_r(z)$  as

$$P_r(z) = \sum_{S \subseteq [d], |S| \leq r} c_S \cdot z_S.$$

In which  $z_S := \prod_{i \in S} z_i$ , and the  $c_S$ 's are the corresponding coefficients. Note that  $P_r(z)$  has

$$m := \sum_{k=0}^r \binom{d}{k} \leq \left( \frac{ed}{r} \right)^r$$

## 14:16 On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product

terms.

Then consider  $P_r(x, y) := P_r(x_1 \cdot y_1, x_2 \cdot y_2, \dots, x_d \cdot y_d)$ , plugging in  $z_i := x_i \cdot y_i$ , it can be written as

$$P_r(x, y) := \sum_{S \subseteq [d], |S| \leq r} c_S \cdot x_S \cdot y_S,$$

where  $x_S := \prod_{i \in S} x_i$ , and  $y_S$  is defined similarly.

**Construction and Analysis of the Batch Evaluation Polynomial  $P_r(X, Y)$ .** Now, let  $X$  and  $Y$  be two sets of  $b = t^{r/2}$  vectors from  $\{0, 1\}^d$ , we define

$$P_r(X, Y) := \sum_{x \in X, y \in Y} P_r(x, y) = \sum_{x \in X, y \in Y} (x \cdot y)^r.$$

By Lemma 3.1, we have

$$P_r(X, Y)^{1/r} \in [\text{OPT}(X, Y), \text{OPT}(X, Y) \cdot t],$$

recall that  $\text{OPT}(X, Y) := \max_{x \in X, y \in Y} x \cdot y$ .

**Embedding into Rectangle Matrix Multiplication.** Now, for  $x, y \in \{0, 1\}^d$ , we define the mapping  $\phi_x(x)$  as

$$\phi_x(x) := (c_{S_1} \cdot x_{S_1}, c_{S_2} \cdot x_{S_2}, \dots, c_{S_m} \cdot x_{S_m})$$

and

$$\phi_y(y) := (y_{S_1}, y_{S_2}, \dots, y_{S_m}),$$

where  $S_1, S_2, \dots, S_m$  is an enumeration of all sets  $S \subseteq [d]$  and  $|S| \leq r$ .

From the definition, it follows that

$$\phi_x(x) \cdot \phi_y(y) = P_r(x, y)$$

for every  $x, y \in \{0, 1\}^d$ .

Then for each  $X$  and  $Y$ , we map them into  $m$ -dimensional vectors  $\phi_X(X)$  and  $\phi_Y(Y)$  simply by a summation:

$$\phi_X(X) := \sum_{x \in X} \phi_x(x) \quad \text{and} \quad \phi_Y(Y) := \sum_{y \in Y} \phi_y(y).$$

We can see

$$\phi_X(X) \cdot \phi_Y(Y) = \sum_{x \in X} \phi_x(x) \cdot \sum_{y \in Y} \phi_y(y) = \sum_{x \in X} \sum_{y \in Y} P_r(x, y) = P_r(X, Y).$$

Given two sets  $A, B$  of  $n$  vectors from  $\{0, 1\}^d$ , we split  $A$  into  $n/b$  sets  $A_1, A_2, \dots, A_{n/b}$  of size  $b$ , and split  $B$  in the same way as well. Then we construct a matrix  $M_A(M_B)$  of size  $n/b \times m$ , such that the  $i$ -th row of  $M_A(M_B)$  is the vector  $\Phi_X(A_i)(\Phi_Y(B_i))$ . After that, the evaluation of  $P_r(A_i, B_j)$  for all  $i, j \in [n/b]$  can be reduced to compute the matrix product  $M_A \cdot M_B^T$ . After knowing all  $P_r(A_i, B_j)$ 's, we simply compute the maximum of them, whose  $r$ -th root gives us a  $t$ -multiplicative-approximating answer of the original problem.



**Analysis of the Running Time.** Finally, we are going to specify the parameter  $r$  and analyze the time complexity. In order to utilize the fast matrix multiplication algorithm from Theorem 2.1, we need to have

$$m \leq (n/b)^{0.313},$$

then our running time is simply  $(n/b)^{2+o(1)} = n^{2+o(1)}/b^2$ .

We are going to set  $r = k \cdot \log n / \log c$ , and our choice of  $k$  will satisfy  $k = \Theta(1)$ . We have

$$m \leq \left( \frac{e \cdot d}{r} \right)^r \leq \left( \frac{c \log n \cdot e}{k \cdot \log n / \log c} \right)^{k \cdot \log n / \log c},$$

and therefore

$$\log m \leq k \cdot \log n \left[ \log \frac{c \log c}{k} + 1 \right] / \log c.$$

Since  $c = \omega(1)$  and  $k = \Theta(1)$ , we have

$$\log m \leq (1 + o(1)) \cdot k \log n = k \log n + o(\log n).$$

Plugging in, we have

$$\begin{aligned} m &\leq (n/b)^{0.313} \\ \Leftrightarrow \log m &\leq 0.313 \cdot (\log n - \log b) \\ \Leftrightarrow k \log n &\leq 0.31 \cdot (\log n - \log b) \\ \Leftrightarrow 0.31 \cdot (r/2) \cdot \log t + k \log n &\leq 0.31 \log n && (b = t^{r/2}) \\ \Leftrightarrow \frac{\log n}{\log c} \cdot k \cdot \log t \cdot \frac{0.31}{2} + k \log n &\leq 0.31 \log n && (r = k \cdot \log n / \log c) \\ \Leftrightarrow k \cdot \left\{ 1 + \frac{\log t}{\log c} \cdot \frac{0.31}{2} \right\} &\leq 0.31 \\ \Leftrightarrow k = \frac{0.31}{1 + \frac{\log t}{\log c} \cdot \frac{0.31}{2}} &= \frac{0.31}{1 + \frac{0.31}{2} \cdot \varepsilon}. \end{aligned}$$

Note since  $\varepsilon \in [0, 1]$ ,  $k$  is indeed  $\Theta(1)$ .

Finally, with our choice of  $k$  specified, our running time is  $n^{2+o(1)}/b^2 = n^{2+o(1)}/t^r$ .

By a simple calculation,

$$\begin{aligned} \log t^r &= r \cdot \log t \\ &= k \cdot \log n / \log c \cdot \log t \\ &= \log n \cdot \left\{ \frac{\log t}{\log c} \cdot \frac{0.31}{1 + \frac{0.31}{2} \cdot \varepsilon} \right\} \\ &= \log n \cdot \frac{0.31\varepsilon}{1 + \frac{0.31}{2} \cdot \varepsilon} \\ &= \log n \cdot \frac{0.31}{\varepsilon^{-1} + \frac{0.31}{2}}. \end{aligned}$$

Hence, our running time is

$$n^{2+o(1)}/t^r = n^{2+o(1) - \frac{0.31}{\varepsilon^{-1} + \frac{0.31}{2}}}$$

as stated.

**The Second Algorithm.** The second algorithm follows exactly the same except for that we apply Theorem 2.2 instead, hence the constant 0.31 is replaced by 0.17. ◀

### Generalization to Non-negative Real Case

Note that Lemma 3.1 indeed works for a set of non-negative reals, we can observe that the above algorithm in fact works for  $\mathbb{R}^+$ -Max-IP $_{n,d}$  (which is the same as Max-IP except for that the sets consisting of non-negative reals):<sup>14</sup>

**Reminder of Corollary 1.7** Assume  $\omega(\log n) < d < n^{o(1)}$  and let  $\varepsilon = \min(\frac{\log t}{\log(d/\log n)}, 1)$ .

There is a  $t$ -multiplicative-approximating deterministic algorithm for  $\mathbb{R}^+$ -Max-IP $_{n,d}$  running in time

$$O\left(n^{2-\Omega(\varepsilon)} \cdot \text{polylog}(n)\right).$$

**Proof Sketch.** We can just use the same algorithm in Lemma 3.2, the only difference is on the analysis of the number of terms in  $P_r(z)$ : since  $z$  is no longer Boolean,  $P_r(z)$  is no longer multi-linear, and we need to switch to a general upper bound  $\binom{d+r}{r}$  on the number of terms for  $r$ -degree polynomials of  $d$  variables. This corollary then follows by a similar calculation as in Lemma 3.2. ◀

### The Lower Bound

The lower bound follows directly from the new MA protocol for Set-Disjointness in [64]. We present an explicit proof here for completeness.

To prove the lower bound, we need the following reduction from OV to  $t$ -multiplicative-approximating Max-IP.

► **Lemma 3.3** (Implicit in Theorem 4.1 of [64]). *There is a universal constant  $c_1$  such that, for every integer  $c$ , reals  $\varepsilon \in (0, 1]$  and  $\tau \geq 2$ ,  $OV_{n,c \log n}$  can be reduced to  $n^\varepsilon$  Max-IP $_{n,d}$  instances  $(A_i, B_i)$  for  $i \in [n^\varepsilon]$ , such that:*

- $d = \tau^{\text{poly}(c/\varepsilon)} \cdot \log n$ .
- Letting  $T = c \log n \cdot \tau^{c_1}$ , if there is  $a \in A$  and  $b \in B$  such that  $a \cdot b = 0$ , then there exists an  $i$  such that  $\text{OPT}(A_i, B_i) \geq T$ .
- Otherwise, for all  $i$  we must have  $\text{OPT}(A_i, B_i) \leq T/\tau$ .

The reduction above follows directly from the new MA communication protocols in [64] together with the use of expander graphs to reduce the amount of random coins. A proof for the lemma above can be found in Appendix D.

Now we are ready to show the lower bound on  $t$ -multiplicative-approximating Max-IP.

► **Corollary 3.4.** *Assuming SETH (or OVC), and letting  $d = \omega(\log n)$  and  $t \geq 2$ . There is no  $n^{2-\Omega(1)}$ -time  $t$ -multiplicative-approximating algorithm for Max-IP $_{n,d}$  if*

$$t = (d/\log n)^{o(1)}.$$

---

<sup>14</sup>In the following we assume a real RAM model of computation for simplicity.

**Proof.** Let  $c = d/\log n$ , then  $t = c^{o(1)}$  (recall that  $t$  and  $d$  are two functions of  $n$ ).

Suppose for contradiction that there is an  $n^{2-\varepsilon'}$  time  $t(n)$ -multiplicative-approximating algorithm  $\mathbb{A}$  for  $\text{Max-IP}(n, d)$  for some  $\varepsilon' > 0$ .

Let  $\varepsilon = \varepsilon'/2$ . Now, for every constant  $c_2$ , we apply the reduction in Lemma 3.3 with  $\tau = t$  to reduce an  $\text{OV}_{n, c_2 \log n}$  instance to  $n^\varepsilon$

$$\text{Max-IP}_{n, t^{\text{poly}(c_2/\varepsilon)} \cdot \log n} \equiv \text{Max-IP}_{n, t^{O(1)} \cdot \log n}$$

instances. Since  $t = c^{o(1)}$ , which means for sufficiently large  $n$ ,  $t^{O(1)} \cdot \log n = c^{o(1)} \cdot \log n = o(d)$ , and it in turn implies that for sufficiently large  $n$ ,  $n^\varepsilon$  calls to  $\mathbb{A}$  are enough to solve the  $\text{OV}_{n, c_2 \log n}$  instance.

Therefore, we can solve  $\text{OV}_{n, c_2 \log n}$  in  $n^{2-\varepsilon'} \cdot n^\varepsilon = n^{2-\varepsilon}$  time for all constant  $c_2$ . Contradiction to OVC.  $\blacktriangleleft$

Finally, the correctness of Theorem 1.5 follows directly from Lemma 3.2 and Corollary 3.4.

### 3.2 The Additive Case

In this subsection we prove Theorem 1.10. We first recap it here for convenience.

**Reminder of Theorem 1.10** *Letting  $\omega(\log n) < d < n^{o(1)}$  and  $0 \leq t \leq d$ , the following holds:*

1. *There is an  $n^{2-\Omega(1)}$ -time  $t$ -additive-approximating algorithm for  $\text{Max-IP}_{n,d}$  if*

$$t = \Omega(d),$$

*and under  $\text{SETH}$  (or  $\text{OVC}$ ), there is no  $n^{2-\Omega(1)}$ -time  $t$ -additive-approximating algorithm for  $\text{Max-IP}_{n,d}$  if*

$$t = o(d).$$

2. *Moreover, letting  $\varepsilon = \frac{t}{d}$ , there is a randomized*

$$O\left(n^{2-\Omega(\varepsilon^{1/3}/\log \varepsilon^{-1})}\right)$$

*time,  $t$ -additive-approximating algorithm for  $\text{Max-IP}_{n,d}$  when  $\varepsilon \gg \log^6 \log n / \log^3 n$ .*

We proceed similarly as in the multiplicative case by establishing the algorithm first.

#### The Algorithm

The algorithm is actually very easy, we simply apply the following algorithm from [13].

► **Lemma 3.5** (Implicit in Theorem 5.1 in [13]). *Assuming  $\varepsilon \gg \log^6 \log(d \log n) / \log^3 n$ , there is an*

$$n^{2-\Omega(\varepsilon^{1/3}/\log(\frac{d}{\varepsilon \log n}))}$$

*time  $\varepsilon \cdot d$ -additive-approximating randomized algorithm for  $\text{Max-IP}_{n,d}$ .*

► **Lemma 3.6.** *Let  $\varepsilon = \frac{\min(t, d)}{d}$ , there is a randomized*

$$O\left(n^{2-\Omega(\varepsilon^{1/3}/\log \varepsilon^{-1})}\right)$$

*time,  $t$ -additive-approximating algorithm for  $\text{Max-IP}_{n,d}$  when  $\varepsilon \gg \log^6 \log n / \log^3 n$ .*

**Proof.** When  $t > d$  the problem becomes trivial, so we can assume  $t \leq d$ , and now  $t = \varepsilon \cdot d$ .

Let  $\varepsilon_1 = \varepsilon/2$  and  $c_1$  be a constant to be specified later. Given an  $\text{Max-IP}_{n,d}$  instance with two sets  $A$  and  $B$  of vectors from  $\{0, 1\}^d$ , we create another  $\text{Max-IP}_{n,d_1}$  instance with sets  $\tilde{A}$ ,  $\tilde{B}$  and  $d_1 = c_1 \cdot \varepsilon_1^{-2} \cdot \log n$  as follows:

- Pick  $d_1$  uniform random indices  $i_1, i_2, i_3, \dots, i_{d_1} \in [d]$ , each  $i_k$  is an independent uniform random number in  $[d]$ .
- Then we construct  $\tilde{A}$  from  $A$  by reducing each  $a \in A$  into  $\tilde{a} = (a_{i_1}, a_{i_2}, \dots, a_{i_{d_1}}) \in \{0, 1\}^{d_1}$  and  $\tilde{B}$  from  $B$  in the same way.

Note for each  $a \in A$  and  $b \in B$ , by a Chernoff bound, we have

$$\Pr \left[ \left| \frac{\tilde{a} \cdot \tilde{b}}{d_1} - \frac{a \cdot b}{d} \right| \geq \varepsilon_1 \right] < 2e^{-2d_1\varepsilon_1^2} = 2n^{-2 \cdot c_1}.$$

By setting  $c_1 = 2$ , the above probability is smaller than  $1/n^3$ .

Hence, by a simple union bound, with probability at least  $1 - 1/n$ , we have

$$\left| \frac{\tilde{a} \cdot \tilde{b}}{d_1} - \frac{a \cdot b}{d} \right| \leq \varepsilon_1$$

for all  $a \in A$  and  $b \in B$ . Hence, it means that this reduction only changes the “relative inner product” ( $\frac{a \cdot b}{d}$  or  $\frac{\tilde{a} \cdot \tilde{b}}{d_1}$ ) of each pair by at most  $\varepsilon_1$ . Hence, the maximum of the “relative inner product” also changes by at most  $\varepsilon_1$ , and we have  $|\text{OPT}(A, B)/d - \text{OPT}(\tilde{A}, \tilde{B})/d_1| \leq \varepsilon_1$ .

Then we apply the algorithm in Lemma 3.5 on the instance with sets  $\tilde{A}$  and  $\tilde{B}$  with error  $\varepsilon = \varepsilon_1$  to obtain an estimate  $\tilde{O}$ , and our final answer is simply  $\frac{\tilde{O}}{d_1} \cdot d$ .

From the guarantee from Lemma 3.5, we have  $|\text{OPT}(\tilde{A}, \tilde{B})/d_1 - \tilde{O}/d_1| \leq \varepsilon_1$ , and therefore we have  $|\text{OPT}(A, B)/d - \tilde{O}/d_1| \leq 2\varepsilon_1 = \varepsilon$ , from which the correctness of our algorithm follows directly.

For the running time, note that the reduction part runs in linear time  $O(n \cdot d)$ , and the rest takes

$$n^{2-\Omega(\varepsilon^{1/3}/\log(\frac{d_1}{\varepsilon_1 \log n}))} = n^{2-\Omega(\varepsilon^{1/3}/\log \varepsilon^{-1})}$$

time. ◀

## The Lower Bound

The lower bound is already established in [64], we show it follows from Lemma 3.3 here for completeness.

► **Lemma 3.7** (Theorem 4.1 of [64]). *Assuming SETH (or OVC), and letting  $d = \omega(\log n)$  and  $t > 0$ , there is no  $n^{2-\Omega(1)}$ -time  $t$ -additive-approximating randomized algorithm for  $\text{Max-IP}_{n,d}$  if*

$$t = o(d).$$

**Proof.** Recall that  $t$  and  $d$  are all functions of  $n$ . Suppose for contradiction that there is an  $n^{2-\varepsilon'}$  time  $t(n)$ -additive-approximating algorithm  $\mathbb{A}$  for  $\text{Max-IP}(n, d)$  for some  $\varepsilon' > 0$ .

Let  $\varepsilon = \varepsilon'/2$ . Now, for every constant  $c_2$ , we apply the reduction in Lemma 3.3 with  $\tau = 2$  to reduce an  $\text{OV}_{n, c_2 \log n}$  instance to  $n^\varepsilon$

$$\text{Max-IP}_{n, 2^{\text{poly}(c_2/\varepsilon)} \cdot \log n} \equiv \text{Max-IP}_{n, d_1} \text{ where } d_1 = O(1) \cdot \log n$$

instances. In addition, from Lemma 3.3, to solve the  $\text{OV}_{c_2 \log n}$  instance, we only need to distinguish an additive gap of  $\frac{T}{2} = \Omega(\log n) = \Omega(d_1)$  for these Max-IP instances obtained via the reduction.

This can be solved, via  $n^\varepsilon$  calls to  $\mathbb{A}$  as follows: for each  $\text{Max-IP}_{n, d_1}$  instance  $\mathcal{I}$  we get, since  $d = \omega(\log n)$ , which means for a sufficiently large  $n$ ,  $d_1 = O(\log n) \ll d$ , and we can duplicate each coordinate  $d/d_1$  times (for simplicity we assume  $d_1 | d$  here), to obtain an  $\text{Max-IP}_{n, d}$  instance  $\mathcal{I}^{\text{new}}$ , such that  $\text{OPT}(\mathcal{I}^{\text{new}}) = d/d_1 \cdot \text{OPT}(\mathcal{I})$ . Then  $\mathbb{A}$  can be used to estimate  $\text{OPT}(\mathcal{I}^{\text{new}})$  within an additive error  $t = o(d)$ . Scaling its estimate by  $\frac{d_1}{d}$ , it can also be used to estimate  $\text{OPT}(\mathcal{I})$  within an additive error  $o(d_1) = o(\log n) \leq T/2$  for sufficiently large  $n$ .

Therefore, we can solve  $\text{OV}_{n, c_2 \log n}$  in  $n^{2-\varepsilon'} \cdot n^\varepsilon = n^{2-\varepsilon}$  time for all constant  $c_2$ . Contradiction to OVC.  $\blacktriangleleft$

Finally, the correctness of Theorem 1.10 follows directly from Lemma 3.6 and Lemma 3.7.

### 3.3 Adaption for All-Pair-Max-IP

Now we sketch the adaption for our algorithms to work for the All-Pair-Max-IP problem.

**Reminder of Corollary 1.12** *Suppose  $\omega(\log n) < d < n^{o(1)}$ , and let*

$$\varepsilon_M := \min\left(\frac{\log t}{\log(d/\log n)}, 1\right) \text{ and } \varepsilon_A := \frac{\min(t, d)}{d}.$$

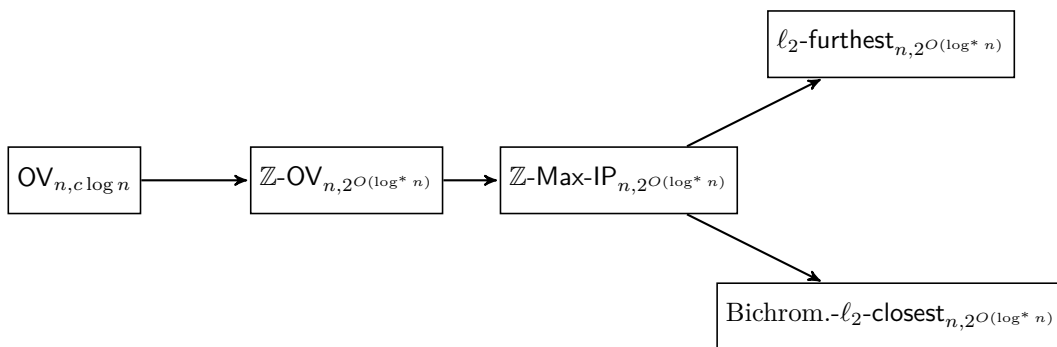
*There is an  $n^{2-\Omega(\varepsilon_M)}$  polylog( $n$ ) time  $t$ -multiplicative-approximating algorithm and an  $n^{2-\Omega(\varepsilon_A^{1/3}/\log \varepsilon_A^{-1})}$  time  $t$ -additive-approximating algorithm for All-Pair-Max-IP $_{n, d}$ , when  $\varepsilon_A \gg \log^6 \log n / \log^3 n$ .*

**Proof Sketch.** Note that the algorithm in Lemma 3.5 from [13] actually works for the All-Pair-Max-IP $_{n, d}$ . Hence, we can simply apply that algorithm after the coordinate sampling phase, and obtain a  $t$ -additive-approximating algorithm for All-Pair-Max-IP $_{n, d}$ .

For  $t$ -multiplicative-approximating algorithm, suppose we are given with two sets  $A$  and  $B$  of  $n$  vectors from  $\{0, 1\}^d$ . Instead of partitioning both of them into  $n/b$  subsets  $A_i$ 's and  $B_i$ 's (the notations used here are the same as in the proof of Lemma 3.2), we only partition  $B$  into  $n/b$  subsets  $B_1, B_2, \dots, B_{n/b}$  of size  $b$ , and calculate  $P_r(x, B_i) := \sum_{y \in B_i} P_r(x, y)$  for all  $x \in A$  and  $i \in [n/b]$  using similar reduction to rectangle matrix multiplication as in Lemma 3.2. By a similar analysis, these can be done in  $n^{2-\Omega(\varepsilon_M)} \cdot \text{polylog}(n)$  time, and with these informations we can compute the  $t$ -multiplicative-approximating answers for the given All-Pair-Max-IP $_{n, d}$  instance.  $\blacktriangleleft$

### 3.4 Improved Hardness for LCS-Closest Pair Problem

We finish this section with the proof of Corollary 1.9. First we abstract the reduction from Max-IP to LCS-Closest-Pair in [5] here.



■ **Figure 1** A diagram for all reductions in this section.

► **Lemma 3.8** (Implicit in Theorem 1.6 in [5]). *For big enough  $t$  and  $n$ ,  $t$ -multiplicative-approximating  $\text{Max-IP}_{n,d}$  reduces to  $t/2$ -multiplicative-approximating  $\text{LCS-Closest-Pair}_{n,d_1}$ , where  $d_1 = O(d^3 \log^2 n)$ .*

Now we are ready to prove Corollary 1.9 (restated below for convenience).

**Reminder of Corollary 1.9** *Assuming SETH (or OVC), for every  $t \geq 2$ ,  $t$ -multiplicative-approximating  $\text{LCS-Closest-Pair}_{n,d}$  requires  $n^{2-o(1)}$  time, if  $d = t^{\omega(1)} \cdot \log^5 n$ .*

**Proof.** From Corollary 3.4, assuming SETH (or OVC), for every  $t \geq 2$ , we have that  $2t$ -multiplicative-approximating  $\text{Max-IP}_{n,d}$  requires  $n^{2-o(1)}$  time if  $d = t^{\omega(1)} \cdot \log n$ . Then from Lemma 3.8,  $t$ -multiplicative-approximating  $\text{LCS-Closest-Pair}_{n,d_1}$  for  $d_1 = O(d^3 \log^2 n) = t^{\omega(1)} \cdot \log^5 n$  requires  $n^{2-o(1)}$  time. ◀

## 4 Hardness of Exact $\mathbb{Z}$ -Max-IP, Hopcroft's Problem and More

In this section we show hardness of Hopcroft's problem, exact  $\mathbb{Z}$ -Max-IP,  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair. Essentially our results follow from the framework of [72], in which it is shown that hardness of Hopcroft's problem implies hardness of other three problems, and is implied by dimensionality reduction for OV.

### The Organization of this Section

In Section 4.1, we prove the improved dimensionality reduction for OV. In Section 4.2, we establish the hardness of Hopcroft's problem in  $2^{O(\log^* n)}$  dimensions with the improved reduction. In Section 4.3, we show Hopcroft's problem can be reduced to  $\mathbb{Z}$ -Max-IP and thus establish the hardness for the later one. In Section 4.4, we show  $\mathbb{Z}$ -Max-IP can be reduced to  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair, therefore the hardness for the later two problems follow. See Figure 1 for a diagram of all reductions covered in this section.

The reduction in last three subsections are all from [72] (either explicit or implicit), we make them explicit here for our ease of exposition and for making the paper self-contained.

#### 4.1 Improved Dimensionality Reduction for OV

We begin with the improved dimensionality reduction for OV. The following theorem is one of the technical cores of this paper, which makes use of the CRR encoding (see Theorem 2.5) recursively.

► **Theorem 4.1.** *Let  $b, \ell$  be two sufficiently large integers. There is a reduction  $\psi_{b,\ell} : \{0, 1\}^{b \cdot \ell} \rightarrow \mathbb{Z}^\ell$  and a set  $V_{b,\ell} \subseteq \mathbb{Z}$ , such that for every  $x, y \in \{0, 1\}^{b \cdot \ell}$ ,*

$$x \cdot y = 0 \Leftrightarrow \psi_{b,\ell}(x) \cdot \psi_{b,\ell}(y) \in V_{b,\ell}$$

and

$$0 \leq \psi_{b,\ell}(x)_i < \ell^{6^{\log^*(b)} \cdot b}$$

for all possible  $x$  and  $i \in [\ell]$ . Moreover, the computation of  $\psi_{b,\ell}(x)$  takes  $\text{poly}(b \cdot \ell)$  time, and the set  $V_{b,\ell}$  can be constructed in  $O\left(\ell^{O(6^{\log^*(b)} \cdot b)} \cdot \text{poly}(b \cdot \ell)\right)$  time.

► **Remark 4.2.** We didn't make much effort to minimize the base 6 above to keep the calculation clean, it can be replaced by any constant  $> 2$  with a tighter calculation.

**Proof.** We are going to construct our reduction in a recursive way.  $\ell$  will be the same throughout the proof, hence in the following we use  $\psi_b$  ( $V_b$ ) instead of  $\psi_{b,\ell}$  ( $V_{b,\ell}$ ) for simplicity.

#### Direct CRR for small $b$ :

When  $b < \ell$ , we use a direct Chinese remainder representation of numbers. We pick  $b$  primes  $q_1, q_2, \dots, q_b$  in  $[\ell + 1, \ell^2]$ , and use them for our CRR encoding.

Let  $x \in \{0, 1\}^{b \cdot \ell}$ , we partition it into  $\ell$  equal size groups, and use  $x^i$  to denote the  $i$ -th group, which is the sub-vector of  $x$  from the  $((i - 1) \cdot b + 1)$ -th bit to the  $(i \cdot b)$ -th bit.

Then we define  $\psi_b(x)$  as

$$\psi_b(x) := \left( \text{CRR} \left( \{x_j^1\}_{j=1}^b \right), \text{CRR} \left( \{x_j^2\}_{j=1}^b \right), \dots, \text{CRR} \left( \{x_j^\ell\}_{j=1}^b \right) \right).$$

That is, the  $i$ -th coordinate of  $\psi_b(x)$  is the CRR encoding of the  $i$ -th sub-vector  $x^i$  with respect to the primes  $q_j$ 's.

Now, for  $x, y \in \{0, 1\}^{b \cdot \ell}$ , note that for  $j \in [b]$ ,

$$\begin{aligned} & \psi_b(x) \cdot \psi_b(y) \pmod{q_j} \\ & \equiv \sum_{i=1}^{\ell} \text{CRR} \left( \{x_j^i\}_{j=1}^b \right) \cdot \text{CRR} \left( \{y_j^i\}_{j=1}^b \right) \pmod{q_j} \\ & \equiv \sum_{i=1}^{\ell} x_j^i \cdot y_j^i \pmod{q_j}. \end{aligned}$$

Since the sum  $\sum_{i=1}^{\ell} x_j^i \cdot y_j^i$  is in  $[0, \ell]$ , and  $q_j > \ell$ , we can see

$$\sum_{i=1}^{\ell} x_j^i \cdot y_j^i = 0 \Leftrightarrow \psi_b(x) \cdot \psi_b(y) \equiv 0 \pmod{q_j}.$$

Therefore,  $x \cdot y = \sum_{j=1}^b \sum_{i=1}^{\ell} x_j^i \cdot y_j^i = 0$  is equivalent to that

$$\psi_b(x) \cdot \psi_b(y) \equiv 0 \pmod{q_j}$$

for every  $j \in [b]$ .

Finally, we have  $0 \leq \psi_b(x)_i < \prod_{j=1}^b p_j < \ell^{2 \cdot b} \leq \ell^{6^{\log^*(b)} \cdot b}$ . Therefore

$$\psi_b(x) \cdot \psi_b(y) < \ell^{6^{\log^*(b)} \cdot 2b+1},$$

and we can set  $V_b$  to be the set of all integers in  $[0, \ell^{6^{\log^*(b)} \cdot 2b+1}]$  that is 0 modulo all the  $p_j$ 's, and it is easy to see that

$$x \cdot y \Leftrightarrow \psi_b(x) \cdot \psi_b(y) \in V_b$$

for all  $x, y \in \{0, 1\}^{b \cdot \ell}$ .

### Recursive Construction for larger $b$ :

When  $b \geq \ell$ , suppose the theorem holds for all  $b' < b$ . Let  $b_{\text{micro}}$  be the number such that (we ignore the rounding issue here and pretend that  $b_{\text{micro}}$  is an integer for simplicity),

$$\ell^{6^{\log^*(b_{\text{micro}})} \cdot b_{\text{micro}}} = b.$$

Then we pick  $b/b_{\text{micro}}$  primes  $p_1, p_2, \dots, p_{b/b_{\text{micro}}}$  in  $[(b^2\ell), (b^2\ell)^2]$ , and use them as our reference primes in the CRR encodings.

Let  $x \in \{0, 1\}^{b \cdot \ell}$ , as before, we partition  $x$  into  $\ell$  equal size sub-vectors  $x^1, x^2, \dots, x^\ell$ , where  $x^i$  consists of the  $((i-1) \cdot b + 1)$ -th bit of  $x$  to the  $(i \cdot b)$ -th bit of  $x$ . Then we partition each  $x^i$  again into  $b/b_{\text{micro}}$  micro groups, each of size  $b_{\text{micro}}$ . We use  $x^{i,j}$  to denote the  $j$ -th micro group of  $x^i$  after the partition.

Now, we use  $x^{[j]}$  to denote the concatenation of the vectors  $x^{1,j}, x^{2,j}, \dots, x^{\ell,j}$ . That is,  $x^{[j]}$  is the concatenation of the  $j$ -th micro group in each of the  $\ell$  groups. Note that  $x^{[j]} \in \{0, 1\}^{b_{\text{micro}} \cdot \ell}$ , and can be seen as a smaller instance, on which we can apply  $\psi_{b_{\text{micro}}}$ .

Our recursive construction then goes in two steps. In the first step, we make use of  $\psi_{b_{\text{micro}}}$ , and transform each  $b_{\text{micro}}$ -size micro group into a single number in  $[0, b)$ . This step transforms  $x$  from a vector in  $\{0, 1\}^{b \cdot \ell}$  into a vector  $S(x)$  in  $\mathbb{Z}^{(b/b_{\text{micro}}) \cdot \ell}$ . And in the second step, we use a similar CRR encoding as in the base case to encode  $S(x)$ , to get our final reduced vector in  $\mathbb{Z}^\ell$ .

$S(x)$  is simply

$$\begin{aligned} S(x) := & \left( \psi_{b_{\text{micro}}}(x^{[1]})_1, \psi_{b_{\text{micro}}}(x^{[2]})_1, \dots, \psi_{b_{\text{micro}}}(x^{[b/b_{\text{micro}}]})_1, \right. \\ & \psi_{b_{\text{micro}}}(x^{[1]})_2, \psi_{b_{\text{micro}}}(x^{[2]})_2, \dots, \psi_{b_{\text{micro}}}(x^{[b/b_{\text{micro}}]})_2, \\ & \dots, \dots, \dots \\ & \left. \psi_{b_{\text{micro}}}(x^{[1]})_\ell, \psi_{b_{\text{micro}}}(x^{[2]})_\ell, \dots, \psi_{b_{\text{micro}}}(x^{[b/b_{\text{micro}}]})_\ell \right). \end{aligned}$$

That is, we apply  $\psi_{b_{\text{micro}}}$  on all the  $x^{[j]}$ 's, and shrink all the corresponding micro-groups in  $x$  into integers. Again, we partition  $S$  into  $\ell$  equal size groups  $S^1, S^2, \dots, S^\ell$ .

Then we define  $\psi_b(x)$  as

$$\psi_b(x) := \left( \text{CRR} \left( \{S_j^1\}_{j=1}^{b/b_{\text{micro}}} \right), \text{CRR} \left( \{S_j^2\}_{j=1}^{b/b_{\text{micro}}} \right), \dots, \text{CRR} \left( \{S_j^\ell\}_{j=1}^{b/b_{\text{micro}}} \right) \right).$$

In other words, the  $i$ -th coordinate of  $\psi_b(x)$  is the CRR representation of the number sequence  $S^i$ , with respect to our primes  $\{q_j\}_{j=1}^{b/b_{\text{micro}}}$ .



Now, note that for  $x, y \in \{0, 1\}^{b \cdot \ell}$ ,  $x \cdot y = 0$  is equivalent to  $x^{[j]} \cdot y^{[j]} = 0$  for every  $j \in [b/b_{\text{micro}}]$ , which is further equivalent to

$$\psi_{b_{\text{micro}}}(x^{[j]}) \cdot \psi_{b_{\text{micro}}}(y^{[j]}) \in V_{b_{\text{micro}}}$$

for all  $j \in [b/b_{\text{micro}}]$ , by our assumption on  $\psi_{b_{\text{micro}}}$ .

Since  $0 \leq \psi_{b_{\text{micro}}}(x^{[j]})_i, \psi_{b_{\text{micro}}}(y^{[j]})_i < b$  for all  $x, y \in \{0, 1\}^{b \cdot \ell}$ ,  $i \in [\ell]$  and  $j \in [b/b_{\text{micro}}]$ , we also have  $\psi_{b_{\text{micro}}}(x^{[j]}) \cdot \psi_{b_{\text{micro}}}(y^{[j]}) < b^2 \cdot \ell$ , therefore we can assume that  $V_{b_{\text{micro}}} \subseteq [0, b^2 \ell)$ .

For all  $x, y \in \{0, 1\}^{b \cdot \ell}$  and  $j \in [b/b_{\text{micro}}]$ , we have

$$\begin{aligned} & \psi_b(x) \cdot \psi_b(y) \\ \equiv & \sum_{i=1}^{\ell} \text{CRR} \left( \{S(x)_j^i\}_{j=1}^{b/b_{\text{micro}}} \right) \cdot \text{CRR} \left( \{S(y)_j^i\}_{j=1}^{b/b_{\text{micro}}} \right) \pmod{p_j} \\ \equiv & \sum_{i=1}^{\ell} S(x)_j^i \cdot S(y)_j^i \pmod{p_j} \\ \equiv & \sum_{i=1}^{\ell} \psi_{b_{\text{micro}}}(x^{[j]})_i \cdot \psi_{b_{\text{micro}}}(y^{[j]})_i \pmod{p_j} \\ \equiv & \psi_{b_{\text{micro}}}(x^{[j]}) \cdot \psi_{b_{\text{micro}}}(y^{[j]}) \pmod{p_j}. \end{aligned}$$

Since  $p_j \geq b^2 \cdot \ell$ , we can determine  $\psi_{b_{\text{micro}}}(x^{[j]}) \cdot \psi_{b_{\text{micro}}}(y^{[j]})$  from  $\psi_b(x) \cdot \psi_b(y)$  by taking modulo  $p_j$ . Therefore,

$$x \cdot y = 0$$

is equivalent to

$$(\psi_b(x) \cdot \psi_b(y) \pmod{p_j}) \in V_{b_{\text{micro}}},$$

for every  $j \in [b/b_{\text{micro}}]$ .

Finally, recall that we have

$$\ell^{6^{\log^*(b_{\text{micro}})} \cdot b_{\text{micro}}} = b.$$

Taking logarithm of both sides, we have

$$6^{\log^*(b_{\text{micro}})} \cdot b_{\text{micro}} \cdot \log \ell = \log b.$$

Then we can upper bound  $\psi_b(x)_i$  by

$$\begin{aligned} \psi_b(x)_i & < \prod_{j=1}^{b/b_{\text{micro}}} p_j \\ & < (b^2 \ell)^{2 \cdot (b/b_{\text{micro}})} && (b \geq \ell.) \\ & \leq 2^{6 \cdot b/b_{\text{micro}}} \cdot b_{\text{micro}} \cdot \log b \\ & \leq 2^{6 \cdot b/b_{\text{micro}}} \cdot 6^{6^{\log^*(b_{\text{micro}})} \cdot b_{\text{micro}}} \cdot \log \ell \\ & \leq \ell^{6 \cdot 6^{\log^*(b_{\text{micro}})} \cdot b} \\ & \leq \ell^{6^{\log^*(b)} \cdot b} && (b_{\text{micro}} \leq \log b, \log^*(b_{\text{micro}}) + 1 \leq \log^*(\log b) + 1 = \log^*(b).) \end{aligned}$$

## 14:26 On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product

Therefore, we can set  $V_b$  as the set of integer  $t$  in  $[0, \ell^{6^{\log^*(b)} \cdot 2b+1})$  such that

$$(t \bmod p_j) \in V_{b_{\text{micro}}}$$

for every  $j \in [b/b_{\text{micro}}]$ . And it is easy to see this  $V_b$  satisfies our requirement.

Finally, it is easy to see that the straightforward way of constructing  $\psi_b(x)$  takes  $O(\text{poly}(b \cdot \ell))$  time, and we can construct  $V_b$  by enumerating all possible values of  $\psi_b(x) \cdot \psi_b(y)$  and check each of them in  $O(\text{poly}(b \cdot \ell))$  time. Since there are at most  $\ell^{O(6^{\log^*(b)} \cdot b)}$  such values,  $V_b$  can be constructed in

$$O\left(\ell^{O(6^{\log^*(b)} \cdot b)} \cdot \text{poly}(b \cdot \ell)\right)$$

time, which completes the proof. ◀

Now we prove Lemma 1.17, we recap its statement here for convenience.

**Reminder of Lemma 1.17** *Let  $1 \leq \ell \leq d$ . There is an*

$$O\left(n \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))} \cdot \text{poly}(d)\right) \text{-time}$$

*reduction from  $\text{OV}_{n,d}$  to  $\ell^{O(6^{\log^* d} \cdot (d/\ell))}$  instances of  $\mathbb{Z}\text{-OV}_{n,\ell+1}$ , with vectors of entries with bit-length  $O(d/\ell \cdot \log \ell \cdot 6^{\log^* d})$ .*

**Proof.** The proof is exactly the same as the proof for Lemma 1.1 in [72] with different parameters, we recap it here for convenience.

Given two sets  $A'$  and  $B'$  of  $n$  vectors from  $\{0, 1\}^d$ , we apply  $\psi_{d/\ell, \ell}$  to each of the vectors in  $A'$  ( $B'$ ) to obtain a set  $A$  ( $B$ ) of vectors from  $\mathbb{Z}^\ell$ . From Theorem 4.1, there is a  $(u, v) \in A' \times B'$  such that  $u \cdot v = 0$  if and only if there is a  $(u, v) \in A \times B$  such that  $u \cdot v \in V_{d/\ell, \ell}$ .

Now, for each element  $t \in V_{d/\ell, \ell}$ , we are going to construct two sets  $A_t$  and  $B_t$  of vectors from  $\mathbb{Z}^{\ell+1}$  such that there is a  $(u, v) \in A \times B$  with  $u \cdot v = t$  if and only if there is a  $(u, v) \in A_t \times B_t$  with  $u \cdot v = 0$ . We construct a set  $A_t$  as a collection of all vectors  $u_A = [u, 1]$  for  $u \in A$ , and a set  $B_t$  as a collection of all vectors  $v_B = [v, -t]$  for  $v \in B$ . It is easy to verify this reduction has the properties we want.

Note that there are at most  $\ell^{O(6^{\log^* d} \cdot (d/\ell))}$  numbers in  $V_{d/\ell, \ell}$ , so we have such a number of  $\mathbb{Z}\text{-OV}_{n,\ell+1}$  instances. And from Theorem 4.1, the reduction takes

$$O\left(n \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))} \cdot \text{poly}(d)\right)$$

time.

Finally, the bit-length of reduced vectors is bounded by

$$\log\left(\ell^{O(6^{\log^* d} \cdot (d/\ell))}\right) = O\left(d/\ell \cdot \log \ell \cdot 6^{\log^* d}\right),$$

which completes the proof. ◀

## A Transformation from Nonuniform Construction to Uniform Construction

The proof for Theorem 4.1 works recursively. In one recursive step, we reduce the construction of  $\psi_{b,\ell}$  to the construction of  $\psi_{b_{\text{micro}},\ell}$ , where  $b_{\text{micro}} \leq \log b$ . Applying this reduction  $\log^* n$  times, we get a sufficiently small instance that we can switch to a direct CRR construction.

An interesting observation here is that after applying the reduction only thrice, the block length parameter becomes  $b' \leq \log \log \log b$ , which is so small that we can actually use brute force to find the “optimal” construction  $\psi_{b',\ell}$  in  $b^{o(1)}$  time instead of recursing deeper. Hence, to find a construction better than Theorem 4.1, we only need to prove the existence of such a construction. See Appendix B for details.

### 4.2 Improved Hardness for Hopcroft’s Problem

In this subsection we are going to prove Theorem 1.18 using our new dimensionality reduction Lemma 1.17, we recap its statement here for completeness.

**Reminder of Theorem 1.18** [*Hardness of Hopcroft’s Problem in  $c^{\log^* n}$  Dimension*] *Assuming SETH (or OVC), there is a constant  $c$  such that  $\mathbb{Z}\text{-OV}_{n,c^{\log^* n}}$  with vectors of  $O(\log n)$ -bit entries requires  $n^{2-o(1)}$  time.*

**Proof.** The proof here follows roughly the same as the proof for Theorem 1.1 in [72].

Let  $c$  be an arbitrary constant and  $d := c \cdot \log n$ . We show that an oracle solving  $\mathbb{Z}\text{-OV}_{n,\ell+1}$  where  $\ell = 7^{\log^* n}$  in  $O(n^{2-\delta})$  time for some  $\delta > 0$  can be used to construct an  $O(n^{2-\delta+o(1)})$  time algorithm for  $\text{OV}_{n,d}$ , and therefore contradicts the OVC.

We simply invoke Lemma 1.17, note that we have

$$\begin{aligned} \log \left\{ \ell^{O(6^{\log^* d} \cdot (d/\ell))} \right\} &= \log \ell \cdot O \left( 6^{\log^* d} \cdot (d/\ell) \right) \\ &= O \left( \log^* n \cdot 6^{\log^* n} \cdot c \cdot \log n / 7^{\log^* n} \right) \\ &= O \left( \log^* n \cdot (6/7)^{\log^* n} \cdot c \cdot \log n \right) \\ &= o(\log n). \end{aligned}$$

Therefore, the reduction takes  $O(n \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))} \cdot \text{poly}(d)) = n^{1+o(1)}$  time, and an  $\text{OV}_{n,d}$  instance is reduced to  $n^{o(1)}$  instances of  $\mathbb{Z}\text{-OV}_{n,\ell+1}$ , and the reduced vectors have bit length  $o(\log n)$  as calculated above. We simply solve all these  $n^{o(1)}$  instances using our oracle, and this gives us an  $O(n^{2-\delta+o(1)})$  time algorithm for  $\text{OV}_{n,d}$ , which completes the proof. ◀

### 4.3 Hardness for $\mathbb{Z}$ -Max-IP

Now we move to hardness of exact  $\mathbb{Z}$ -Max-IP.

► **Theorem 4.3** (Implicit in Theorem 1.2 [72]). *There is an  $O(\text{poly}(d) \cdot n)$ -time algorithm which reduces a  $\mathbb{Z}\text{-OV}_{n,d}$  instance into a  $\mathbb{Z}\text{-Max-IP}_{n,d^2}$  instance.*

**Proof.** We remark here that this reduction is implicitly used in the proof of Theorem 1.2 in [72], we abstract it here only for our exposition.

## 14:28 On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product

Given a  $\mathbb{Z}$ -OV $_{n,d}$  instance with sets  $A, B$ . Consider the following polynomial  $P(x, y)$ , where  $x, y \in \mathbb{Z}^d$ .

$$P(x, y) = -(x \cdot y)^2 = \sum_{i,j \in [d]} -(x_i \cdot y_i) \cdot (x_j \cdot y_j) = \sum_{i,j \in [d]} -(x_i \cdot x_j) \cdot (y_i \cdot y_j).$$

It is easy to see that whether there is a  $(x, y) \in A \times B$  such that  $x \cdot y = 0$  is equivalent to whether the maximum value of  $P(x, y)$  is 0.

Now, for each  $x \in A$  and  $y \in B$ , we identify  $[d^2]$  with  $[d] \times [d]$  and construct  $\tilde{x}, \tilde{y} \in \mathbb{Z}^{d^2}$  such that

$$\tilde{x}_{(i,j)} = x_i \cdot x_j \quad \text{and} \quad \tilde{y}_{(i,j)} = -y_i \cdot y_j.$$

Then we have  $\tilde{x} \cdot \tilde{y} = P(x, y)$ . Hence, let  $\tilde{A}$  be the set of all these  $\tilde{x}$ 's, and  $\tilde{B}$  be the set of all these  $\tilde{y}$ 's, whether there is a  $(x, y) \in A \times B$  such that  $x \cdot y = 0$  is equivalent to whether  $\text{OPT}(\tilde{A}, \tilde{B}) = 0$ , and our reduction is completed. ◀

Now, Theorem 1.14 (restated below) is just a simple corollary of Theorem 4.3 and Theorem 1.18.

**Reminder of Theorem 1.14** *Assuming SETH (or OVC), there is a constant  $c$  such that every exact algorithm for  $\mathbb{Z}$ -Max-IP $_{n,d}$  for  $d = c^{\log^* n}$  dimensions requires  $n^{2-o(1)}$  time, with vectors of  $O(\log n)$ -bit entries.*

### A Dimensionality Reduction for Max-IP

The reduction  $\psi_{b,\ell}$  from Theorem 4.1 actually does more: for  $x, y \in \{0, 1\}^{b \cdot \ell}$ , from  $\psi_{b,\ell}(x) \cdot \psi_{b,\ell}(y)$  we can in fact determine the inner product  $x \cdot y$  itself, not only whether  $x \cdot y = 0$ .

Starting from this observation, together with Theorem 4.3, we can in fact derive a similar dimensionality self reduction from Max-IP to  $\mathbb{Z}$ -Max-IP, we defer its proof to Appendix A.

► **Corollary 4.4.** *Let  $1 \leq \ell \leq d$ . There is an*

$$O\left(n \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))} \cdot \text{poly}(d)\right)\text{-time}$$

*reduction from Max-IP $_{n,d}$  to  $d \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))}$  instances of  $\mathbb{Z}$ -Max-IP $_{n,(\ell+1)^2}$ , with vectors of entries with bit-length  $O\left(d/\ell \cdot \log \ell \cdot 6^{\log^* d}\right)$ .*

### 4.4 Hardness for $\ell_2$ -Furthest Pair and Bichromatic $\ell_2$ -Closest Pair

We finish the whole section with the proof of hardness of  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair. The two reductions below are slight adaptations of the ones in the proofs of Theorem 1.2 and Corollary 2.1 in [72].

► **Lemma 4.5.** *Assuming  $d = n^{o(1)}$ , there is an  $O(\text{poly}(d) \cdot n)$ -time algorithm which reduces a  $\mathbb{Z}$ -Max-IP $_{n,d}$  instance into an instance of  $\ell_2$ -Furthest Pair on  $2n$  points in  $\mathbb{R}^{d+2}$ . Moreover, if the  $\mathbb{Z}$ -Max-IP instance consists of vectors of  $O(\log n)$ -bit entries, so does the  $\ell_2$ -Furthest Pair instance.*

**Proof.** Let  $A, B$  be the sets in the  $\mathbb{Z}$ -Max-IP $_{n,d}$  instance, and  $k$  be the smallest integer such that all vectors from  $A$  and  $B$  consist of  $(k \cdot \log n)$ -bit entries.

Let  $W$  be  $n^{C \cdot k}$  where  $C$  is a large enough constant. Given  $x \in A$  and  $y \in B$ , we construct point

$$\tilde{x} = \left(x, \sqrt{W - \|x\|^2}, 0\right) \quad \text{and} \quad \tilde{y} = \left(-y, 0, \sqrt{W - \|y\|^2}\right),$$

that is, appending two corresponding values into the end of vectors  $x$  and  $-y$ .

Now, we can see that for  $x_1, x_2 \in A$ , the squared distance between their reduced points is

$$\|\tilde{x}_1 - \tilde{x}_2\|^2 = \|x_1 - x_2\|^2 \leq 4 \cdot d \cdot n^{2k}.$$

Similarly we have

$$\|\tilde{y}_1 - \tilde{y}_2\|^2 \leq 4 \cdot d \cdot n^{2k}$$

for  $y_1, y_2 \in B$ .

Next, for  $x \in A$  and  $y \in B$ , we have

$$\|\tilde{x} - \tilde{y}\|^2 = \|\tilde{x}\|^2 + \|\tilde{y}\|^2 - 2 \cdot \tilde{x} \cdot \tilde{y} = 2 \cdot W + 2 \cdot (x \cdot y) \geq 2 \cdot W - d \cdot n^{2k} \gg 4 \cdot d \cdot n^{2k},$$

the last inequality holds when we set  $C$  to be 5.

Putting everything together, we can see the  $\ell_2$ -furthest pair among all points  $\tilde{x}$ 's and  $\tilde{y}$ 's must be a pair of  $\tilde{x}$  and  $\tilde{y}$  with  $x \in A$  and  $y \in B$ . And maximizing  $\|\tilde{x} - \tilde{y}\|$  is equivalent to maximize  $x \cdot y$ , which proves the correctness of our reduction. Furthermore, when  $k$  is a constant, the reduced instance clearly only needs vectors with  $O(k) \cdot \log n = O(\log n)$ -bit entries.  $\blacktriangleleft$

**► Lemma 4.6.** *Assuming  $d = n^{o(1)}$ , there is an  $O(\text{poly}(d) \cdot n)$ -time algorithm which reduces a  $\mathbb{Z}$ -Max-IP $_{n,d}$  instance into an instance of Bichromatic  $\ell_2$ -Closest Pair on  $2n$  points in  $\mathbb{R}^{d+2}$ . Moreover, if the  $\mathbb{Z}$ -Max-IP instance consists of vectors of  $O(\log n)$ -bit entries, so does the Bichromatic  $\ell_2$ -Closest Pair instance.*

**Proof.** Let  $A, B$  be the sets in the  $\mathbb{Z}$ -Max-IP $_{n,d}$  instance, and  $k$  be the smallest integer such that all vectors from  $A$  and  $B$  consist of  $(k \cdot \log n)$ -bit entries.

Let  $W$  be  $n^{C \cdot k}$  where  $C$  is a large enough constant. Given  $x \in A$  and  $y \in B$ , we construct point

$$\tilde{x} = \left(x, \sqrt{W - \|x\|^2}, 0\right) \quad \text{and} \quad \tilde{y} = \left(y, 0, \sqrt{W - \|y\|^2}\right),$$

that is, appending two corresponding values into the end of vectors  $x$  and  $-y$ . And our reduced instance is to find the closest point between the set  $\tilde{A}$  (consisting of all these  $\tilde{x}$  where  $x \in A$ ) and the set  $\tilde{B}$  (consisting of all these  $\tilde{y}$  where  $y \in B$ ).

Next, for  $x \in A$  and  $y \in B$ , we have

$$\|\tilde{x} - \tilde{y}\|^2 = \|\tilde{x}\|^2 + \|\tilde{y}\|^2 - 2 \cdot \tilde{x} \cdot \tilde{y} = 2 \cdot W - 2 \cdot (x \cdot y) \geq 2 \cdot W - d \cdot n^{2k} \gg 4 \cdot d \cdot n^{2k},$$

the last inequality holds when we set  $C$  to be 5.

Hence minimizing  $\|\tilde{x} - \tilde{y}\|$  where  $x \in A$  and  $y \in B$  is equivalent to maximize  $x \cdot y$ , which proves the correctness of our reduction. Furthermore, when  $k$  is a constant, the reduced instance clearly only needs vectors with  $O(k) \cdot \log n = O(\log n)$ -bit entries.  $\blacktriangleleft$

## 14:30 On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product

Now Theorem 1.15 and Theorem 1.16 (restated below) are simple corollaries of Lemma 4.5, Lemma 4.6 and Theorem 1.14.

**Reminder of Theorem 1.15** [Hardness of  $\ell_2$ -Furthest Pair in  $c^{\log^* n}$  Dimension] Assuming *SETH* (or *OVC*), there is a constant  $c$  such that  $\ell_2$ -Furthest Pair in  $c^{\log^* n}$  dimensions requires  $n^{2-o(1)}$  time, with vectors of  $O(\log n)$ -bit entries.

**Reminder of Theorem 1.16** [Hardness of Bichromatic  $\ell_2$ -closest Pair in  $c^{\log^* n}$  Dimension] Assuming *SETH* (or *OVC*), there is a constant  $c$  such that Bichromatic  $\ell_2$ -Closest Pair in  $c^{\log^* n}$  dimensions requires  $n^{2-o(1)}$  time, with vectors of  $O(\log n)$ -bit entries.

### 5 NP · UPP communication protocol and Exact Hardness for $\mathbb{Z}$ -Max-IP

We note that the inapproximability results for (Boolean) Max-IP is established via a connection to the MA communication complexity protocol of Set-Disjointness [5]. In the light of this, in this section we view our reduction from OV to  $\mathbb{Z}$ -Max-IP (Lemma 1.17 and Theorem 4.3) in the perspective of communication complexity.

We observe that in fact, our reduction can be understood as an NP · UPP communication protocol for Set Disjointness. Moreover, we show that if we can get a slightly better NP · UPP communication protocol for Set-Disjointness, then we would be able to prove  $\mathbb{Z}$ -Max-IP is hard even for  $\omega(1)$  dimensions (and also  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair).

#### 5.1 NP · UPP Communication Protocol for Set-Disjointness

First, we rephrase the results of Lemma 1.17 and Theorem 4.3 in a more convenience way for our use here.

► **Lemma 5.1** (Rephrasing of Lemma 1.17 and Theorem 4.3). *Let  $1 \leq \ell \leq d$ , and  $m = \ell^{O(6^{\log^* d} \cdot (d/\ell))}$ . There exists a family of functions*

$$\psi_{\text{Alice}}^i, \psi_{\text{Bob}}^i : \{0, 1\}^d \rightarrow \mathbb{R}^{(\ell+1)^2}$$

for  $i \in [m]$  such that:

- when  $x \cdot y = 0$ , there is an  $i$  such that  $\psi_{\text{Alice}}^i(x) \cdot \psi_{\text{Bob}}^i(y) \geq 0$ ;
- when  $x \cdot y > 0$ , for all  $i$   $\psi_{\text{Alice}}^i(x) \cdot \psi_{\text{Bob}}^i(y) < 0$ ;
- all  $\psi_{\text{Alice}}^i(x)$  and  $\psi_{\text{Bob}}^i(y)$  can be computed in  $\text{poly}(d)$  time.

We also need the standard connection between UPP communication protocols and sign-rank [60] (see also Chapter 4.11 of [48]).

► **Lemma 5.2** (Equivalence of sign-rank and UPP communication protocol [60]). *The following statements are equivalent:*

- There is a  $d$ -cost UPP communication protocol for a problem  $F : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ , where  $\mathcal{X}$  and  $\mathcal{Y}$  are input sets of Alice and Bob respectively.
- There are mappings  $\psi^{\mathcal{X}} : \mathcal{X} \rightarrow \mathbb{R}^{2^d}$  and  $\psi^{\mathcal{Y}} : \mathcal{Y} \rightarrow \mathbb{R}^{2^d}$  such that for all  $(x, y) \in \mathcal{X} \times \mathcal{Y}$ :
  - if  $F(x, y) = 1$ ,  $\psi^{\mathcal{X}}(x) \cdot \psi^{\mathcal{Y}}(y) \geq 0$ ;
  - otherwise,  $\psi^{\mathcal{X}}(x) \cdot \psi^{\mathcal{Y}}(y) < 0$ .

From the above lemmas, we immediately get the needed communication protocol and prove Theorem 1.21 (restated below for convenience).

**Reminder of Theorem 1.21** *For all  $1 \leq \alpha \leq n$ , there is an*

$$\left(\alpha \cdot 6^{\log^* n} \cdot (n/2^\alpha), O(\alpha)\right)\text{-computational-efficient}$$

NP · UPP communication protocol for  $DISJ_n$ .

**Proof Sketch.** We set  $\alpha = \log \ell$  here. Given the function families  $\{\psi_{\text{Alice}}^i\}, \{\psi_{\text{Bob}}^i\}$  from Lemma 5.1, Merlin just sends the index  $i \in [m]$  and the rest follows from Lemma 5.2. ◀

## 5.2 Slightly Better Protocols Imply Hardness in $\omega(1)$ Dimensions

Finally, we show that if we have a slightly better NP · UPP protocol for Set-Disjointness, then we can show  $\mathbb{Z}$ -Max-IP requires  $n^{2-o(1)}$  time even for  $\omega(1)$  dimensions (and so do  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair). We restate Theorem 1.22 here for convenience.

**Reminder of Theorem 1.22** *Assuming SETH (or OVC), if there is an increasing and unbounded function  $f$  such that for all  $1 \leq \alpha \leq n$ , there is a*

$$(n/f(\alpha), \alpha)\text{-computational-efficient}$$

NP · UPP communication protocol for  $DISJ_n$ , then  $\mathbb{Z}$ -Max-IP $_{n,\omega(1)}$  requires  $n^{2-o(1)}$  time with vectors of  $\text{polylog}(n)$ -bit entries. The same holds for  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair.

**Proof.** Suppose otherwise, there is an algorithm  $\mathbb{A}$  for  $\mathbb{Z}$ -Max-IP $_{n,d}$  running in  $n^{2-\varepsilon_1}$  time for all constant  $d$  and for a constant  $\varepsilon_1 > 0$  (note for the sake of Lemma 4.5 and Lemma 4.6, we only need to consider  $\mathbb{Z}$ -Max-IP here).

Now, let  $c$  be an arbitrary constant, we are going to construct an algorithm for  $OV_{n,c \log n}$  in  $n^{2-\Omega(1)}$  time, which contradicts OVC.

Let  $\varepsilon = \varepsilon_1/2$ , and  $\alpha$  be the first number such that  $c/f(\alpha) < \varepsilon$ , note that  $\alpha$  is also a constant. Consider the  $(c \log n/f(\alpha), \alpha)$ -computational-efficient NP · UPP protocol  $\Pi$  for  $DISJ_{c \log n}$ , and let  $A, B$  be the two sets in the  $OV_{n,c \log n}$  instance. Our algorithm via reduction works as follows:

- There are  $2^\alpha$  possible messages in  $\{0, 1\}^\alpha$ , let  $m_1, m_2, \dots, m_{2^\alpha}$  be an enumeration of them.
- We first enumerate all possible advice strings from Merlin in  $\Pi$ , there are  $2^{c \log n/f(\alpha)} \leq 2^{\varepsilon \cdot \log n} = n^\varepsilon$  such strings, let  $\phi \in \{0, 1\}^{\varepsilon \cdot \log n}$  be such an advice string.
  - For each  $x \in A$ , let  $\psi_{\text{Alice}}(x) \in \mathbb{R}^{2^\alpha}$  be the probabilities that Alice accepts each message from Bob. That is,  $\psi_{\text{Alice}}(x)_i$  is the probability that Alice accepts the message  $m_i$ , given its input  $x$  and the advice  $\phi$ .
  - Similarly, for each  $y \in B$ , let  $\psi_{\text{Bob}}(y) \in \mathbb{R}^{2^\alpha}$  be the probabilities that Bob sends each message. That is,  $\psi_{\text{Bob}}(y)_i$  is the probability that Bob sends the message  $m_i$ , give its input  $y$  and the advice  $\phi$ .
  - Then, for each  $x \in A$  and  $y \in B$ ,  $\psi_{\text{Alice}}(x) \cdot \psi_{\text{Bob}}(y)$  is precisely the probability that Alice accepts at the end when Alice and Bob holds  $x$  and  $y$  correspondingly and the advice is  $\phi$ . Now we let  $A_\phi$  be the set of all the  $\psi_{\text{Alice}}(x)$ 's, and  $B_\phi$  be the set of all the  $\psi_{\text{Bob}}(y)$ 's.

## 14:32 On The Hardness of Approximate and Exact (Bichromatic) Maximum Inner Product

- If there is a  $\phi$  such that  $\text{OPT}(A_\phi, B_\phi) \geq 1/2$ , then we output yes, and otherwise output no.

From the definition of  $\Pi$ , it is straightforward to see that the above algorithm solves  $\text{OV}_{n, c \cdot \log n}$ . Moreover, notice that from the computational-efficient property of  $\Pi$ , the reduction itself works in  $n^{1+\varepsilon} \cdot \text{polylog}(n)$  time, and all the vectors in  $A_\phi$ 's and  $B_\phi$ 's have at most  $\text{polylog}(n)$  bit precision, which means  $\text{OPT}(A_\phi, B_\phi)$  can be solved by a call to  $\mathbb{Z}\text{-Max-IP}_{n, 2^\alpha}$  with vectors of  $\text{polylog}(n)$ -bit entries.

Hence, the final running time for the above algorithm is bounded by  $n^\varepsilon \cdot n^{2-\varepsilon_1} = n^{2-\varepsilon}$  ( $2^\alpha$  is still a constant), which contradicts the OVC. ◀

### 6 Improved MA Protocols

In this section we prove Theorem 1.24 (restated below for convenience).

**Reminder of Theorem 1.24** *There is an MA protocol for  $\text{DISJ}_n$  and  $\text{IP}_n$  with communication complexity*

$$O\left(\sqrt{n \log n \log \log n}\right).$$

To prove Theorem 1.24, we need the following intermediate problem.

► **Definition 6.1** (The Inner Product Modulo  $p$  Problem ( $\text{IP}_n^p$ )). Let  $p$  and  $n$  be two positive integers, in  $\text{IP}_n^p$ , Alice and Bob are given two vectors  $X$  and  $Y$  in  $\{0, 1\}^n$ , and they want to compute  $X \cdot Y \pmod{p}$ .

Note that  $\text{IP}_n$  and  $\text{IP}_n^p$  are not Boolean functions, so we need to generalize the definition of an MA protocol. In an MA protocol for  $\text{IP}_n$ , Merlin sends the answer directly to Alice together with a proof to convince Alice and Bob. The correctness condition becomes that for the right answer  $X \cdot Y$ , Merlin has a proof such that Alice and Bob will accept with high probability (like  $2/3$ ). And the soundness condition becomes that for the wrong answers, every proof from Merlin will be rejected with high probability.

We are going to use the following MA protocol for  $\text{IP}_n^p$ , which is a slight adaption from the protocol in [64].

► **Lemma 6.2** (Implicit in Theorem 3.1 of [64]). *For a sufficiently large prime  $q$  and integers  $T$  and  $n$ , there is an*

$$\left(O(n/T \cdot \log q), \log n + O(1), O(T \cdot \log q), 1/2\right)\text{-efficient}$$

MA protocol for  $\text{IP}_n^q$ .

**Proof Sketch.** The only adaption is that we just use the field  $\mathbb{F}_{q^2}$  with respect to the given prime  $q$ . (In the original protocol it is required that  $q \geq T$ .) ◀

Now we ready to prove Theorem 1.24.

**Proof of Theorem 1.24.** Since a  $\text{IP}_n$  protocol trivially implies a  $\text{DISJ}_n$  protocol, we only need to consider  $\text{IP}_n$  in the following.

Now, let  $x$  be the number such that  $x^x = n$ , for convenience we are going to pretend that  $x$  is an integer. It is easy to see that  $x = \Theta(\log n / \log \log n)$ . Then we pick  $10x$



distinct primes  $p_1, p_2, \dots, p_{10x}$  in  $[x+1, x^2]$  (we can assume that  $n$  is large enough to make  $x$  satisfy the requirement of Lemma 2.4). Let  $T$  be a parameter, we use  $\Pi_{p_i}$  to denote the  $(O(n/T \cdot \log p_i), \log n + O(1), O(T \cdot \log p_i), 1/2)$ -efficient MA protocol for  $\mathbb{IP}_n^{p_i}$ .

Our protocol for  $\mathbb{IP}_n$  works as follows:

- Merlin sends Alice all the advice strings from the protocols  $\Pi_{p_1}, \Pi_{p_2}, \dots, \Pi_{p_{10x}}$ , together with a presumed inner product  $0 \leq z \leq n$ .
- Note that  $\Pi_{p_i}$  contains the presumed value of  $X \cdot Y \pmod{p_i}$ , Alice first checks whether  $z$  is consistent with all these  $\Pi_{p_i}$ 's, and rejects immediately if it does not.
- Alice and Bob jointly toss  $O(\log(10x))$  coins, to pick a uniform random number  $i^* \in [10x]$ , and then they simulate  $\Pi_{p_{i^*}}$ . That is, they pretend they are the Alice and Bob in the protocol  $\Pi_{p_{i^*}}$  with the advice from Merlin in  $\Pi_{p_{i^*}}$  (which Alice does have).

**Correctness.** Let  $X, Y \in \{0, 1\}^n$  be the vectors of Alice and Bob. If  $X \cdot Y = z$ , then by the definition of these protocols  $\Pi_{p_i}$ 's, Alice always accepts with the correct advice from Merlin.

Otherwise, let  $d = X \cdot Y \neq z$ , we are going to analyze the probability that we pick a "good"  $p_{i^*}$  such that  $p_{i^*}$  does not divide  $|d - z|$ . Since  $p_i > x$  for all  $p_i$ 's and  $x^x > n \geq |d - z|$ ,  $|d - z|$  cannot be a multiplier of more than  $x$  primes in  $p_i$ 's.

Therefore, with probability at least 0.9, our pick of  $p_{i^*}$  is good. And in this case, from the definition of the protocols  $\Pi_{p_i}$ 's, Alice and Bob would reject afterward with probability at least 1/2, as  $d \pmod{p_{i^*}}$  differs from  $z \pmod{p_{i^*}}$ . In summary, when  $X \cdot Y \neq z$ , Alice rejects with probability at least  $0.9/2 = 0.45$ , which finishes the proof for the correctness.

**Complexity.** Now, note that the total advice length is

$$O\left(n/T \cdot \sum_{i=1}^{10x} \log p_i\right) = O\left(n/T \cdot \log \prod_{i=1}^{10x} x^2\right) = O(n/T \cdot \log x^{20x}) = O(n/T \cdot \log n).$$

And the communication complexity between Alice and Bob is bounded by

$$O(T \cdot \log x^2) = O(T \cdot \log \log n).$$

Setting  $T = \sqrt{n \log n / \log \log n}$  balances the above two quantities, and we obtain the needed MA-protocol for  $\text{DISJ}_n$ . ◀

## 7 Future Works

We end our paper by discussing a few interesting research directions.

1. The most important open question from this paper is that can we further improve the dimensionality reduction for OV? It is certainly weird to consider  $2^{O(\log^* n)}$  to be the right answer for the limit of the dimensionality reduction. This term seems more like a product of the nature of our recursive construction and not the problem itself. We conjecture that there should be an  $\omega(1)$  dimensional reduction with a more direct construction. One possible direction is to combine the original polynomial-based construction from [72] together with our new number theoretical one. These two approaches seem completely different, hence a clever combination of them may solve our problem.
2. In order to prove  $\omega(1)$  dimensional hardness for  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair, we can also bypass the OV dimensionality reduction things by proving  $\omega(1)$  dimensional hardness for  $\mathbb{Z}$ -Max-IP directly. One possible way to approach this question is to start from the NP · UPP communication protocol connection as in Section 5

(apply Theorem 1.22), and (potentially) draw some connections from some known UPP communication protocols.

3. We have seen an efficient reduction from  $\mathbb{Z}$ -OV to  $\mathbb{Z}$ -Max-IP which only blows up the dimension quadratically, is there a similar reduction from  $\mathbb{Z}$ -Max-IP back to  $\mathbb{Z}$ -OV? Are  $\mathbb{Z}$ -Max-IP and  $\mathbb{Z}$ -OV equivalent?
4. By making use of the new AG-code based MA protocols, we can shave a  $\tilde{O}(\sqrt{\log n})$  factor from the communication complexity, can we obtain an  $O(\sqrt{n})$  MA communication protocol matching the lower bound for  $\text{DISJ}_n$ ? It seems new ideas are required. Since our MA protocol works for both DISJ and IP, and IP does seem to be a harder problem. It may be better to find an MA protocol only works for DISJ. It is worth noting that an  $O(\sqrt{n})$  AMA communication protocol for DISJ is given by [64], which doesn't work for IP.
5. Can the dependence on  $\varepsilon$  in the algorithms from Theorem 1.5 be further improved? Is it possible to apply ideas in the  $n^{2-1/\tilde{\Omega}(\sqrt{c})}$  algorithm for  $\text{Max-IP}_{n,c \log n}$  from [13]?
6. For the complexity of 2-multiplicative-approximation to  $\text{Max-IP}_{n,c \log n}$ , Theorem 1.5 implies that there is an algorithm running in  $n^{2-1/O(\log c)}$  time, the same as the best algorithm for  $\text{OV}_{n,c \log n}$  [9]. Is this just a coincidence? Or are there some connections between these two problems?
7. We obtain a connection between hardness of  $\mathbb{Z}$ -Max-IP and  $\text{NP} \cdot \text{UPP}$  communication protocols for Set-Disjointness. Can we get similar connections from other  $\text{NP} \cdot \mathcal{C}$  type communication protocols for Set-Disjointness? Some candidates include  $\text{NP} \cdot \text{SBP}$  and  $\text{NP} \cdot \text{promiseBQP}$  (QCMA).

---

## References

- 1 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *TOCT*, 1(1):2:1–2:54, 2009. doi:10.1145/1490270.1490272.
- 2 Amir Abboud and Arturs Backurs. Towards hardness of approximation for polynomial time problems. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 67. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- 3 Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In *Proceedings of the IEEE 57th Annual Symposium on Foundations of Computer Science*, pages 477–486, 2016.
- 4 Amir Abboud and Aviad Rubinfeld. Fast and deterministic constant factor approximation algorithms for lcs imply new circuit lower bounds. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 94. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- 5 Amir Abboud, Aviad Rubinfeld, and R. Ryan Williams. Distributed PCP theorems for hardness of approximation in P. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 25–36. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.12.
- 6 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *Proc. of the 55th FOCS*, pages 434–443, 2014.
- 7 Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *Proc. of the 41st ICALP*, pages 39–51, 2014.
- 8 Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 41–50. ACM, 2015.

- 9 Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 218–230. Society for Industrial and Applied Mathematics, 2015.
- 10 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *Proceedings of the IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 434–443, 2014.
- 11 Pankaj K Agarwal, Herbert Edelsbrunner, Otfried Schwarzkopf, and Emo Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete & Computational Geometry*, 6(3):407–422, 1991.
- 12 Thomas Dybdahl Ahle, Rasmus Pagh, Ilya Razenshteyn, and Francesco Silvestri. On the complexity of inner product similarity join. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 151–164. ACM, 2016.
- 13 Josh Alman, Timothy M Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 467–476. IEEE, 2016.
- 14 Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *Proc. of the 56th FOCS*, pages 136–150. IEEE, 2015.
- 15 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. of the 47th FOCS*, pages 459–468. IEEE, 2006.
- 16 Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Advances in Neural Information Processing Systems*, pages 1225–1233, 2015.
- 17 Alexandr Andoni, Piotr Indyk, Huy L Nguyen, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *Proc. of the 25th SODA*, pages 1018–1028. SIAM, 2014.
- 18 Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proc. of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 793–801. ACM, 2015.
- 19 Tom M. Apostol. *Introduction to analytic number theory*. Springer Science & Business Media, 2013.
- 20 Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: <http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264>.
- 21 Arturs Backurs and Piotr Indyk. Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false). In *Proc. of the 47th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 51–58, 2015.
- 22 Arturs Backurs and Piotr Indyk. Which regular expression patterns are hard to match? In *Proc. of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 457–466, 2016.
- 23 Jon Louis Bentley and Michael Ian Shamos. Divide-and-conquer in multidimensional space. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 220–230. ACM, 1976.
- 24 Karl Bringman and Marvin Künnemann. Multivariate fine-grained complexity of longest common subsequence. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1216–1235. SIAM, 2018.
- 25 Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In *Proc. of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 661–670, 2014.

- 26 Karl Bringmann, Allan Grønlund, and Kasper Green Larsen. A dichotomy for regular expression membership testing. *arXiv preprint arXiv:1611.00918*, 2016.
- 27 Harry Buhrman, Richard Cleve, Ronald De Wolf, and Christof Zalka. Bounds for small-error and zero-error quantum algorithms. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 358–368. IEEE, 1999.
- 28 Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 63–68. ACM, 1998.
- 29 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *IWPEC*, volume 5917, pages 75–85. Springer, 2009.
- 30 Timothy M Chan. A (slightly) faster algorithm for klee’s measure problem. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 94–100. ACM, 2008.
- 31 Tobias Christiani. A framework for similarity search with space-time tradeoffs using locality-sensitive filtering. In *Proc. of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 31–46. SIAM, 2017.
- 32 Tobias Christiani and Rasmus Pagh. Set similarity search beyond minhash. *arXiv preprint arXiv:1612.07710*, 2016.
- 33 Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal on Computing*, 11(3):467–471, 1982.
- 34 Svyatoslav Covanov and Emmanuel Thomé. Fast integer multiplication using generalized fermat primes. *arXiv preprint arXiv:1502.02800*, 2015.
- 35 Roe David, CS Karthik, and Bundit Laekhanukit. On the complexity of closest pair via polar-pair of point-sets. *CoRR*, abs/1608.03245, 2016.
- 36 Ronald de Wolf. A note on quantum algorithms and the minimal degree of epsilon-error polynomials for symmetric functions. *arXiv preprint arXiv:0802.1816*, 2008.
- 37 Martin Dietzfelbinger, Torben Hagerup, Jyrki Katajainen, and Martti Penttonen. A reliable randomized algorithm for the closest-pair problem. *Journal of Algorithms*, 25(1):19–51, 1997.
- 38 Martin Fürer. Faster integer multiplication. *SIAM Journal on Computing*, 39(3):979–1005, 2009.
- 39 Francois Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1029–1046. SIAM, 2018.
- 40 Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and R. Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. In *Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2162–2181, 2017.
- 41 Isaac Goldstein, Tsvi Kopelowitz, Moshe Lewenstein, and Ely Porat. Conditional lower bounds for space/time tradeoffs. In Faith Ellen, Antonina Kolokolova, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures*, pages 421–436, Cham, 2017. Springer International Publishing.
- 42 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.
- 43 David Harvey, Joris Van Der Hoeven, and Grégoire Lecerf. Even faster integer multiplication. *Journal of Complexity*, 36:1–30, 2016.
- 44 Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-

- vector multiplication conjecture. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 21–30, 2015.
- 45 Monika Henzinger, Andrea Lincoln, Stefan Neumann, and Virginia Vassilevska Williams. Conditional hardness for sensitivity problems. *arXiv preprint arXiv:1703.01638*, 2017.
  - 46 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.
  - 47 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.
  - 48 Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer Science & Business Media, 2012.
  - 49 Matti Karppa, Petteri Kaski, and Jukka Kohonen. A faster subquadratic algorithm for finding outlier correlations. In *Proc. of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1288–1305. Society for Industrial and Applied Mathematics, 2016.
  - 50 C.S. Karthik, Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *arXiv preprint arXiv:1711.11029*, 2017.
  - 51 Samir Khuller and Yossi Matias. A simple randomized sieve algorithm for the closest-pair problem. *Information and Computation*, 118(1):34–37, 1995.
  - 52 Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *Computational Complexity, 2003. Proceedings. 18th IEEE Annual Conference on*, pages 118–134. IEEE, 2003.
  - 53 Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3sum conjecture. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1272–1287, 2016.
  - 54 Robert Krauthgamer and Ohad Trabelsi. Conditional lower bounds for all-pairs max-flow. *arXiv preprint arXiv:1702.05805*, 2017.
  - 55 Jiří Matoušek. Efficient partition trees. *Discrete & Computational Geometry*, 8(3):315–334, 1992.
  - 56 Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete & Computational Geometry*, 10(2):157–182, 1993.
  - 57 Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric lshs for inner product search. In *Proc. of the 32nd International Conference on Machine Learning, ICML*, pages 1926–1934, 2015.
  - 58 Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 603–610, 2010.
  - 59 Mihai Patrascu and Ryan Williams. On the possibility of faster sat algorithms. In *Proc. of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1065–1075. SIAM, 2010.
  - 60 Ramamohan Paturi and Janos Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986.
  - 61 Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5, 2007.
  - 62 Parikshit Ram and Alexander G Gray. Maximum inner-product search using cone trees. In *Proc. of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 931–939. ACM, 2012.
  - 63 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proc. of the 45th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 515–524, 2013.

- 64 Aviad Rubinfeld. Hardness of approximate nearest neighbor search. In *STOC*, page To appear, 2018.
- 65 Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *Advances in Neural Information Processing Systems*, pages 2321–2329, 2014.
- 66 Anshumali Shrivastava and Ping Li. Asymmetric minwise hashing for indexing binary inner products and set containment. In *Proc. of the 24th International Conference on World Wide Web*, pages 981–991. ACM, 2015.
- 67 Christina Teflioudi and Rainer Gemulla. Exact and approximate maximum inner product search with lemp. *ACM Transactions on Database Systems (TODS)*, 42(1):5, 2016.
- 68 Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *Journal of the ACM (JACM)*, 62(2):13, 2015.
- 69 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *To appear in the proceedings of the ICM*, 2018.
- 70 R. Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2–3):357–365, 2005.
- 71 Ryan Williams. Faster all-pairs shortest paths via circuit complexity. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 664–673. ACM, 2014.
- 72 Ryan Williams. On the difference between closest, furthest, and orthogonal pairs: Nearly-linear vs barely-subquadratic complexity. 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 1207–1215. SIAM, 2018. doi:10.1137/1.9781611975031.78.
- 73 Ryan Williams and Huacheng Yu. Finding orthogonal vectors in discrete structures. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1867–1877. SIAM, 2014.
- 74 Andrew Chi-Chih Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.

## A A Dimensionality Reduction for Max-IP

In fact, tracing the proof of Theorem 4.1, we observe that it is possible to compute the inner product  $x \cdot y$  itself from  $\psi_{b,\ell}(x) \cdot \psi_{b,\ell}(y)$ , that is:

► **Corollary A.1.** *Let  $b, \ell$  be two sufficiently large integers. There is a reduction  $\psi_{b,\ell} : \{0, 1\}^{b \cdot \ell} \rightarrow \mathbb{Z}^\ell$  and  $b \cdot \ell + 1$  sets  $V_{b,\ell}^0, V_{b,\ell}^1, \dots, V_{b,\ell}^{b \cdot \ell} \subseteq \mathbb{Z}$ , such that for every  $x, y \in \{0, 1\}^{b \cdot \ell}$ ,*

$$x \cdot y = k \Leftrightarrow \psi_{b,\ell}(x) \cdot \psi_{b,\ell}(y) \in V_{b,\ell}^k \quad \text{for all } 0 \leq k \leq b \cdot \ell,$$

and

$$0 \leq \psi_{b,\ell}(x)_i < \ell^{6^{\log^*(b)} \cdot b}$$

for all possible  $x$  and  $i \in [\ell]$ . Moreover, the computation of  $\psi_{b,\ell}(x)$  takes  $\text{poly}(b \cdot \ell)$  time, and the sets  $V_{b,\ell}^k$ 's can be constructed in  $O\left(\ell^{O(6^{\log^*(b)} \cdot b)} \cdot \text{poly}(b \cdot \ell)\right)$  time.

Together with Theorem 4.3, it proves Corollary 4.4 (restated below).

**Reminder of Corollary 4.4** *Let  $1 \leq \ell \leq d$ . There is an*

$$O\left(n \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))} \cdot \text{poly}(d)\right)\text{-time}$$

reduction from  $\text{Max-IP}_{n,d}$  to  $d \cdot \ell^{O(6^{\log^* d} \cdot (d/\ell))}$  instances of  $\mathbb{Z}\text{-Max-IP}_{n,(\ell+1)^2}$ , with vectors of entries with bit-length  $O\left(d/\ell \cdot \log \ell \cdot 6^{\log^* d}\right)$ .

**Proof Sketch.** Let  $b = d/\ell$  (assume  $\ell$  divides  $d$  here for simplicity),  $A$  and  $B$  be the sets in the given  $\text{Max-IP}_{n,d}$  instance, we proceed similarly as the case for  $\text{OV}$ .

We first enumerate a number  $k$  from 0 to  $d$ , for each  $k$  we construct the set  $V_{b,\ell}^k$  as specified in Corollary A.1. Then there is  $(x, y) \in A \times B$  such that  $x \cdot y = k$  if and only if there is  $(x, y) \in A \times B$  such that  $\psi_{b,\ell}(x) \cdot \psi_{b,\ell}(y) \in V_{b,\ell}^k$ . Using exactly the same reduction as in Lemma 1.17, we can in turn reduce this into  $\ell^{O(6^{\log^*(b)} \cdot b)}$  instances of  $\mathbb{Z}\text{-OV}_{n,\ell+1}$ .

Applying Theorem 4.3, with evaluation of  $(d+1) \cdot \ell^{O(6^{\log^*(b)} \cdot b)}$   $\mathbb{Z}\text{-Max-IP}_{n,(\ell+1)^2}$  instances, we can determine whether there is  $(x, y) \in A \times B$  such that  $x \cdot y = k$  for every  $k$ , from which we can compute the answer to the  $\text{Max-IP}_{n,d}$  instance. ◀

## B Nonuniform to Uniform Transformation for Dimensionality Reduction for OV

In this section we discuss the transformation from nonuniform construction to uniform one for dimensionality reduction for  $\text{OV}$ . In order to state our result formally, we need to introduce some definitions.

► **Definition B.1** (Nonuniform Reduction). Let  $b, \ell, \kappa \in \mathbb{N}$ . We say a function  $\varphi : \{0, 1\}^{b \cdot \ell} \rightarrow \mathbb{Z}^\ell$  together with a set  $V \subseteq \mathbb{Z}$  is a  $(b, \ell, \kappa)$ -reduction, if the following holds:

- For every  $x, y \in \{0, 1\}^{b \cdot \ell}$ ,
 
$$x \cdot y = 0 \Leftrightarrow \varphi(x) \cdot \varphi(y) \in V.$$
- For every  $x$  and  $i \in [\ell]$ ,
 
$$0 \leq \varphi(x)_i < \ell^{\kappa \cdot b}.$$

Similarly, let  $\tau$  be an increasing function, we say a function family  $\{\varphi_{b,\ell}\}_{b,\ell}$  together with a set family  $\{V_{b,\ell}\}_{b,\ell}$  is a  $\tau$ -reduction family, if for every  $b$  and  $\ell$ ,  $(\varphi_{b,\ell}, V_{b,\ell})$  is a  $(b, \ell, \tau(b))$ -reduction.

Moreover, if for all  $b$  and all  $\ell \leq \log \log \log b$ , there is an algorithm  $\mathbb{A}$  which computes  $\varphi_{b,\ell}(x)$  in  $\text{poly}(b)$  time given  $b, \ell$  and  $x \in \{0, 1\}^{b \cdot \ell}$ , and constructs the set  $V_{b,\ell}$  in  $O\left(\ell^{O(\tau(b) \cdot b)} \cdot \text{poly}(b)\right)$  time given  $b$  and  $\ell$ , then we call  $(\varphi_{b,\ell}, V_{b,\ell})$  a uniform- $\tau$ -reduction family.

► **Remark B.2.** The reason we assume  $\ell$  to be small is that in our applications we only care about very small  $\ell$ , and that greatly simplifies the notation. From Theorem 4.1, there is a uniform- $(6^{\log^* b})$ -reduction family, and a better uniform-reduction family implies better hardness for  $\mathbb{Z}\text{-OV}$  and other related problems as well (Lemma 1.17, Theorem 4.3, Lemma 4.6 and Lemma 4.5).

Now we are ready to state our nonuniform to uniform transformation result formally.

► **Theorem B.3.** Letting  $\tau$  be an increasing function such that  $\tau(n) = O(\log \log \log n)$  and supposing there is a  $\tau$ -reduction family, then there is a uniform- $O(\tau)$ -reduction family.



**Proof Sketch.** The construction in Theorem 4.1 is recursive, it constructs the reduction  $\psi_{b,\ell}$  from a much smaller reduction  $\psi_{b_{\text{micro}},\ell}$ , where  $b_{\text{micro}} \leq \log b$ . In the original construction, it takes  $\log^* b$  recursions to make the problem sufficiently small so that a direct construction can be used. Here we only apply the reduction thrice. First let us abstract the following lemma from the proof of Theorem 4.1.

► **Lemma B.4** (Implicit in Theorem 4.1). *Letting  $b, \ell, b_{\text{micro}}, \kappa \in \mathbb{N}$  and supposing  $\ell^{\kappa \cdot b_{\text{micro}}} = b$  and there is a  $(b_{\text{micro}}, \ell, \kappa)$ -reduction  $(\varphi, V')$ , the following holds:*

- *There is a  $(b, \ell, 6 \cdot \kappa)$ -reduction  $(\psi, V)$ .*
- *Given  $(\varphi, V')$ , for all  $x \in \{0, 1\}^{b \cdot \ell}$ ,  $\psi(x)$  can be computed in  $\text{poly}(b \cdot \ell)$ , and  $V$  can be constructed in  $O(\ell^{O(\kappa \cdot b)} \cdot \text{poly}(b \cdot \ell))$  time.*

Now, let  $b, \ell \in \mathbb{N}$ , we are going to construct our reduction as follows.

Let  $b_1$  be the number such that

$$\ell^{\tau(b) \cdot 6^2 \cdot b_1} = b,$$

and similarly we set  $b_2$  and  $b_3$  so that

$$\ell^{\tau(b) \cdot 6 \cdot b_2} = b_1 \quad \text{and} \quad \ell^{\tau(b) \cdot b_3} = b_2.$$

We can calculate from above that  $b_3 \leq \log \log \log b$ .

From the assumption that there is a  $\tau$ -reduction, there is a  $(b_3, \ell, \tau(b_3))$ -reduction  $(\varphi_{b_3,\ell}, V_{b_3,\ell})$ , which is also a  $(b_3, \ell, \tau(b))$ -reduction, as  $\tau$  is increasing. Note that we can assume  $\ell \leq \log \log \log b$  and  $\tau(b) \leq \log \log \log b$  from assumption. Now we simply use a brute force algorithm to find  $(\varphi_{b_3,\ell}, V_{b_3,\ell})$ . There are

$$\ell^{\tau(b) \cdot b_3 \cdot \ell \cdot 2^{b_3 \cdot \ell}} = b^{o(1)}$$

possible functions from  $\{0, 1\}^{b_3 \cdot \ell} \rightarrow \{0, \dots, \ell^{\tau(b_3) \cdot b_3} - 1\}^\ell$ . Given such a function  $\varphi$ , one can check in  $\text{poly}(2^{b_3 \cdot \ell}) = b^{o(1)}$  time that whether one can construct a corresponding set  $V$  to obtain our  $(b_3, \ell, \tau(b))$ -reduction.

Applying Lemma B.4 thrice, one obtain a  $(b, \ell, O(\tau(b)))$ -reduction  $(\psi, V)$ . And since  $\varphi_{b_3,\ell}$  can be found in  $b^{o(1)}$  time, together with Lemma B.4, we obtain a uniform- $\tau$ -reduction family. ◀

Finally, we give a direct corollary of Theorem B.3 that the existence of an  $O(1)$ -reduction family implies hardness of  $\mathbb{Z}$ -OV,  $\mathbb{Z}$ -Max-IP,  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair in  $\omega(1)$  dimensions.

► **Corollary B.5.** *If there is an  $O(1)$ -reduction family, then for every  $\varepsilon > 0$ , there exists a  $c \geq 1$  such that  $\mathbb{Z}$ -OV,  $\mathbb{Z}$ -Max-IP,  $\ell_2$ -Furthest Pair and Bichromatic  $\ell_2$ -Closest Pair in  $c$  dimensions with  $O(\log n)$ -bit entries require  $n^{2-\varepsilon}$  time.*

**Proof Sketch.** Note that since its hardness implies the harnesses of other three, we only need to consider  $\mathbb{Z}$ -OV here.

From Theorem B.3 and the assumption, there exists a uniform- $O(1)$ -reduction. Proceeding similar as in Lemma 1.17 with the uniform- $O(1)$ -reduction, we obtain a better dimensionality self reduction from OV to  $\mathbb{Z}$ -OV. Then exactly the same argument as in Theorem 1.18 with different parameters gives us the lower bound required. ◀



## C Hardness of Approximate $\{-1,1\}$ -Max-IP via Approximate Polynomial for OR

We first show that making use of the  $O(\sqrt{n})$ -degree approximate polynomial for OR [27, 36], OV can be reduced to approximating  $\{-1,1\}$ -Max-IP.

► **Theorem C.1.** *Letting  $\varepsilon \in (0,1)$ , there is an algorithm reducing an  $OV_{n,d}$  instance with sets  $A, B$  to a  $\{-1,1\}$ -Max-IP $_{n,d_1}$  instance with sets  $\tilde{A}$  and  $\tilde{B}$ , such that:*

- $d_1 = \left( \leq O\left(\frac{d}{\sqrt{d \log 1/\varepsilon}}\right) \right)^3 \cdot 2^{O(\sqrt{d \log 1/\varepsilon})} \cdot \varepsilon^{-1}$ , in which the notation  $\binom{n}{\leq m}$  denotes  $\sum_{i=0}^m \binom{n}{i}$ .
- There is an integer  $T > \varepsilon^{-1}$  such that if there is an  $(a,b) \in A \times B$  such that  $a \cdot b = 0$ , then  $OPT(\tilde{A}, \tilde{B}) \geq T$ .
- Otherwise,  $|OPT(\tilde{A}, \tilde{B})| \leq T \cdot \varepsilon$ .
- Moreover, the reduction takes  $n \cdot \text{poly}(d_1)$  time.

We remark here that the above reduction fails to achieve a characterization: setting  $\varepsilon = 1/2$  and  $d = c \log n$  for an arbitrary constant  $c$ , we have  $d_1 = 2^{\tilde{O}(\sqrt{\log n})}$ , much larger than  $\log n$ . Another interesting difference between the above theorem and Lemma 3.3 (the reduction from OV to approximating Max-IP) is that Lemma 3.3 reduces one OV instance to many Max-IP instances, while the above reduction only reduces it to one  $\{-1,1\}$ -Max-IP instance.

### Proof of Theorem C.1.

**Construction and Analysis of Polynomial  $P_\varepsilon(z)$ .** By [27, 36], there is a polynomial

$P_\varepsilon : \{0,1\}^d \rightarrow \mathbb{R}$  such that:

- $P_\varepsilon$  is of degree  $D = O\left(\sqrt{d \log 1/\varepsilon}\right)$ .
- For every  $z \in \{0,1\}^d$ ,  $P_\varepsilon(z) \in [0,1]$ .
- Given  $z \in \{0,1\}^d$ , if  $\text{OR}(z) = 0$ , then  $P_\varepsilon(z) \geq 1 - \varepsilon$ , otherwise  $P_\varepsilon(z) \leq \varepsilon$ .
- $P_\varepsilon$  can be constructed in time polynomial in its description size.

Now, let us analyze  $P_\varepsilon$  further. For a set  $S \subseteq [d]$ , let  $\chi_S : \{0,1\}^d \rightarrow \mathbb{R}$  be  $\chi_S(z) := \prod_{i \in S} (-1)^{z_i}$ . Then we can write  $P_\varepsilon$  as:

$$P_\varepsilon := \sum_{S \subseteq [d], |S| \leq D} \chi_S \cdot \langle \chi_S, P_\varepsilon \rangle,$$

where  $\langle \chi_S, P_\varepsilon \rangle$  is the inner product of  $\chi_S$  and  $P_\varepsilon$ , defined as  $\langle \chi_S, P_\varepsilon \rangle := \mathbb{E}_{x \in \{0,1\}^d} \chi_S(x) \cdot P_\varepsilon(x)$ .

Let  $c_S = \langle \chi_S, P_\varepsilon \rangle$ , from the definition it is easy to see that  $c_S \in [-1,1]$ .

**Discretization of Polynomial  $P_\varepsilon$ .** Note that  $P_\varepsilon(z)$  has real coefficients, we need to turn it into another polynomial with integer coefficients first.

Let  $M := \binom{d}{\leq D}$ , consider the following polynomial  $\hat{P}_\varepsilon$ :

$$\hat{P}_\varepsilon := \sum_{S \subseteq [d], |S| \leq D} \lfloor c_S \cdot 2M/\varepsilon \rfloor \cdot \chi_S.$$

We can see that  $|\widehat{P}_\varepsilon(z)/(2M/\varepsilon) - P_\varepsilon(z)| \leq \varepsilon$  for every  $z \in \{0,1\}^d$ , and we let  $\hat{c}_S := \lfloor c_S \cdot M \cdot 2/\varepsilon \rfloor$  for convenience.

**Simplification of Polynomial  $\widehat{P}_\varepsilon$ .**  $\widehat{P}_\varepsilon(z)$  is expressed over the basis  $\chi_S$ 's, we need to turn it into a polynomial over standard basis.

For each  $S \subseteq [d]$ , consider  $\chi_S$ , it can also be written as:

$$\chi_S(z) = \prod_{i \in S} (-1)^{z_i} := \prod_{i \in S} (1 - 2z_i) = \sum_{T \subseteq S} (-2)^{|T|} z_T,$$

where  $z_T := \prod_{i \in T} z_i$ . Plugging it into the expression of  $\widehat{P}_\varepsilon$ , we have

$$\widehat{P}_\varepsilon(z) := \sum_{T \subseteq [d], |T| \leq D} \left( \sum_{S \subseteq [d], |S| \leq D, T \subseteq S} \hat{c}_S \right) \cdot (-2)^{|T|} z_T.$$

Set

$$\tilde{c}_T := \left( \sum_{S \subseteq [d], |S| \leq D, T \subseteq S} \hat{c}_S \right) \cdot (-2)^{|T|},$$

the above simplifies to

$$\widehat{P}_\varepsilon(z) := \sum_{T \subseteq [d], |T| \leq D} \tilde{c}_T \cdot z_T.$$

**Properties of Polynomial  $\widehat{P}_\varepsilon$ .** Let us summarize some properties of  $\widehat{P}_\varepsilon$  for now. First we need a bound on  $|\tilde{c}_T|$ , we can see  $|\hat{c}_S| \leq M \cdot 2/\varepsilon$ , and by a simple calculation we have

$$|\tilde{c}_T| \leq M^2 \cdot 2^D \cdot 2/\varepsilon.$$

Let  $B = M^2 \cdot 2^D \cdot 2/\varepsilon$  for convenience. For  $x, y \in \{0,1\}^d$ , consider  $\widehat{P}_\varepsilon(x, y) := \widehat{P}_\varepsilon(x_1 y_1, x_2 y_2, \dots, x_d y_d)$  (that is, plugging in  $z_i = x_i y_i$ ), we have

$$\widehat{P}_\varepsilon(x, y) := \sum_{T \subseteq [d], |T| \leq D} \tilde{c}_T \cdot x_T \cdot y_T,$$

where  $x_T := \prod_{i \in T} x_i$  and  $y_T$  is defined similarly. Moreover, we have

- If  $x \cdot y = 0$ , then  $\widehat{P}_\varepsilon(x, y) \geq (2M/\varepsilon) \cdot (1 - 2\varepsilon)$ .
- If  $x \cdot y \neq 0$ , then  $|\widehat{P}_\varepsilon(x, y)| \leq (2M/\varepsilon) \cdot 2\varepsilon$ .

**The Reduction.** Now, let us construct the reduction, we begin with some notations. For two vectors  $a, b$ , we use  $a \circ b$  to denote their concatenation. For a vector  $a$  and a real  $x$ , we use  $a \cdot x$  to denote the vector resulting from multiplying each coordinate of  $a$  by  $x$ . Let  $\text{sgn}(x)$  be the sign function that outputs 1 when  $x > 0$ ,  $-1$  when  $x < 0$ , and 0 when  $x = 0$ . For  $x \in \{-B, -B+1, \dots, B\}$ , we use  $e_x \in \{-1, 0, 1\}^B$  to denote the vector whose first  $|x|$  elements are  $\text{sgn}(x)$  and the rest are zeros. We also use  $\mathbf{1}$  to denote the all-1 vector with length  $B$ .

Let  $T_1, T_2, \dots, T_M$  be an enumeration of all subsets  $T \subseteq [d]$  such that  $|T| \leq D$ , we define

$$\varphi_x(x) := \circ_{i=1}^M (e_{\tilde{c}_{T_i}} \cdot x_{T_i}) \text{ and } \varphi_y(y) := \circ_{i=1}^M (\mathbf{1} \cdot y_{T_i}).$$

And we have

$$\varphi_x(x) \cdot \varphi_y(y) = \sum_{i=1}^M (e_{\tilde{c}_{T_i}} \cdot \mathbf{1}) \cdot (x_{T_i} \cdot y_{T_i}) = \sum_{i=1}^M \tilde{c}_{T_i} \cdot x_{T_i} \cdot y_{T_i} = \widehat{P}_\varepsilon(x, y).$$

To move from  $\{-1, 0, 1\}$  to  $\{-1, 1\}$ , we use the following carefully designed reductions  $\psi_x, \psi_y : \{-1, 0, 1\} \rightarrow \{-1, 1\}^2$ , such that

$$\psi_x(-1) = \psi_y(-1) = (-1, -1), \quad \psi_x(0) = (-1, 1),$$

$$\psi_y(0) := (1, -1), \quad \text{and} \quad \psi_x(1) = \psi_y(1) = (1, 1).$$

It is easy to check that for  $x, y \in \{-1, 0, 1\}$ , we have  $\psi_x(x) \cdot \psi_y(y) = 2 \cdot (x \cdot y)$ .

Hence, composing the above two reductions, we get our desired reductions  $\phi_x = \psi_x^{\otimes(B \cdot M)} \circ \varphi_x$  and  $\phi_y = \psi_y^{\otimes(B \cdot M)} \circ \varphi_y$  such that for  $x, y \in \{0, 1\}^d$ ,  $\phi_x(x), \phi_y(y) \in \{-1, 1\}^{2B \cdot M}$  and  $\phi_x(x) \cdot \phi_y(y) = 2 \cdot \widehat{P}_\varepsilon(x, y)$ .

Finally, given an  $\text{OV}_{n,d}$  instance with two sets  $A$  and  $B$ , we construct two sets  $\tilde{A}$  and  $\tilde{B}$ , such that  $\tilde{A}$  consists of all  $\phi_x(x)$ 's for  $x \in A$ , and  $\tilde{B}$  consists of all  $\phi_y(y)$ 's for  $y \in B$ .

Then we can see  $\tilde{A}$  and  $\tilde{B}$  consist of  $n$  vectors from  $\{-1, 1\}^{d_1}$ , where

$$d_1 = 2B \cdot M = M^3 \cdot 2^D \cdot 2/\varepsilon = \left( \leq O\left(\frac{d}{\sqrt{d \log 1/\varepsilon}}\right) \right)^3 \cdot 2^{O(\sqrt{d \log 1/\varepsilon})} \cdot \varepsilon^{-1}$$

as stated.

It is not hard to see the above reduction takes  $n \cdot \text{poly}(d_1)$  time. Moreover, if there is a  $(x, y) \in A \times B$  such that  $x \cdot y = 0$ , then  $\text{OPT}(\tilde{A}, \tilde{B}) \geq (4M/\varepsilon) \cdot (1 - 2\varepsilon)$ , otherwise,  $\text{OPT}(\tilde{A}, \tilde{B}) \leq (4M/\varepsilon) \cdot 2\varepsilon$ . Setting  $\varepsilon$  above to be  $1/3$  times the  $\varepsilon$  in the statement finishes the proof.  $\blacktriangleleft$

With Theorem C.1, we are ready to prove our hardness results on  $\{-1, 1\}$ -Max-IP.

**► Theorem C.2.** *Assume SETH (or OVC). Letting  $\alpha : \mathbb{N} \rightarrow \mathbb{R}$  be any function of  $n$  such that  $\alpha(n) = n^{o(1)}$ , there is another function  $\beta$  satisfying  $\beta(n) = n^{o(1)}$  and an integer  $T > \alpha$  ( $\beta$  and  $T$  depend on  $\alpha$ ), such that there is no  $n^{2-\Omega(1)}$ -time algorithm for  $\{-1, 1\}$ -Max-IP $_{n,\beta(n)}$  distinguishing the following two cases:*

- $\text{OPT}(A, B) \geq T$  ( $A$  and  $B$  are the sets in the  $\{-1, 1\}$ -Max-IP instance).
- $|\text{OPT}(A, B)| \leq T/\alpha(n)$ .

**Proof.** Letting  $\alpha = n^{o(1)}$  and  $k = \log \alpha / \log n$ , we have  $k = o(1)$ . Setting  $d = c \log n$  where  $c$  is an arbitrary constant and  $\varepsilon = \alpha^{-1}$  in Theorem C.1, we have that an  $\text{OV}_{c \log n}$  reduces to a certain  $\alpha(n)$ -approximation to a  $\{-1, 1\}$ -Max-IP $_{n,d_1}$  instance with sets  $A$  and  $B$ , where

$$d_1 = \left( \leq O(\sqrt{ck} \log n) \right)^3 \cdot 2^{O(\sqrt{ck} \log n)} \leq \left( \frac{\sqrt{c}}{\sqrt{k}} \right)^{O(\sqrt{ck} \log n)} \cdot 2^{O(\sqrt{ck} \log n)} = n^{O(\log(c/k) \cdot \sqrt{ck})}.$$

Now set  $\beta = n^{k^{1/3}}$  and  $T$  be the integer specified by Theorem C.1, since  $k = o(1)$ ,  $\beta = n^{o(1)}$ . Suppose otherwise there is an  $n^{2-\Omega(1)}$ -time algorithm for distinguishing whether  $\text{OPT}(A, B) \geq T$  or  $|\text{OPT}(A, B)| \leq T/\alpha(n)$ . Then for any constant  $c$ ,  $O(\log(c/k) \sqrt{ck}) \leq k^{1/3}$  for sufficiently large  $n$ , which means  $d_1 \leq \beta(n)$  for a sufficiently large  $n$ , and there is an  $n^{2-\Omega(1)}$ -time algorithm for  $\text{OV}_{c \log n}$  by Theorem C.1, contradiction to OVC.  $\blacktriangleleft$

**D A Proof of Lemma 3.3**

Finally, we present a proof of Lemma 3.3, which is implicit in [64].

We need the following efficient MA protocol for Set-Disjointness from [64], which is also used in [50].<sup>15</sup>

► **Lemma D.1** (Theorem 3.2 of [64]). *For every  $\alpha$  and  $m$ , there is an  $(m/\alpha, \log_2 m, \text{poly}(\alpha), 1/2)$ -efficient MA protocol for  $\text{DISJ}_m$ .*

We want to reduce the error probability while keeping the number of total random coins relatively low. To achieve this, we can use an expander graph (Theorem 2.7) to prove the following theorem.

► **Lemma D.2.** *For every  $\alpha$ ,  $m$  and  $\varepsilon < 1/2$ , there is an  $(m/\alpha, \log_2 m + O(\log \varepsilon^{-1}), \text{poly}(\alpha) \cdot \log \varepsilon^{-1}, \varepsilon)$ -efficient MA protocol for  $\text{DISJ}_m$ .*

**Proof.** Let  $c_1$  and  $\mathcal{F} : \{0, 1\}^{\log m + c_1 \cdot \log \varepsilon^{-1}} \rightarrow [m]^{c_1 \cdot \log \varepsilon^{-1}}$  be the corresponding constant and function as in Theorem 2.7, and let  $\Pi$  denote the  $(m/\alpha, \log_2 m, \text{poly}(\alpha), 1/2)$ -efficient MA protocol for  $\text{DISJ}_m$  in Lemma D.1. Set  $q = c_1 \cdot \log \varepsilon^{-1}$  and our new protocol  $\Pi_{\text{new}}$  works as follows:

- Merlin still sends the same advice to Alice as in  $\Pi$ .
- Alice and Bob jointly toss  $r = \log m + q$  coins to get a string  $w \in \{0, 1\}^r$ . Then we let  $w_1, w_2, \dots, w_q$  be the sequence corresponding to  $\mathcal{F}(w)$ , each of them can be interpreted as  $\log m$  bits.
- Bob sends Alice  $q$  messages, the  $i$ -th message  $m_i$  corresponds to Bob's message in  $\Pi$  when the random bits is  $w_i$ .
- After that, Alice decides whether to accept or not as follows:
  - If for every  $i \in [q]$ , Alice would accept Bob's message  $m_i$  with random bits  $w_i$  in  $\Pi$ , then Alice accepts.
  - Otherwise, Alice rejects.

It is easy to verify that the advice length, message length and number of random coins satisfy our requirements.

For the error probability, note that when these two sets are disjoint, the same advice in  $\Pi$  leads to acceptance of Alice. Otherwise, suppose the advice from Merlin is either wrong or these two sets are intersecting, then half of the random bits in  $\{0, 1\}^{\log m}$  leads to the rejection of Alice in  $\Pi$ . Hence, from Theorem 2.7, with probability at least  $1 - \varepsilon$ , at least one of the random bits  $w_i$ 's would lead to the rejection of Alice, which completes the proof. ◀

Finally, we prove Lemma 3.3 (restated below).

**Reminder of Lemma 3.3** *There is a universal constant  $c_1$  such that, for every integer  $c$ , reals  $\varepsilon \in (0, 1]$  and  $\tau \geq 2$ ,  $\text{OV}_{n, c \log n}$  can be reduced to  $n^\varepsilon \text{Max-IP}_{n, d}$  instances  $(A_i, B_i)$  for  $i \in [n^\varepsilon]$ , such that:*

- $d = \tau^{\text{poly}(c/\varepsilon)} \cdot \log n$ .
- Letting  $T = c \log n \cdot \tau^{c_1}$ , if there is  $a \in A$  and  $b \in B$  such that  $a \cdot b = 0$ , then there exists an  $i$  such that  $\text{OPT}(A_i, B_i) \geq T$ .
- Otherwise, for all  $i$  we must have  $\text{OPT}(A_i, B_i) \leq T/\tau$ .

<sup>15</sup>The protocol in [50] also works for the  $k$ -party number-in-hand model.

**Proof.** The reduction follows exactly the same as in [5], we recap here for completeness.

Set  $\alpha = c/\varepsilon$ ,  $m = c \cdot \log n$  and  $\varepsilon = 1/\tau$ , and let  $\Pi$  be the  $(m/\alpha, \log_2 m + O(\log \varepsilon^{-1}), \text{poly}(\alpha) \cdot \log \varepsilon^{-1}, \varepsilon)$ -efficient MA protocol for Set-Disjointness as in Lemma D.2.

Now, we first enumerate all of  $2^{m/\alpha} = 2^{\varepsilon \cdot \log n} = n^\varepsilon$  possible advice strings, and create an Max-IP instance for each of the advice strings.

For a fix advice  $\psi \in \{0, 1\}^{\varepsilon \cdot \log n}$ , we create an Max-IP instance with sets  $A_\psi$  and  $B_\psi$  as follows. We use  $a \circ b$  to denote the concatenation of the strings  $a$  and  $b$ .

Let  $r = \log_2 m + c_1 \cdot \log \varepsilon^{-1}$ , where  $c_1$  is the constant hidden in the big  $O$  notation in Lemma D.2, and  $\ell = \text{poly}(\alpha) \cdot \log \varepsilon^{-1}$ . Let  $m_1, m_2, \dots, m_{2^\ell}$  be an enumeration of all strings in  $\{0, 1\}^\ell$ .

- For each  $a \in A$ , and for each string  $w \in \{0, 1\}^r$ , we create a vector  $a^w \in \{0, 1\}^{2^\ell}$ , such that  $a_i^w$  indicates that given advice  $\psi$  and randomness  $w$ , whether Alice accepts message  $m_i$  or not (1 for acceptance, 0 for rejection). Let the concatenation of all these  $a^w$ 's be  $a_\psi$ . Then  $A_\psi$  is the set of all these  $a_\psi$ 's for  $a \in A$ .
- For each  $b \in B$ , and for each string  $w \in \{0, 1\}^r$ , we create a vector  $b^w \in \{0, 1\}^{2^\ell}$ , such that  $b_i^w = 1$  if Bob sends the message  $m_i$  given advice  $\psi$  and randomness  $w$ , and  $= 0$  otherwise. Let the concatenation of all these  $b^w$ 's be  $b_\psi$ . Then  $B_\psi$  is the set of all these  $b_\psi$ 's for  $b \in B$ .

We can see that for  $a \in A$  and  $b \in B$ ,  $a_\psi \cdot b_\psi$  is precisely the number of random coins leading Alice to accept the message from Bob given advice  $\psi$  when Alice and Bob holds  $a$  and  $b$  correspondingly. Therefore, let  $T = 2^r = c \log n \cdot \tau^{c_1}$ , from the properties of the protocol  $\Pi$ , we can see that:

- If there is  $a \in A$  and  $b \in B$  such that  $a \cdot b = 0$ , then there is  $\psi \in \{0, 1\}^{\varepsilon \cdot \log n}$  such that  $a_\psi \cdot b_\psi \geq T$ .
- Otherwise, for all  $a \in A$ ,  $b \in B$  and advice  $\psi \in \{0, 1\}^{\varepsilon \cdot \log n}$ ,  $a_\psi \cdot b_\psi \leq T/\tau$ .

And this completes the proof. ◀




# Hardness of Function Composition for Semantic Read once Branching Programs

**Jeff Edmonds**

York University, 4700 Keele Street, Toronto, CANADA


jeff@cse.yorku.ca

 <http://www.cs.yorku.ca/~jeff/>

**Venkatesh Medabalimi**

University of Toronto, 10 King's College Road, Toronto, CANADA


venkatm@cs.toronto.edu

 <https://www.cs.toronto.edu/~venkatm>

**Toniann Pitassi**

University of Toronto, 10 King's College Road, Toronto, CANADA, and Institute for Advanced Study, Princeton NJ

toni@cs.toronto.edu

 <https://www.cs.toronto.edu/~toni/>

---

## Abstract

In this work, we study time/space trade-offs for function composition. We prove asymptotically optimal lower bounds for function composition in the setting of *nondeterministic read once branching programs*, for the syntactic model as well as the stronger semantic model of read-once nondeterministic computation. We prove that such branching programs for solving the tree evaluation problem over an alphabet of size  $k$  requires size roughly  $k^{\Omega(h)}$ , i.e. space  $\Omega(h \log k)$ . Our lower bound nearly matches the natural upper bound which follows the best strategy for black-white pebbling the underlying tree. While previous super-polynomial lower bounds have been proven for read-once nondeterministic branching programs (for both the syntactic as well as the semantic models), we give the first lower bounds for iterated function composition, and in these models our lower bounds are near optimal.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Complexity classes

**Keywords and phrases** Branching Programs, Function Composition, Time-Space Tradeoffs, Semantic Read Once, Tree Evaluation Problem

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.15

**Funding** Research supported by NSERC

**Acknowledgements** We would like to thank Stephen A. Cook for many helpful discussions.

## 1 Introduction

One of the most promising approaches to proving major separations in complexity theory is to understand the complexity of function composition. Given two Boolean functions,  $f : \{0, 1\}^m \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^n \rightarrow \{0, 1\}$ , their composition is the function  $f \circ g : \{0, 1\}^{mn} \rightarrow \{0, 1\}$  defined by

$$(f \circ g)(x_1, \dots, x_m) = f(g(x_1), \dots, g(x_m)).$$



© Jeff Edmonds, Venkatesh Medabalimi, and Toniann Pitassi;

licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 15; pp. 15:1–15:22



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



The complexity of function composition is one of the most tantalizing and basic problems in complexity theory, and has been studied in a variety of models. There are very few settings where function composition can be computed with substantially less resources than first computing each instance of  $g$ , followed by computing  $f$  on the outputs of the  $g$ 's. Indeed, lower bounds for function composition are known to resolve several longstanding open problems in complexity theory.

The most famous conjecture about function composition in complexity theory is the Karchmer-Raz-Wigderson (KRW) conjecture [25], asserting that the minimum Boolean circuit depth for computing  $f \circ g$  for nontrivial functions  $f$  and  $g$  is the minimum depth of computing  $f$  plus the minimum depth of computing  $g$ . Karchmer, Raz and Wigderson show that repeated applications of this conjecture implies super-logarithmic lower bounds on the depth complexity of an explicit function, thus resolving a major open problem in complexity theory (separating  $P$  from  $NC^1$ ). In particular, The *tree evaluation problem* defines iterated function composition with parameters  $d$  and  $h$  as follows. The input is an ordered  $d$ -ary tree of depth  $h + 1$ . Each of the  $d^h$  leaf nodes of the tree is labelled with an input bit, and each non-leaf node of the tree is labelled by a  $2^d$  Boolean vector, which is the truth table of a Boolean function from  $\{0, 1\}^d \rightarrow \{0, 1\}$ . This induces a 0/1 value for each intermediate node in the tree in the natural way: for a node  $v$  with corresponding function  $f_v$ , we label  $v$  with  $f_v$  applied to bits that label the children of  $v$ . The output is the value of the root node. The basic idea is to apply  $h = O(\log n / \log \log n)$  compositions of a random  $d = \log n$ -ary function  $f : \{0, 1\}^{\log n} \rightarrow \{0, 1\}$  to obtain a new function over  $O(n^2)$  bits that is computable in polynomial time but that requires depth  $\Omega(\log^2 n)$  (ignoring lower order terms).

In communication complexity, lower bounds for function composition have been successful for solving several open problems. For example, *lifting theorems* in communication complexity reduce lower bounds in communication complexity to query complexity lower bounds, via function composition. Raz and McKenzie [33] proved a general lifting theorem for deterministic communication complexity, which implies a separation of  $NC^i$  from  $P$  for all  $i > 1$ . Subsequent lifting theorems (proving hardness of function composition for other communication models) have resolved open problems in game theory, proof complexity, extension complexity, and communication complexity [20, 8, 26, 28, 12].

The complexity of function composition for space-bounded computation has also been studied since the 1960's. The classical result of Nečiporuk [31] proves  $\Omega(n^2 / \log^2 n)$  size lower bounds for deterministic branching programs for function composition<sup>1</sup>. Subsequently, Pudlak observed that Nečiporuk's method can be extended to prove  $\Omega(n^{3/2} / \log n)$  size lower bounds for *nondeterministic branching programs*. These classical results are still the best unrestricted branching program size lower bounds known, and it is a longstanding open problem to break this barrier. Furthermore, it is known that Nečiporuk method cannot fetch lower bounds better than those mentioned above for both deterministic and non-deterministic branching programs [23, 4].

In this work, we study time/space tradeoffs for function composition. We prove asymptotically optimal lower bounds for function composition in the setting of *nondeterministic read once branching programs*, for the syntactic model as well as the stronger semantic model of read-once nondeterministic computation. We prove that such branching programs for solving

---

<sup>1</sup> While Nečiporuk's result is not usually stated this way, it can be seen as a lower bound for function composition. We present this alternative proof in section B in the Appendix.



the tree evaluation problem over an alphabet of size  $k$  requires size roughly  $k^{\Omega(h)}$ , i.e. space  $\Omega(h \log k)$ . Our lower bound nearly matches the natural upper bound which follows the best strategy for black-white pebbling [10] the underlying tree. While previous super-polynomial lower bounds have been proven for read-once nondeterministic branching programs (for both the syntactic as well as the semantic models), we give the first lower bounds for iterated function composition, and in these models our lower bounds are near optimal.

## 1.1 History and Related Work

### 1.1.1 Function Composition and Direct Sum Conjectures

Karchmer, Raz and Wigderson [25] resolved their conjecture in the context of monotone circuit depth. In an attempt to prove the conjecture in the non-monotone case, they proposed an intermediate conjecture, known as the universal relation composition conjecture. This intermediate conjecture was proven by Edmonds et.al [15] using novel information-theoretic techniques. More recently some important steps have been taken towards replacing the universal relation by a function using information complexity [19] and communication complexity techniques [14]. Dinur and Meir [14] prove a "composition theorem" for  $f \circ g$  where  $g$  is the parity function, and obtain an alternative proof of cubic formula size lower bounds as a corollary. The cubic formula size lower bound was originally proven by Håstad [34] and more recently by Tal [35].

### 1.1.2 Time-Space Tradeoffs

In the uniform setting, time-space tradeoffs for SAT were achieved in a series of papers [16, 29, 17, 18]. Fortnow-Lipton-Viglas-Van Melkebeek [18] shows that any algorithm for SAT running in space  $n^{o(1)}$  requires time at least  $\Omega(n^{\phi-\epsilon})$  where  $\phi$  is the golden ratio  $((\sqrt{5} + 1)/2)$  and  $\epsilon > 0$ . Subsequent works [36, 13] improved the time lower bound to greater than  $n^{1.759}$ .

The state of the art time/space tradeoffs for branching programs were proven in the remarkable papers by Ajtai [1] and Beame-et-al [3]. In the first paper, Ajtai exhibited a polynomial-time computable Boolean function such that any sub-exponential size deterministic branching program requires superlinear length. This result was significantly improved and extended by Beame-et-al who showed that any sub-exponential size randomized branching program requires length  $\Omega(n^{\frac{\log n}{\log \log n}})$ .

Lower bounds for nondeterministic branching programs have been more difficult to obtain. Length-restricted nondeterministic branching programs come in two flavors: *syntactic* and *semantic*. A length  $l$  syntactic model requires that every path in the branching program has length at most  $l$ , and similarly a read- $c$  syntactic model requires that every path in the branching program reads every variable at most  $c$  times. In the less restricted semantic model, the read- $c$  requirement is only for *consistent* accepting paths from the source to the 1-node; that is, accepting paths along which no two tests  $x_i = d_1$  and  $x_i = d_2$ ,  $d_1 \neq d_2$  are made. Thus for a nondeterministic read- $c$  semantic branching program, the overall length of the program can be unbounded.

Note that any syntactic read-once branching program is also a semantic read-once branching program, but the opposite direction does not hold. In fact, Jukna [22] proved that semantic read-once branching programs are exponentially more powerful than syntactic read-once branching programs, via the "Exact Perfect Matching"(EPM) problem. The input is a (Boolean) matrix  $A$ , and  $A$  is accepted if and only if every row and column of  $A$  has exactly one 1 and rest of the entries are 0's i.e. if it's a permutation matrix. Jukna gave a

polynomial-size semantic read-once branching program for EPM, while it was known that syntactic read-once branching programs require exponential size [27, 24].

Lower bounds for syntactic read- $c$  (nondeterministic) branching programs have been known for some time [32, 6]. However, for *semantic* nondeterministic branching programs, even for read-once, no lower bounds are known for polynomial time computable functions for the boolean,  $k = 2$  case. Nevertheless exponential lower bounds for semantic read- $c$  (nondeterministic)  $k$ -way branching programs, where  $k \geq 2^{3c+10}$  were shown by Jukna[21]. More recently [11] obtain exponential size lower bounds for semantic read-once nondeterministic branching programs for  $k = 3$ , leaving only the boolean case open. Liu [30] proved near optimal size lower bounds for *deterministic* read once branching programs for function composition.

The rest of the paper is organized as follows. In Section 2 we give the formal definitions, present the natural upper bound and state our main result. In Section 3 we give the intuition and proof outline. Sections 4,5 and 6 are devoted to individual parts of the proof.

## 2 Definitions and Statement of Results

► **Definition 1.** Let  $f : [k]^n \rightarrow \{0, 1\}$  be a boolean valued function whose input variables are  $x_1, \dots, x_n$  where  $x_i \in [k]$ . A  **$k$ -way nondeterministic branching program** for  $f$  is an acyclic directed graph  $G$  with a distinguished source node  $q_{start}$  and sink node (the accept node)  $q_{accept}$ . We refer to the nodes as *states*. Each non-sink state is labeled with some input variable  $x_i$ , and each edge directed out of a state is labelled with a value  $b \in [k]$  for  $x_i$ . For each input  $\vec{\xi} \in [k]^n$ , the branching program accepts  $\vec{\xi}$  if and only if there exists at least one path starting at  $q_{start}$  leading to the accepting state  $q_{accept}$ , and such that all labels along this path are consistent with  $\vec{\xi}$ . The *size* of a branching program is the number of states in the graph. A nondeterministic branching program is ***semantic read-once*** if for every path from  $q_{start}$  to  $q_{accept}$  that is consistent with some input, each variable occurs at most once along the path.

Syntactic read-once branching programs are a more restricted model where no path can read a variable more than once; in the semantic read-once case, variables may be read more than once, but each accepting path may only query each variable once.

► **Definition 2.** The (ternary) height  $h$  tree evaluation problem  $Tree_{\vec{F}}$ , has an underlying 3-ary tree of height  $h$  with  $n = 3^{h-1}$  leaves. Each leaf is labelled by a corresponding variable in  $x_1, \dots, x_n$ . (Note that a tree with a single node has height 1.) Each internal node  $v$  is labeled with a function  $F : [k]^3 \rightarrow [k]$ , where  $\vec{F}$  denotes the vector of these functions. The input  $\vec{\xi} \in [k]^n$  gives a value in  $[k]$  to the leaf variables  $\vec{x}$ . This induces a value for each internal node in the natural way, and the output  $Tree_{\vec{F}}(\vec{\xi})$  is the labeling of the root. In the boolean version, the input  $\vec{\xi}$  is accepted if and only if  $Tree_{\vec{F}}(\vec{\xi}) \in [k^{1-\epsilon}]$  where  $\epsilon \in (0, 1)$  is a parameter.

The most natural way to solve the tree evaluation problem is to evaluate the vertices of the tree, via a strategy that mimics the optimal black-white pebbling of the underlying tree. In the next section, we review this upper bound, and show that it corresponds to a nondeterministic semantic read-once branching program of size  $\Theta(k^{h+1})$ . Our main result gives a nearly matching lower bound (when  $k$  is sufficiently large compared to  $h$ ).

► **Theorem 3.** *For any  $h$ , and  $k$  sufficiently large ( $k > 2^{42h}$ ), there exists  $\epsilon$  and  $\vec{F}$  such that any  $k$ -ary nondeterministic semantic read-once branching program for  $Tree_{\vec{F}}$  requires size  $\Omega\left(\frac{k}{\log k}\right)^h$*

We prove the lower bound for the decision version of the tree evaluation problem, with  $\epsilon$  chosen to be  $\frac{9h}{\log k}$ . Secondly, we actually show (See appendix C) that the lower bound holds for almost all  $\vec{F}$ , whenever each  $F$  is independently chosen to be a random 4-invertible function:

► **Definition 4.** A function  $F : [k]^3 \rightarrow [k]$  is 4-invertible if whenever the output value and two of its inputs from  $\{a, b, c\}$  are known, then the third input can be determined up to a set of four values. That is, for each pair of values  $(a, b) \in [k]^2$ , the mapping  $F(a, b, *) : [k] \rightarrow [k]$  is at most 4-to-1, and likewise for pairs  $(b, c)$  and  $(a, c)$ .

We expect that the lower bound should still hold even if every function in  $\vec{F}$  is fixed to be a particular function with nice properties, although we are not able to prove this at present. In particular, we conjecture that the lower bound still holds where for every  $v$ ,  $F_v(a, b, c) = a^3 + b^3 + c^3$  over the field  $[k]$ . On the other hand, if we take an associative function such as  $F_v(a, b, c) = a^3 \cdot b^3 \cdot c^3$  again over the field  $[k]$ , then there is a very small branching program, since we can compute the root value by reading the elements one at a time and remembering the product so far. One thing that makes proving the lower bound difficult is not being able to properly isolate or take advantage of the differences between these functions over a finite field. For the rest of the paper, we will refer to nondeterministic semantic read-once branching programs as simply branching programs.

## 2.1 Black/White pebbling, A natural upper bound

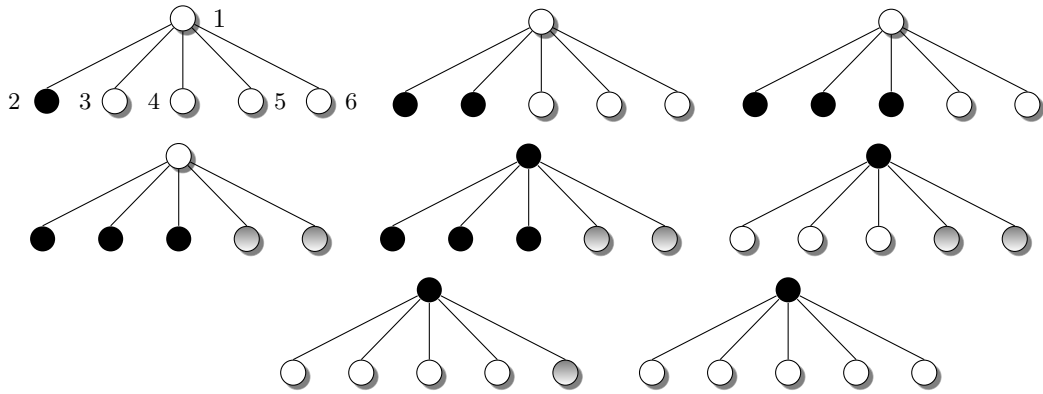
In order to get some intuition, we first review the matching upper bound. As mentioned earlier, the upper bound mimics the optimal black/white pebbling strategy for a tree [9]. A black pebble placement on a node  $v$  corresponds to remembering the value in  $[k]$  labelling that node, and a white pebble on  $v$  corresponds to nondeterministically guessing  $v$ 's value (which must later be verified.) The goal is to start with no pebbles on the tree, and end up with one black pebble on the root (and no other pebbles). The legal moves in a black/white pebbling game are:

1. A black pebble can be placed at any leaf.
2. If all children of node  $v$  are pebbled (black or white), place a black pebble at  $v$  and remove any black pebbles at the children. (When all children are pebbled, a black pebble on a child of  $v$  can be slid to  $v$ .)
3. Remove a black pebble at any time.
4. A white pebble can be placed at any node at any time.
5. A white pebble can be removed from  $v$  if  $v$  is a leaf or if all of  $v$ 's children are pebbled. (When all children but one are pebbled, the white pebble on  $v$  can be slid to the unpebbled child.)

► **Lemma 5.** *Black pebbling the root of a  $d$ -ary tree of height  $h$  can be done with  $(d-1)(h-1)+1$  pebbles. With both black and white pebbles, only  $\lceil \frac{1}{2}(d-1)h + 1 \rceil$  pebbles are needed.*

**Proof.** We will assume that  $d$  is odd; the case of  $d$  even is similar. With only black pebbles, recursively pebble  $d-1$  of the  $d$  children of the root. Then use  $d-1$  pebbles to remember these values as you use  $(d-1)(h-2)+1$  more pebbles to pebble its  $d^{\text{th}}$  child for a total of  $(d-1) + (d-1)(h-2) + 1 = (d-1)(h-1) + 1$  pebbles. Then pebble the root.

Now suppose white pebbles are also allowed (see Figure 1). Recursively pebble  $\frac{1}{2}(d-1)+1$  of the  $d$  children of the root. Then use  $\frac{1}{2}(d-1)$  pebbles to remember these values as you use  $\lceil \frac{1}{2}(d-1)(h-1) + 1 \rceil$  more pebbles to pebble its next child for a total of  $\frac{1}{2}(d-1) + \frac{1}{2}(d-$



■ **Figure 1** This figure describes a black/white pebbling for a  $d$ -ary tree  $T$  of height  $h$  at  $d=5$ . We start by pebbling the height  $h-1$  subtrees rooted at nodes 2,3 and 4. Then we proceed to the second half of children and guess the value that subtrees at node 5 and 6 would evaluate to. Now we can pebble the root node 1 and remove the black pebbles. The white pebble or guess at node 5 can now be verified and then the same is done subsequently for node 6.

$1)(h-1) + 1 = \frac{1}{2}(d-1)h + 1$  pebbles. Then use white pebbles to pebble the remaining  $\frac{1}{2}(d-1)$  children of the root. Pebble the root and pick up the black pebbles from the children. Replacing the first of these whites requires  $\frac{1}{2}(d-1)(h-1) + 1$  in addition to the  $\frac{1}{2}(d-1)$  white ones, again for a total of  $\frac{1}{2}(d-1)h + 1$ . Note as a base case, when  $h = 2$  and there is a root with  $d$  children,  $d$  pebbles are needed, no matter what the color. ◀

► **Lemma 6.** *A pebbling procedure with  $p$  black or white pebbles (and  $t$  time) translates to a layered nondeterministic branching program with  $tk^p$  states. If only black pebbles are used, the branching program is deterministic.*

**Proof.** On input  $\vec{\xi}$  the branching program moves through a sequence of states  $\beta_1, \beta_2, \dots, \beta_t$  where the state  $\beta_{t'}$  corresponds to the pebbling configuration at time  $t'$ . Each layer of the branching program will have  $k^p$  states one for each possible assignment of values in  $[k]$  to each of the pebbles. If a black pebble is placed on a leaf during the pebbling procedure, then the branching program queries this leaf. If all of the children of node  $v$  are pebbled, then the branching program knows their values  $v_1, v_2$  and  $v_3$  and hence can compute the value  $f_v(v_1, v_2, v_3)$  of the node. Remembering this new value corresponds to placing a black pebble at  $v$ . Removing a black pebble corresponds to the branching program forgetting this computed value. If a white pebble is placed at  $v$ , then the branching program nondeterministically guesses the required value for this node. This white pebble cannot be removed until this value has been verified to be  $f_v(v_1, v_2, v_3)$  using the values of its children that were either computed (black pebble) or also guessed (white). ◀

Observe that when we transform the black/white pebbling algorithm in Lemma 5 using the translation procedure presented in Lemma 6 we obtain a syntactic read once branching program.

### 3 Proof Overview

The crux of the proof is a compression argument, showing that from a small branching program, we can encode the information for a function label at a single special vertex of the ternary input tree more efficiently than is information-theoretically possible, thereby

obtaining a contradiction. We accomplish this by looking at the inputs read before and after any state  $q$  in the branching program on a particular accepting computation path, and finding one particular state  $q$  that has an associated "nice" collection of inputs. As in earlier papers, we prove that this nice collection of inputs forms an *embedded rectangle*. An embedded rectangle (formally defined in Definition 8) is a subset of inputs, all of which are accepted and all of which pass through a special state  $q$  in the branching program. These inputs form a combinatorial rectangle but where some of the input coordinates can be fixed. However, unlike earlier results, our embedded rectangle is required to have a very specific structure, in order to get a simple and short encoding of a function label. Below are more details about this specific structure and how we obtain it.

For each accepting input, we consider its accepting computation path in the branching program. This computation path,  $P$ , induces a permutation  $\Pi$  on the leaf variables of the ternary tree, defined by the order in which the leaf variables to the ternary tree are queried along the accepting computational path  $P$ . In Lemma 12 we prove that for each accepting input, there is a special state  $q$  along the computation path querying a special leaf variable  $l_q$ , such that many of the other leaf variables are read before  $q$  and many are read after  $q$  along this path. More specifically, let us visualize the ternary tree for this input, with the path from the root to the special leaf variable  $l_q$  going down the middle of the tree.<sup>2</sup> We show that the subtrees hanging off the left of this path (which we call the "red" subtrees) each contain many leaf variables that have been read before reaching state  $q$ , and the subtrees hanging off the right of this path (the "white" subtrees) each contain many leaf variables that are read after reaching state  $q$ .

Using Lemma 12, by averaging (over all accepting inputs, permutations and states), we prove in Lemma 9 that there exists an embedded rectangle with the following properties: we can find a single state  $q$  (which queries leaf variable  $l_q$ ), a single set of "red" leaf variables, and a single set of "white" leaf variables such that for a large collection of accepting inputs, they all pass through state  $q$ , and the set of red leaf variables are in one-to-one correspondence with the left subtrees, and the white leaf variables are in one-to-one corresponds with the right subtrees. (See Figure 2.)

From there, in Lemma 15, we further refine our embedded rectangle, by identifying a special internal node  $v_*$  in the ternary tree, such that we can encode the function  $F_{v_*}$  associated with  $v_*$  too succinctly. The reason we get compression is because the branching program is read-once, so the only way to transmit the information about the values of the red variables is via the state  $q$  we are passing through. Similarly the only way to nondeterministically guess information about the values of the white variables is also via the same state  $q$ . Since there are only  $s \ll k^h$  states, focusing on one particularly popular special node  $q$  amongst accepting inputs allows us to show that there is one node  $v_*$  in the ternary tree, that has a single red variable  $x$  (in the left subtree of  $v_*$ ) and a single white variable  $y$  (in the right subtree) that can each take on about  $r$  values.

If all of the internal functions  $\vec{F}$  of the ternary tree are invertible, then these  $r$  distinct values for the red variable  $x$  produce  $r$  distinct values as they propagate up the tree to  $v_*$ . Similarly the  $r$  distinct values for the white variable  $y$  produce this many distinct values as they propagate up the tree to  $v_*$ . Fixing the middle input to  $F_{v_*}$ , this gives rise to  $r^2$  distinct inputs to  $F_{v_*}$ : the left input to  $F_{v_*}$  runs over  $r$  distinct values (corresponding to the  $r$  values for  $x$  that propagate up the tree), and the right input runs over  $r$  distinct values

<sup>2</sup> Note that the *computation path*  $P$  is a sequence of states in the branching program, whereas a path in the ternary tree is defined on the *input tree*.

(corresponding to the  $r$  values for  $y$  that propagate up the tree). Since  $Tree_{\vec{F}}$  is a decision problem, each input is accepted if and only if the value of the root is in the restricted set  $[k^{1-\epsilon}]$ . Again if the internal functions are invertible, the size of this set would be retained as it propagates down the tree from the root to  $v_*$ . Thus the embedded rectangle enables us to encode the function  $F_{v_*}$  on these  $r^2$  inputs much more succinctly than should be possible as follows. First, the label  $L$  will specify the  $r^2$  special inputs to  $F_{v_*}$ . What is key about an  $r$ -by- $r$  square is that though its *area* consists of  $r^2$  values, the *length* of its two sides is only  $r \ll r^2$ . This allows us to specify  $L$  using only  $O(r \log k)$  bits. Secondly, the  $r^2$  output values of  $F_{v_*}$  on these  $r^2$  special inputs can be communicated with only  $r^2 \log(k^{1-\epsilon})$  bits instead of the usual  $r^2 \log(k)$  bits (since as we argued above, the output is restricted to a set of size  $k^{1-\epsilon}$  rather than to a set of size  $k$ .) The details of the compression argument are given in Section 6.

Some complications arise when trying to carry out the above proof outline, making the actual proof more intricate. First, the compression argument requires that each  $\vec{F}$  has a lot of accepting instances, so we need to show that most random  $\vec{F}$  have this property. The more serious complication is the fact that we cannot easily count over random invertible functions, so instead we use functions that are almost invertible. More specifically  $\vec{F}$  is a vector of 4-invertible functions which means that for each  $F \in \vec{F}$ , knowing two of the inputs to  $F$  and the output value, there are at most four consistent values for the third input. We use a novel argument that allows us to count over 4-invertible functions (Section 6). Our compression argument sketched above is then adapted to handle the case of 4-invertible functions with a small quantitative loss. Namely when going down the path  $P$  to determine the constraints on the output of  $F_{i_*}$  on an input  $(a_i, b_j, c_{i,j}) \in R$ , the number of allowable values for  $F_{i_*}(a_i, b_j, c_{i,j})$  will be  $k^{1-\epsilon}$  at the root vertex, and by 4-invertibility, we will gain a factor of four for each subsequent function along the path. Since the path height is very small relative to  $r$  this will still give us adequate compression.

#### 4 Most $\vec{F}$ have a lot of accepting instances

Let  $S_{yes} = \{\vec{\xi} \mid Tree_{\vec{F}}(\vec{\xi}) \in [k^{1-\epsilon}]\}$ . That is,  $S_{yes}$  is the set of accepting inputs to  $Tree_{\vec{F}}$ . Let  $Bad(\vec{F})$  be the event that the size of  $S_{yes}$  is significantly smaller than expected – in particular  $|S_{yes}| \leq \frac{1}{6k^\epsilon} \cdot k^n$ . Let  $\mathcal{F}$  be the uniform distribution over 4-invertible functions, and let  $\vec{\mathcal{F}}$  be the uniform distribution over vectors of 4-invertible functions (one for each non-leaf vertex in the tree). Lemma 7 proves that  $\Pr_{\vec{F}}[Bad(\vec{F})]$  is exponentially small, where  $\vec{F}$  is sampled from  $\vec{\mathcal{F}}$ .

► **Lemma 7.** For  $k > 2^{42h}$  and  $\epsilon = \frac{9h}{\log k}$ ,  $\Pr_{\vec{F}}[Bad(\vec{F})] \leq \frac{1}{10}$ .

See section A in the Appendix for the proof. The above probability is in fact much smaller but the above bound suffices for our purpose.

#### 5 Finding an Embedded Rectangle

This section proves that the accepted instances of  $Tree_{\vec{F}}$  solvable by a small branching program contain a large embedded rectangle whenever  $Bad(\vec{F})$  does not occur.

**Parameters.** The number of variables is  $n = 3^{h-1}$  and each variable is from  $[k]$ . In what follows we will fix  $r = \frac{2^{6h}}{\epsilon}$  and  $\epsilon = \frac{9h}{\log k}$ . The lower bound will hold for  $s \leq \left(\frac{k}{n^{26} \log k}\right)^h$ . For  $k$  sufficiently large ( $k > 2^{42h}$ ), the lower bound is  $\Omega(k/\log k)^h$ .



► **Definition 8.** For  $\pi \subset \{1, \dots, n\}$ , let  $x_\pi$  denote the set of variables  $\{x_i \mid i \in \pi\}$ . An *embedded rectangle* [2, 21] is defined by a 5-tuple  $(\pi_{red}, \pi_{white}, A, B, \vec{w})$ , where:

- (i)  $\pi_{red}, \pi_{white}$  are disjoint subsets of  $\{1, \dots, n\}$ ,
- (ii)  $A \subseteq [k]^{|\pi_{red}|}$  is a set of assignments to  $x_{\pi_{red}}$  and  $B \subseteq [k]^{|\pi_{white}|}$  is a set of assignments to  $x_{\pi_{white}}$ ;
- (iii)  $\vec{w} \in [k]^{n-|\pi_{red}|-|\pi_{white}|}$  is a fixed assignment to the remaining variables.

The assignments defined by the rectangle are all assignments  $(\vec{\alpha}, \vec{\beta}, \vec{w})$  where  $x_{\pi_{red}} = \vec{\alpha}$ ,  $x_{\pi_{white}} = \vec{\beta}$  and the rest of the variables are assigned  $\vec{w}$ , where  $\vec{\alpha} \in A$  and  $\vec{\beta} \in B$ .

## 5.1 Finding a rectangle over the leaves

In this section, we prove the following lemma, that shows the existence of a large embedded rectangle of accepting instances if the branching program solving  $Tree_{\vec{F}}$  is small.

► **Lemma 9.** *Let  $\mathcal{B}$  be a size  $s$  nondeterministic, semantic read-once BP over  $\{x_1, \dots, x_n\}$  solving  $Tree_{\vec{F}}$  for some  $\vec{F}$  such that  $\neg \text{Bad}(\vec{F})$  holds. Let  $s$  be chosen as above. Then there exists an embedded rectangle  $(\pi_{red}, \pi_{white}, A, B, \vec{w})$  such that:*

1.  $|\pi_{red}| = |\pi_{white}| = h$ ,
2.  $|A| \times |B| \geq \frac{k^{2h-\epsilon}}{s^{23h^2}}$ ,
3.  $\mathcal{B}$  accepts all inputs in the embedded rectangle.

In order to prove the above Lemma, we will need the following definitions.

► **Definition 10.** Let  $\vec{\xi}$  be an accepting input, and let  $Comp_{\vec{\xi}}$  be an accepting computation path for  $\vec{\xi}$ . Since every variable is read exactly once,  $Comp_{\vec{\xi}}$  defines a permutation  $\Pi$  of  $\{1, \dots, n\}$ . If  $q$  is a state that  $Comp_{\vec{\xi}}$  passes through at time  $t \in [n]$ , the pair  $(\Pi, q)$  partitions the variables  $x_1, \dots, x_n$  into two sets,  $Red(\Pi, q) = \{x_i \mid \Pi(i) \leq t\}$  and  $White(\Pi, q) = \{x_j \mid \Pi(j) > t\}$ . Intuitively, since the branching program reads the variables in the order given by  $\Pi$  (on input  $\vec{\xi}$ ), then  $Red(\Pi, q)$  are the variables that are read at or before reaching state  $q$ , and  $White(\Pi, q)$  are the variables that are read after reaching state  $q$ .

► **Definition 11.** A labelled path  $P$  down the ternary tree is a sequence of vertices  $v_h, \dots, v_1$  that forms a path from the root to a leaf of the ternary input tree. For each vertex  $v_j$  of height  $j$  along the path, its three subtrees are labelled as follows: one of its subtrees is labelled *red* and is referred to as  $Redtree(v_j)$ , another is labelled *white* and is referred to as  $Whitetree(v_j)$  and lastly,  $Thirddtree(v_j)$  refers to the subtree with root  $v_{j-1}$  that continues along the path  $P$ . The root of  $Redtree(v_j)$  will be called  $redchild(v_j)$ , the root of  $Whitetree(v_j)$  will be called  $whitechild(v_j)$ , and the root of  $Thirddtree(v_j)$  will be called  $thirdchild(v_j)$ .

► **Lemma 12.** *Let  $\vec{\xi}$  be an accepting input with computation path  $Comp_{\vec{\xi}}$ , where the ordering of variables read along  $Comp_{\vec{\xi}}$  is given by permutation  $\Pi$  of  $\{1, \dots, n\}$ . Then there exists a state  $q$  and a labelled path  $P = v_h, \dots, v_1$  in the ternary tree such that for all  $v_j$  in the path,  $2 \leq j \leq h$   $Redtree(v_j)$  contains greater than  $2^{j-2}$  variables in  $Red(\Pi, q)$  and  $Whitetree(v_j)$  contains greater than  $2^{j-2}$  variables in  $White(\Pi, q)$ .*

**Proof.** We will prove the above lemma by (downwards) induction on the path length. At step  $j$ ,  $2 \leq j \leq h$ , we will have constructed a labelled partial path  $v_h, v_{h-1}, \dots, v_j$ , an interval  $[t_0(j), t_1(j)]$ , and a partial coloring of the variables such that the following properties hold:

1. All variables  $x_i$  such that  $\Pi(x_i) \leq t_0(j)$  will be Red and all variables  $x_i$  such that  $\Pi(x_i) \geq t_1(j)$  will be White. (The remaining variables that are read between time step  $t_0(j)$  and  $t_1(j)$  are still uncolored.)

2. For each  $v_{j'}$ ,  $j < j' \leq h$ ,  $Redtree(v_{j'})$  contains greater than  $2^{j'-2}$  red variables, and  $Whitetree(v_{j'})$  contains greater than  $2^{j'-2}$  white variables.
3. The subtree of  $v_j$  that continues the path,  $Thirddtree(v_j)$ , has at most  $2^{j-2}$  red variables and at most  $2^{j-2}$  white variables.

While we construct our labelled path with the above properties it is worth mentioning that  $t_0(j) \leq t_1(j)$  always since all red variables come before white variables. Initially  $j = h$ , the path is empty,  $t_0[h] = 1$  and  $t_1[h] = n$ . Thus the size of the interval is  $n = 3^{h-1}$  and since no variables have been assigned to be red or white, the above properties trivially hold. For the inductive step, assume that we have constructed the partial path  $v_h, \dots, v_{j+1}$ . By the inductive hypothesis, the tree rooted at  $v_{j+1}$  contains at most  $2^{j-1}$  red variables and at most  $2^{j-1}$  white variables. Thus at most one subtree of  $v_{j+1}$  can contain greater than  $2^{j-2}$  red variables. If one subtree of  $v_{j+1}$  does contain greater than  $2^{j-2}$  red variables, then let this be  $Redtree(v_{j+1})$ . Otherwise, increase  $t_0[j+1]$  until one of  $v_{j+1}$ 's three subtrees contains (for the first time) more than  $2^{j-2}$  red variables and let this subtree be  $Redtree(v_{j+1})$ . Since each of  $v_{j+1}$ 's three subtrees has  $3^{j-1}$  leaves and at most  $2^{j-1}$  white variables, there are at least  $3^{j-1} - 2^{j-1} \geq 2^{j-2}$  variables remaining in each subtree that are either uncolored or colored red, and thus the process is well-defined.

Next we work with the remaining two subtrees of  $v_{j+1}$  in order to define  $Whitetree(v_{j+1})$ . Again by the inductive hypothesis, the tree rooted at  $v_{j+1}$  contains at most  $2^{j-1}$  white variables, and thus as most one subtree of the remaining two can contain greater than  $2^{j-2}$  white variables. If one is found, then designate it as  $Whitetree(v_{j+1})$ , and otherwise, decrease  $t_1[j+1]$  until one of  $v_{j+1}$ 's remaining two subtrees contains (for the first time)  $2^{j-2}$  white variables and designate it as  $Whitetree(v_{j+1})$ . Again since each subtree has  $3^{j-1}$  leaves and at most  $2^{j-1}$  red variables, there are at least  $3^{j-1} - 2^{j-1} \geq 2^{j-2}$  variables remaining in each of the two subtrees that are uncolored or colored white and thus the process is well-defined.

Let the remaining subtree of  $v_{j+1}$  be  $Thirddtree(v_{j+1})$  and let the next vertex  $v_j$  in our path be  $thirdchild(v_{j+1})$ . By construction  $Thirddtree(v_{j+1})$  contains at most  $2^{j-2}$  red variables and at most this same number of white variables. For the base case  $j = 2$ , by induction we will have reached a vertex  $v_2$  with 3 child vertices, where at most one is colored red and at most one is colored white and thus the size of the interval  $[t_0[2], t_1[2]]$  is between one and three. Increase  $t_0$  and then decrease  $t_1$  so that  $v_2$  has exactly one red vertex and two white vertices and let  $q$  be the state that  $Comp_{\vec{\xi}}$  passes through as it reads the red child. ◀

**Proof of Lemma 9.** Consider a nondeterministic semantic read-once branching program  $\mathcal{B}$  for  $Tree_{\vec{F}}$ . For each accepting input  $\vec{\xi}$ , fix one accepting path  $Comp_{\vec{\xi}}$  in the branching program. Each of the  $n$  variables must be read in this path exactly once, and thus it defines a permutation  $\Pi_{\vec{\xi}}$  of the  $n$  variables. Apply Lemma 12 for  $\vec{\xi}$  (and corresponding permutation  $\Pi_{\vec{\xi}}$ ) to obtain an associated labelled path  $P_{\vec{\xi}}$  and state  $q_{\vec{\xi}}$ . Do this for all accepting inputs, and pick the pair  $P, q$  that occurs the most frequently. There are at most  $s$  possible values for  $q$  and at most  $6^{h-1}$  possible labelled paths:  $n = 3^{h-1}$  ending leaves of the path and then for each of the  $h$  vertices  $v_{h'}$  along this path, we specify which of its subtrees are Red and White, for another  $2^{h-1}$  choices. Let  $S$  be those accepting inputs that give rise to the popular pair  $P, q$ . Since there are at least  $|S_{Yes}| > \frac{1}{6k^\epsilon} \cdot k^n$  accepting inputs,  $S$  is of size at least  $(\frac{1}{6^h s k^\epsilon}) k^n$ .

Next we will select one common red variable in each of the  $h$  Red subtrees, and one common white variable in each of the  $h$  White subtrees. Denoting the vertices of  $P$  by  $v_h, v_{h-1}, \dots, v_1$ , we will select the Red and White variables iteratively for  $j = h, h-1, \dots, 1$  as follows. Starting at  $Redtree(v_j)$ : for each  $\vec{\xi} \in S$ , by Lemma 12 at least  $2^{j-2}$  of its  $3^{j-1}$



variables are red, and thus there is one variable that is red in at least a  $\frac{2^{j-2}}{3^{j-1}}$  fraction of  $S$ . Choose this variable, and update  $S$  to contain only those inputs in  $S$  where this variable is red. (That is,  $\xi \in S$  will stay in  $S$  if and only if the variable is read by  $Comp_{\xi}$  before reaching state  $q$ .) Do the same thing for  $Whitetree(v_j)$ . At the end, we will have selected for each  $j$  one variable that is red in  $Redtree(v_j)$ , and one variable that is white in  $Whitetree(v_j)$ , and a set of inputs  $S$  such that all  $h$  of the selected red variables (one per subtree) are read before reaching  $q$  and all  $h$  of the selected white variables are read after reaching  $q$ . Let  $\pi_{red}$  be vector of  $h$  indices corresponding to these  $h$  red variables, where  $\pi_{red,j}$  is the index of the common red variable in  $Redtree(v_j)$ . and let  $\pi_{white}$  be the vector of  $h$  indices corresponding to these  $h$  white variables, where  $\pi_{white,j}$  is the index of the common white variable in  $Whitetree(v_j)$ . The size of  $S$  after this process will be reduced by a factor of

$$\prod_{j \in [2, \dots, h]} \left( \frac{2^{j-2}}{3^{j-1}} \right)^2 \geq 2^{-2h} \cdot 1.5^{-h^2}.$$

Our final pruning of  $S$  is to fix a partial assignment,  $\vec{w}$ , to the remaining  $n - 2h$  variables that have not been identified as red or white. There are  $k^{n-2h}$  choices here. Once again choose the most popular one. Overall, for  $h \geq 2$  this gives

$$|S| \geq \frac{1}{k^\epsilon 6^h 2^{2h} 1.5^{h^2} s k^{n-2h}} k^n \geq \frac{k^{2h-\epsilon}}{s 1.5^{h^2+8h}} \geq \frac{k^{2h-\epsilon}}{s 2^{3h^2}}.$$

Let  $S_{red} \subseteq [k]^{\pi_{red}}$  be the projection of  $S$  onto the coordinates of  $\pi_{red}$ , the red variables and let  $S_{white} \subseteq [k]^{\pi_{white}}$  be the projection of  $S$  onto the coordinates of  $\pi_{white}$ , the white variables. Let all the other variables be set according to the vector  $\vec{w}$ . It is clear that this gives an embedded rectangle,  $(\pi_{red}, \pi_{white}, S_{red}, S_{white}, \vec{w})$ . We want to show that all assignments in the rectangle are accepted by  $\mathcal{B}$ . To see this, consider an assignment  $\vec{\alpha}\vec{\beta}\vec{w}$  in the embedded rectangle, where  $\vec{\alpha} \in S_{red}$  is an assignment to  $x_{\pi_{red}}$ , and  $\vec{\beta} \in S_{white}$  is an assignment to  $x_{\pi_{white}}$ , and  $\vec{w}$  is an assignment to the remaining variables. By definition  $\vec{\alpha}$  is in the projection of  $S$  onto  $\pi_{red}$ , and thus there must be an assignment  $\vec{\alpha}'\vec{\beta}'\vec{w} \in S$ . Similarly, there must be an assignment  $\vec{\alpha}\vec{\beta}'\vec{w} \in S$ . Since these assignments are in  $S$ , the computation paths on each of them goes through  $q$ , and all variables  $x_{\pi_{red}}$  are read before reaching  $q$ , and all variables  $x_{\pi_{white}}$  are read after  $q$ . We want to show that  $\vec{\alpha}\vec{\beta}\vec{w}$  is also an accepting input (in  $S$ ). To see this, we follow the first half of the computation path of  $\vec{\alpha}'\vec{\beta}'\vec{w}$  until we reach  $q$ , and then we follow the second half of the computation path of  $\vec{\alpha}\vec{\beta}'\vec{w}$  after  $q$ . In this new spliced computation path, the variables  $x_{\pi_{red}}$  are all read (and have value  $\vec{\alpha}$ ) prior to reaching  $q$ , and the variables  $x_{\pi_{white}}$  are all read after reaching  $q$  (and have value  $\vec{\beta}$ ), and since all other variables have the same values on all paths, the new spliced computation path must be consistent and must be accepting. Therefore the input  $\vec{\alpha}\vec{\beta}\vec{w}$  is in  $S$  and is an accepting input. ◀

## 5.2 Refining the Rectangle

In this section, we refine the embedded rectangle given above, so that it will be a *square*  $r$ -by- $r$  rectangle.

► **Definition 13.** Let  $\mathcal{B}$  be a branching program for  $Tree_{\vec{F}}$  for some  $\vec{F}$  such that  $\neg Bad(\vec{F})$  holds, and let  $(\pi_{red}, \pi_{white}, S_{red}, S_{white}, \vec{w})$  be the embedded rectangle guaranteed by Lemma 9. We recall the notation/concepts from the proof of Lemma 9:

1. Let  $P = v_h, \dots, v_1$  be the common labelled path in the ternary tree, where  $Redtree(v_i)$ ,  $Whitetree(v_i)$  denotes the Red and White subtrees of  $v_i$ .

2. Let  $q$  be the common state in the branching program;
3. Let  $\pi_{red}, \pi_{white}$  be the indices of the red/white variables ( $h$  red variables altogether, one per Red subtree, and  $h$  white variables altogether, one per White subtree);
4. For all (accepting) inputs in the rectangle, all of the variables  $x_{\pi_{red}}$  are read before  $q$ , and all variables  $x_{\pi_{white}}$  are read after  $q$ .

We will now define a special kind of embedded rectangle that isolates a particular vertex  $v$  along the path  $P$  (which corresponds to a particular function  $F_v$ ).

► **Definition 14.** Let  $P = v_h, \dots, v_1$  be the labelled path in the ternary tree, and let  $r = 2^{6h}/\epsilon$ . Let  $v_{i^*}$  be a special vertex in the path  $P$ , where  $\pi_{red, i^*}$  is the index of the red variable in  $Redtree(v_{i^*})$ , and  $\pi_{white, i^*}$  is the index of the white variable in  $Whitetree(v_{i^*})$ . An embedded rectangle  $(\pi_{red}, \pi_{white}, A, B, \vec{w})$  is *special* for  $v_{i^*}$  if:

1.  $|A| = |B| = r$ ;
2. The projection of  $A$  onto  $x_{\pi_{red, i^*}}$  has size  $r$ , and the projection of  $B$  onto  $x_{\pi_{white, i^*}}$  has size  $r$ . In other words, no two elements of  $A$  agree on the value taken by  $x_{\pi_{red, i^*}}$  and likewise, no two elements of  $B$  agree on the value taken by  $x_{\pi_{white, i^*}}$ .

► **Lemma 15.** Let  $\mathcal{B}$  be a size  $s$  branching program for  $Tree_{\vec{F}}$  for some  $\vec{F}$  such that  $\neg \text{Bad}(\vec{F})$  holds. Then (for our choice of parameters) there is an  $i^* \in [h]$  and an embedded rectangle that is special for  $v_{i^*}$ .

**Proof.** Let  $\mathcal{B}$  be a size  $s$  branching program for  $Tree_{\vec{F}}$  and let  $(\pi_{red}, \pi_{white}, S_{red}, S_{white}, \vec{w})$  be the embedded rectangle guaranteed by Lemma 9. For each  $j \in [h]$ , call  $v_j$  *red-good* if  $|Proj(S_{red}, \pi_{red, j})| \geq r$ . That is,  $v_j$  is red-good if  $S_{red}$  projected to the red variable in  $Redtree(v_j)$  has size at least  $r$ . Similarly,  $j$  is white-good if  $|Proj(S_{white}, \pi_{white, j})| \geq r$ .

If there are  $l_{red}$  vertices that are red-good, then it is not hard to see that  $|S_{red}| \leq (r-1)^{h-l_{red}} k^{l_{red}}$ . To see this, every  $v_j$  that is not red-good can take on at most  $r-1$  values, and the red-good ones could take on at most  $k$  values. If we similarly define  $l_{white}$  to be the number of vertices that are white-good, then similarly we have,  $|S_{white}| \leq (r-1)^{h-l_{white}} k^{l_{white}}$ .

We want to show that there must exist an  $i^*$  such that  $v_{i^*}$  is both red-good and white-good. If not, then  $l_{red} + l_{white} \leq h$ , and therefore  $|S_{red} \times S_{white}| \leq (r-1)^h k^h < r^h k^h$ . But on the other hand, Lemma 9 dictates that  $|S_{red} \times S_{white}| \geq \frac{k^{2h-\epsilon}}{s^{2^{3h^2}}}$ . This is a contradiction since by our choice of parameters ( $r = 2^{6h}/\epsilon$ ,  $\epsilon = 9h/\log k$ ,  $s \leq \left(\frac{k}{n^{2^6} \log k}\right)^h$ ,  $n = 3^{h-1}$ ) we have:

$$\begin{aligned} \frac{k^{2h-\epsilon}}{s^{2^{3h^2}}} &\geq \frac{k^{2h-\epsilon}}{2^{3h^2}} \cdot \left(\frac{3^{26(h-1)} \log k}{k}\right)^h \geq k^{h-\epsilon} 2^{10h^2} (\log k)^h \\ &= k^h 2^{10h^2} \left(\frac{\log k}{2^9}\right)^h \quad \text{since } \epsilon = \frac{9h}{\log k}, \\ &= k^h \frac{2^{10h^2}}{2^{9h}} \left(\frac{2^h \log 9h}{\epsilon^h}\right) \geq \frac{k^h 2^{6h^2}}{\epsilon^h} = r^h k^h \quad \text{since } 4h + \log(9h) - 9 > 0, \forall h \geq 2 \end{aligned}$$

Let  $i^* \in [h]$  denote the index such that vertex  $v_{i^*}$  along the path  $P$  is both red-good and white-good. Thus  $Redtree(v_{i^*})$  contains the red variable indexed by  $\pi_{red, i^*}$ , and the projection of  $S_{red}$  to  $x_{\pi_{red, i^*}}$  has size at least  $r$ . Prune  $S_{red}$  to contain  $r$  assignments to  $x_{\pi_{red, i^*}}$ , where we have exactly one assignment for each of the  $r$  distinct values for  $x_{\pi_{red, i^*}}$ . In other words, while retaining  $r$  distinct assignments to  $x_{\pi_{red, i^*}}$  remove all but one of the assignments in  $S_{red}$  consistent with the value taken by  $x_{\pi_{red, i^*}}$ . Similarly,  $Whitetree(v_{i^*})$  contains the white variable indexed by  $\pi_{white, i^*}$ , and the projection of  $S_{white}$  to  $x_{\pi_{white, i^*}}$

has size at least  $r$ . Prune  $S_{white}$  to contain  $r$  assignments to  $x_{\pi_{white}}$ , where we have exactly one assignment for each of the  $r$  distinct values for  $x_{\pi_{white},i^*}$ . Because the pruned sets  $S_{red}$  and  $S_{white}$  will be important for our encoding, the following definition describes these sets more explicitly.

► **Definition 16.** The (pruned) assignments in  $S_{red}$  consist of  $r$  partial assignment to  $x_{\pi_{red}}$ . Each such assignment gives a distinct value for  $x_{\pi_{red},i^*}$ , with the values for the rest of the variables in  $x_{\pi_{red}}$  being completely determined by these. Let  $\vec{\alpha}_i, i \in [r]$  denote the partial assignments in  $S_{red}$ . That is, for each  $i \in [r]$ ,  $\vec{\alpha}_i = \alpha_i^1, \dots, \alpha_i^h$  is a vector of  $h$  values given to  $redchild(v_i)$  for all  $i \in [h]$ . Viewing the vectors  $\vec{\alpha}_i, i \in [r]$  as an  $r$ -by- $h$  matrix, the entries in column  $i^*$  ( $\alpha^{i^*}$ ) run over the  $r$  distinct values given to  $x_{\pi_{red}}$ . Similarly,  $S_{white}$  consists of  $r$  partial assignments to  $x_{\pi_{white}}$ . Let  $\vec{\beta}_i, i \in [r]$  denote the partial assignments in  $S_{white}$ . That is, for each  $i \in [r]$ ,  $\vec{\beta}_i = \beta_i^1, \dots, \beta_i^h$  is a vector of  $h$  values given to  $whitechild(v_i)$  for all  $i \in [h]$ . Viewed as an  $r$ -by- $h$  matrix, the entries in column  $i^*$  ( $\beta^{i^*}$ ) run over the  $r$  distinct values given to  $x_{\pi_{white}}$ .

It is clear from our construction that  $(\pi_{red}, \pi_{white}, S_{red}, S_{white}, \vec{w})$  is an embedded rectangle that is accepted by  $\mathcal{B}$  and that is special for  $v_{i^*}$ . ◀

## 6 The Encoding

In this section,  $\vec{F}$  is a vector of functions, one function each for each non-leaf vertex of the ternary tree, where each  $F$  in  $\vec{F}$  is a 4-invertible function from  $[k]^3$  to  $[k]$ . Let  $\mathcal{F}$  denote the uniform distribution on 4-invertible functions. Let  $H(\mathcal{F})$  refer to the entropy of  $\mathcal{F}$ . Assume that for each  $\vec{F}$  where every constituent function is 4-invertible, we have a size  $s$  branching program,  $\mathcal{B}_{\vec{F}}$  for  $Tree_{\vec{F}}$ .

Our goal is to communicate a random  $\vec{F}$  using less bits than is information-theoretically possible (under the assumption of a small branching program for  $Tree_{\vec{F}}$ ). If  $Bad(\vec{F})$  is true, then we simply communicate  $\vec{F}$  using the full  $H(\mathcal{F})$  bits that describe a uniformly random 4-invertible function at all the internal nodes of the tree. This requires  $H(\vec{F}) = (\text{number of internal nodes}) \times H(\mathcal{F})$  bits. If  $Bad(\vec{F})$  is false, using Lemma 15, from  $\mathcal{B}_{\vec{F}}$ , we will define a vector of information,  $L_{\vec{F}}$ , which we call a *label* that will allow us to encode  $\vec{F}$  with fewer bits than is possible on average to get a contradiction. The following lemma describes how one can come up with  $L_{\vec{F}}$ .

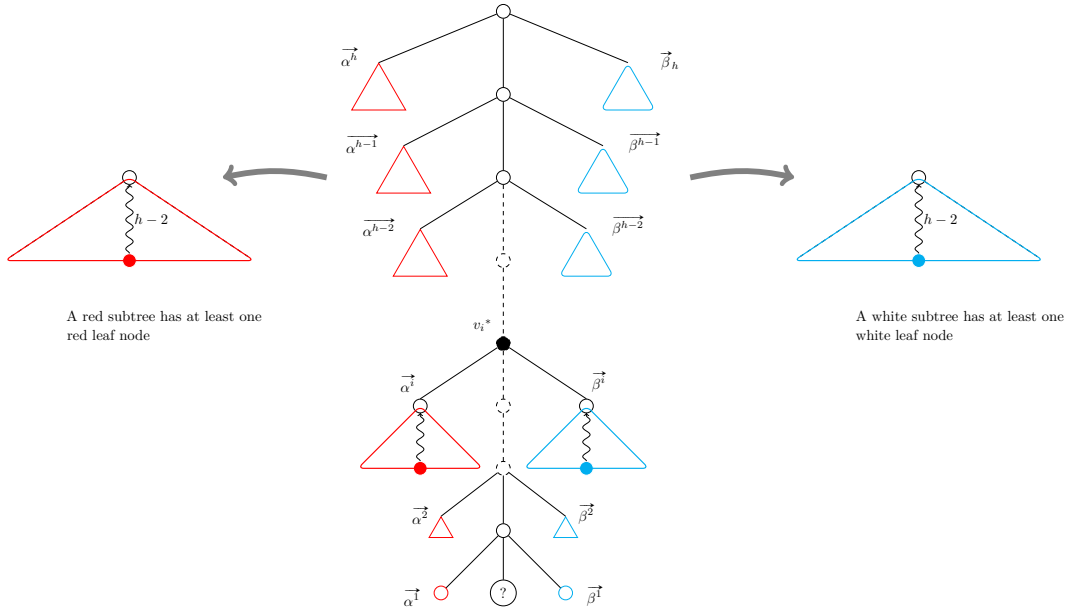
► **Lemma 17.** *Let  $\vec{F}$  be such that  $Bad(\vec{F})$  is false, and assume that  $Tree_{\vec{F}}$  has a small branching program  $\mathcal{B}_{\vec{F}}$ . Then there exists a vector  $L_{\vec{F}}$  that can be specified with at most  $4hr \log k = O(hr \log k)$  bits such that given  $\vec{F}_{-*}$ : the knowledge of all functions in  $\vec{F}$  except for  $F_*$  at one special node,  $L_{\vec{F}}$  can be used to infer  $r'^2$  inputs  $(a_i, b_j, c_{i,j}) \in [k]^3, i, j \in [r']$  in the domain of function  $F_*$ , where  $r' = \frac{r}{4^{i^*}}$  and  $i^*$  is the height of node of  $F_*$  and corresponding to these inputs one can infer  $r'^2$  sets of outputs  $C(i, j) \subset [k], i, j \in [r']$ , specifying a small set of values such that  $F_*(a_i, b_j, c_{i,j}) \in C(i, j)$ . Moreover,*

$$Pr_{F \sim \mathcal{F}}[\forall i, j \in [r'] F(a_i, b_j, c_{i,j}) \in C(i, j)] \leq k^{-\frac{7}{9 \cdot 2^{4h}} \epsilon r^2}.$$

**Proof.** By Lemmas 9 and 15, there is a path  $P$ , a vertex  $v_{i^*} \in P$  and an embedded rectangle  $(\pi_{red}, \pi_{white}, S_{red}, S_{white}, \vec{w})$  that is special for  $v_{i^*}$ .

The vector  $L_{\vec{F}}$  will consist of:

- (0) a description of  $\vec{w}$ ;
- (1) a description of the labelled path  $P$ ;



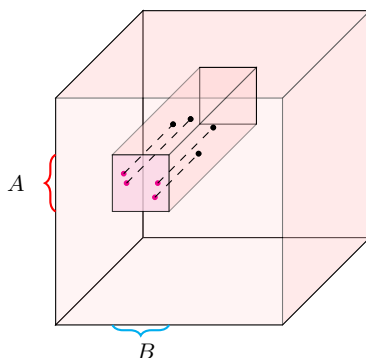
■ **Figure 2** This figure depicts a label  $L_{\vec{F}}$  associated with a problem instance  $Tree_{\vec{F}}$  obtained as a consequence of having a *small* branching program  $\mathcal{B}_{\vec{F}}$ . A label as guaranteed by lemma 17 consists of a labelled path  $P$  reaching a leaf node, a special vertex  $v_{i^*}$  along the path and a vector of  $r$  values each:  $\vec{\alpha}$  and  $\vec{\beta}$  respectively for the red and white sub trees at each node along the path. (We use blue for white here).

- (2) the index  $i^*$  of the special vertex along the path;
- (3) a vector  $\langle \vec{\alpha}_1, \dots, \vec{\alpha}_r \rangle$  of  $r$  assignments as described in Definition 16.
- (4) the vector  $\langle \vec{\beta}_1, \dots, \vec{\beta}_r \rangle$  of  $r$  assignments as described in Definition 16.

Figure 2 depicts a labelling that is induced by a small branching program. We first check that the length of  $L_{\vec{F}}$  is  $O(hr \log k)$ . The length of (0) is  $n \log k = 3^{h-1} \log k$ . The length of (1) is  $h \log 6$ , since there are  $6^h$  labelled paths ( $3^{h-1}$  different paths, and  $2^h$  choices for the labels). The length of (2) is  $\log h$ . The length of (3) is  $hr \log k$ , and similarly the length of (4) is  $hr \log k$ . Thus the total length is at most  $4hr \log k$ .

Given the vector  $L_{\vec{F}}$ , the special function  $F_*$  will be the function associated with the vertex  $v_{i^*}$ . For each  $i, j \in [r]$ , the corresponding input values  $(a_i, b_j, c_{i,j})$  for  $F_*$  are obtained by a bottom-up evaluation of the subtree rooted at  $v_{i^*}$  as follows. First, using  $L_{\vec{F}}$  parts (3) and (4) we extract values for all red and white children of vertices in the path below  $v_{i^*}$ . Secondly, using  $L_{\vec{F}}$  part (0) we extract from  $\vec{w}$  values for all other leaf vertices of the subtree rooted at  $v_{i^*}$ . Now using the knowledge of all internal functions corresponding to nodes below  $v_{i^*}$  (given in  $\vec{F}_{-i^*}$ ), we can evaluate the subtree rooted at  $v_{i^*}$  in a bottom-up fashion in order to determine the values  $(a_i, b_j, c_{i,j})$  for  $redchild(v_{i^*})$ ,  $whitechild(v_{i^*})$  and  $thirdchild(v_{i^*})$ . Clearly, the value  $c_{i,j}$  of  $thirdchild(v_{i^*})$  depends on both  $i, j$  since both red and white children appear downstream to this node unlike say  $redchild(v_{i^*})$  or  $whitechild(v_{i^*})$ .

Note that when we evaluate  $redchild(v_{i^*})$ ,  $whitechild(v_{i^*})$  and  $thirdchild(v_{i^*})$  for each pair of  $i, j \in [r]$  since all of the functions in  $\vec{F}$  are 4-invertible, we are guaranteed that there will be at least  $r' = \frac{r}{4^{i^*}}$  distinct values taken by  $redchild(v_{i^*})$  and similarly  $r' = \frac{r}{4^{i^*}}$  distinct values taken by  $whitechild(v_{i^*})$  resulting in at least  $r'^2$  distinct inputs  $(a_i, b_j, c_{i,j})$  with  $i, j \in [r']$  in the domain of  $F_*$ .



$$A, B \subset [k] \qquad |A| = |B| = r' \qquad |\{(a_i, b_j, c_{i,j}) | a_i \in A, b_j \in B\}| = r'^2$$

■ **Figure 3** A subset in the input domain of  $F_{v^*}$  with product structure in two coordinates and over which the possible values taken by  $F_{v^*}$  has low entropy.

We will now describe how to obtain the sets  $C(i, j) \subset [k]$ ,  $i, j \in [r']$ , using  $L_{\vec{F}}$  and the functions  $\vec{F}_{-*}$ . Fix an input  $(a_i, b_j, c_{i,j})$ . We want to determine the set  $C(i, j)$  of possible values for  $F_*(a_i, b_j, c_{i,j})$ . Recall that for each  $i, j \in [r']$ , we know the value given to all inputs of the ternary tree. We want to work our way down the path  $P$ , starting at the root vertex  $v_h$  in order to determine  $C(i, j)$ . If the functions in  $\vec{F}$  were all invertible, then knowing that  $(a_i, b_j, c_{i,j})$  is a yes input, this limits the number of possible values of the root vertex to the set  $C(i, j)^h = [k^{1-\epsilon}]$ . Working down the path, since we know the values of the red child and white child of  $v_h$ , this in turn gives us another set of at most  $k^{1-\epsilon}$  values,  $C(i, j)^{h-1}$  that  $v_{h-1}$  can have. We continue in this way down the path until we arrive at a set of at most  $k^{1-\epsilon}$  values,  $C(i, j)$  that  $v_{i^*}$  can take on.

However we are not working with invertible functions, but instead with 4-invertible functions. This can be handled by a simple modification of the above argument. Again we start at the root of the path  $v_h$ . As before, we know the values associated with the root is the set  $C(i, j)^h = [k^{1-\epsilon}]$ . At vertex  $v_{h'}$ , we define the set  $C(i, j)^{h'}$  based on the previous set  $C(i, j)^{h'+1}$ . For a particular value  $z \in C(i, j)^{h'+1}$ , we know the value of  $redchild(v_{h'})$ , and  $whitechild(v_{h'})$ . This gives us values  $z, a, b$ . By the definition of  $F_{v_{h'}}$  being 4-invertible, there are at most 4 values of  $c$  such that  $z = F_{v_{h'}}(a, b, c)$ . Thus we know the four possible values of  $c$  that can lead to  $z$  at  $a, b$ . Running over all  $z$ 's in  $C(i, j)^{h'+1}$  defines the set  $C(i, j)^{h'}$  which has size at most four times the size of  $C(i, j)^{h'+1}$ . Thus, the size of  $C(i, j)^{i^*}$  is at most  $4^{h-i^*} k^{1-\epsilon}$ . We set  $C(i, j)$  equal to  $C(i, j)^{i^*}$ .

Let  $\mathcal{F}$  be the uniform distribution over all 4-invertible functions from  $[k]^3$  to  $[k]$ . Let  $E$  denote the event that for every  $(i, j)$ ,  $F(a_i, b_j, c_{i,j}) \in C(i, j)$ . It is left to show that  $Pr_{F \sim \mathcal{F}}[E] \leq k^{-\frac{7}{9}\epsilon r^2 2^{-4h}}$ . Let  $\mathcal{F}'$  be the uniform distribution over all functions from  $[k]^3$  to  $[k/4]$ . Lemma 18 below shows that  $Pr_{F \sim \mathcal{F}}[E] \leq Pr_{F' \sim \mathcal{F}'}[E]$ . Thus we have:

$$Pr_{F \sim \mathcal{F}}[E] \leq Pr_{F' \sim \mathcal{F}'}[E] = \left( \frac{|C(i, j)|}{k/4} \right)^{(r')^2} \leq (4 \cdot 4^{h-i^*} \cdot k^{-\epsilon})^{(r/4^{i^*})^2} \leq k^{-\frac{7}{9 \cdot 2^{4h}} \epsilon r^2}. \quad \blacktriangleleft$$

**Proof.** (of Theorem 3) We are now ready to complete the proof of our main theorem. Let  $\vec{\mathcal{F}}$  be the uniform distribution over vectors  $\vec{F}$  of all 4-invertible functions from  $[k]^3$  to  $[k]$ . We prove the theorem by showing that if for every  $\vec{F}$ , if  $Tree_{\vec{F}}$  has a size  $s$  branching program where  $s \leq \left( \frac{k}{n^{26} \log k} \right)^h$ , then the expected number of bits required for encoding an  $\vec{F}$  sampled from the distribution  $\vec{\mathcal{F}}$  is less than the minimum number of bits required, which

is  $3^{h-1}H(\mathcal{F})$ , giving us the contradiction. Given  $\vec{F}$ , the encoding is as follows.

- (1) If  $\vec{F} \in \text{Bad}(\vec{F})$ , encode each function using  $H(\mathcal{F})$  bits, thus using  $3^{h-1}H(\mathcal{F})$  bits over all the internal functions.
- (2) If  $\vec{F} \notin \text{Bad}(\vec{F})$ , encode as follows.
  - (2a) The first part is the description of  $L_{\vec{F}}$ .
  - (2b) The second part is an optimal encoding of all of  $\vec{F}$  except for  $F_*$ .
  - (2c) The third part is an optimal encoding of  $F_*$ . Recall that  $F_*$  is an element from the (uniform) distribution  $(\mathcal{F} \mid E)$  where  $E$  denotes the event that for every  $(i, j)$ ,  $F(a_i, b_j, c_{i,j}) \in C(i, j)$ .

Using this encoding, the decoding procedure is as follows. Whenever  $\text{Bad}(\vec{F})$  holds, we use the information in (1) in order to recover  $\vec{F}$ . Otherwise, if  $\neg \text{Bad}(\vec{F})$  holds<sup>3</sup>, we proceed as follows. First we use the label  $L_{\vec{F}}$  from (2a) in order to determine  $v_{i^*}$ . Then we use label  $L_{\vec{F}}$  from (2a) along with information about the rest of the functions from (2b) to find the special  $(r')$ <sup>2</sup> inputs  $(a_i, b_j, c_{i,j})$ ,  $i, j \in [r']$  to the function  $F_*$ . We also use the label  $L_{\vec{F}}$  from (2a) and information from (2b) to determine the sets  $C(i, j) \subset [k]$  such that  $F_*(a_i, b_j, c_{i,j}) \in C(i, j)$  for all  $i, j \in [r']$ . We can then determine using the information from (2c) the values  $F_*(a_i, b_j, c_{i,j})$  for all  $i, j \in [r']$  (and also the remaining inputs in  $[k]$ <sup>3</sup>).

We want to compare the savings of this encoding over the optimal one that uses  $H(\vec{F})$  bits. Let  $p = \Pr_{F \sim \mathcal{F}}[E]$ . Then  $1/p$  is equal to the number of 4-invertible functions divided by the number of 4-invertible functions satisfying  $E$ . Thus, when  $\neg \text{Bad}(\vec{F})$  holds, the savings of our encoding in bits is  $\log(1/p) - |L_{\vec{F}}|$ , and therefore the overall savings in bits is

$$(1 - p_{\text{Bad}})[\log(1/p) - |L_{\vec{F}}|] \geq (1 - p_{\text{Bad}}) \left[ \frac{7}{9 \cdot 2^{4h}} \epsilon r^2 \log k - 4hr \log k \right] \\ = \left[ \frac{7}{9 \cdot 2^{4h}} \epsilon r^2 - 4hr \right] (1 - p_{\text{Bad}}) \log k$$

since by Lemma 17,  $|L_{\vec{F}}| \leq 4hr \log k$  and  $p \leq k^{-\frac{7}{9} \epsilon r^2 2^{-4h}}$ .

In the expression  $\left[ \frac{7}{9 \cdot 2^{4h}} \epsilon r^2 - 4hr \right]$ , the quadratic dependence on  $r$  in the first term whereas only a linear dependence in the second allows us to choose  $r = \frac{2^{6h}}{\epsilon}$ , large enough so that we make savings. At  $r = \frac{2^{6h}}{\epsilon}$ ,  $\left[ \frac{7}{9 \cdot 2^{4h}} \epsilon r^2 - 4hr \right] = r \left[ \frac{7}{9 \cdot 2^{4h}} 2^{6h} - 4h \right] > r \quad \forall h \geq 1$ . Also, by Lemma 7 we know  $p_{\text{Bad}} \leq \frac{1}{10}$  and since  $k \geq 2^{42h}$  this implies  $(1 - p_{\text{Bad}}) \log k > 1$ . Thus our savings is greater than  $r$  bits, giving a contradiction.  $\blacktriangleleft$

► **Lemma 18.** *Let  $\mathcal{F}$  be the uniform distribution over all 4-invertible functions from  $[k]^3$  to  $[k]$  and let  $\mathcal{F}'$  be the uniform distribution over all functions from  $[k]^3$  to  $[k/4]$ . Fix  $r^2$  inputs  $\tau_i$ ,  $i \in [r^2]$ , and let  $C_i$  be a corresponding subset of  $[k]$ , such that  $\cup_i C_i \subseteq [k/4]$ .  $E$  be the event that for all  $i$ ,  $F(\tau_i) \in C_i$ . Then  $\Pr_{F \sim \mathcal{F}}[E] \leq \Pr_{F' \sim \mathcal{F}'}[E]$ .*

**Proof.** Before we proceed with the proof, wish to mention that when we use this lemma in the proof of Lemma 17 the sets  $C_i$  involved need not be such that  $\cup_i C_i \subseteq [k/4]$ . However, since  $|\cup_i C_i| \leq k/4$ , one can simply consider an alternative range of size  $k/4$  that contains  $\cup_i C_i$  for functions in  $\mathcal{F}'$  instead of  $[k/4]$  to arrive at the same upperbound estimate on  $\Pr_{F \sim \mathcal{F}}[E]$ . So we assume here in the hypothesis just for the ease of exposition that  $\cup_i C_i \subseteq [k/4]$ . Proceeding with the proof, let  $E_i$  denote the event that  $F(\tau_i) \in C_i$ , and let  $E_{<i}$  denote the event that for all  $j < i$ ,  $F(\tau_j) \in C_j$ . Then  $\Pr_{F \sim \mathcal{F}}[E] = \prod_i \Pr_{F \sim \mathcal{F}}[E_i \mid E_{<i}]$ . We will show that for any  $i$ ,  $\Pr_{F \sim \mathcal{F}}[E_i \mid E_{<i}] \leq \Pr_{F' \sim \mathcal{F}'}[E_i]$ . Let  $\sigma$  specify the values of  $F$  for all tuples except for

<sup>3</sup> Astute reader might have observed that in order to recognize if  $\text{Bad}(\vec{F})$  holds or not one needs to convey information, albeit just 1 bit. We end up saving a lot more so we ignore it.

$\tau_i$ . Then  $\Pr_{F \sim \mathcal{F}}[E_i \mid E_{<i}] \leq \max_{\sigma} \Pr_{F \sim \mathcal{F}}[E_i \mid \sigma]$ . That is, the true probability is at most the probability where we fix all values except for the value of  $F$  on  $\tau_i$  to the worst possible scenario.

We want to show that this probability only increases when the distribution switches from  $\mathcal{F}$  to  $\mathcal{F}'$ . But then note that under the distribution  $\mathcal{F}'$ , the values  $\sigma$  do not change the probability. Thus we want to show:  $\Pr_{F \sim \mathcal{F}}[E_i \mid \sigma] \leq \Pr_{F' \sim \mathcal{F}'}[E_i \mid \sigma] \leq \Pr_{F' \sim \mathcal{F}'}[E_i]$ .

To prove the first inequality, note that  $\sigma$  specifies all but one of the  $[k]^3$  inputs to  $F$ . We visualize this as a  $k$ -by- $k$ -by- $k$  cube, where all entries  $(x, y, z)$  are filled in with a value in  $[k]$  except for the one entry corresponding to  $\tau_i$ . We want to get an upper bound on how many values we can choose for this last entry and still have a 4-invertible function. When choosing this last value, in order for  $F$  to be 4-invertible, we cannot choose one of the at most  $k/4$  values that already appears four times along the “x” dimension, or one of the at most  $k/4$  values that already appears four times in the “y” dimension, or  $k/4$  times in the “z” dimension. This rules out at most  $3k/4$  values, leaving at least  $k/4$  possible values. Thus there is a set of at least  $k/4$  values that can legally be filled in for  $F(\tau_i)$  (even under the worst possible  $\sigma$ ), and because  $\mathcal{F}$  is uniform on such functions, these completions all have the same probability. The event  $E_i$  is when  $F(\tau_i)$  is chosen to be in  $C_i$ . This probability is at most that for the distribution  $\mathcal{F}'$  on all functions from  $[k]^3$  to  $[k/4]$ . ◀

## 7 Conclusion

It is open to prove lower bounds for function composition for the case of Boolean non-deterministic semantic read-once branching programs. In fact, it is open to prove lower bounds for the Boolean case for any explicit function. Another longstanding open problem is to break the Nečiporuk barrier of  $n^2/\log^2 n$  for deterministic branching programs, and  $n^{3/2}/\log n$  for nondeterministic branching programs. When  $g$  is the parity function, this bound is optimal. Lower bounds for  $f \circ g$  for  $g$  equal to the element distinctness function (or even for the majority function) would be a significant breakthrough.

---

## References

- 1 M. Ajtai. A non-linear time lower bound for boolean branching programs. In *Proceedings 40th FOCS*, pages 60–70, 1999.
- 2 P. Beame, T.S. Jayram, and M. Saks. Time-space tradeoffs for branching programs. *J. Comput. Syst. Sci.*, 63(4):542–572, 2001.
- 3 P. Beame, M. Saks, X. Sun, and E. Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *Journal of the ACM*, 50(2):154–195, 2003.
- 4 Paul Beame, Nathan Grosshans, Pierre McKenzie, and Luc Segoufin. Nondeterminism and an abstract formulation of nečiporuk’s lower bound method. *ACM Transactions on Computation Theory*, 9, 08 2016.
- 5 Paul Beame and Pierre McKenzie. A note on neciporuk’s method for nondeterministic branching programs. *Manuscript*, August, 2011.
- 6 Allan Borodin, A Razborov, and Roman Smolensky. On lower bounds for read- $k$ -times branching programs. *Computational Complexity*, 3(1):1–18, 1993.
- 7 Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.
- 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 Stephen Cook, Pierre McKenzie, Dustin Wehr, Mark Braverman, and Rahul Santhanam. Pebbles and branching programs for tree evaluation. *ACM Transactions on Computation Theory (TOCT)*, 3(2):4, 2012.
- 10 Stephen Cook and Ravi Sethi. Storage requirements for deterministic polynomialtime recognizable languages. *Journal of Computer and System Sciences*, 13(1):25–37, 1976.
- 11 Stephen A. Cook, Jeff Edmonds, Venkatesh Medabalimi, and Toniann Pitassi. Lower bounds for nondeterministic semantic read-once branching programs. 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 36:1–36:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.ICALP.2016.36.
- 12 Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 295–304, 2016.
- 13 Scott Diehl and Dieter Van Melkebeek. Time-space lower bounds for the polynomial-time hierarchy on randomized machines. *SIAM Journal on Computing*, 36(3):563–594, 2006.
- 14 Irit Dinur and Or Meir. Toward the krw composition conjecture: Cubic formula lower bounds via communication complexity. In *LIPICs-Leibniz International Proceedings in Informatics*, volume 50. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- 15 Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiri Sgall. Communication complexity towards lower bounds on circuit depth. *Computational Complexity*, 10(3):210–246, 2001.
- 16 L. Fortnow. Nondeterministic polynomial time versus nondeterministic logarithmic space: Time space tradeoffs for satisfiability. In *Proceedings 12th Conference on Computational Complexity*, pages 52–60, 1997.
- 17 L. Fortnow and D. Van Melkebeek. Time-space tradeoffs for nondeterministic computation. In *Proceedings 15th Conference on Computational Complexity*, pages 2–13, 2000.
- 18 Lance Fortnow, Richard Lipton, Dieter Van Melkebeek, and Anastasios Viglas. Time-space lower bounds for satisfiability. *Journal of the ACM (JACM)*, 52(6):835–865, 2005.
- 19 Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: an information complexity approach to the krw composition conjecture. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 213–222. ACM, 2014.
- 20 Mika Göös. Lower bounds for clique vs. independent set. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1066–1076, 2015.
- 21 S. Jukna. A nondeterministic space-time tradeoff for linear codes. *Information Processing Letters*, 109(5):286–289, 2009.
- 22 Stasys Jukna. A note on read- $k$  times branching programs. *Informatique théorique et applications*, 29(1):75–83, 1995.
- 23 Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer Science & Business Media, 2012.
- 24 Stasys P Jukna. The effect of null-chains on the complexity of contact schemes. In *Fundamentals of Computation Theory*, pages 246–256. Springer, 1989.
- 25 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.
- 26 Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. In *Proceedings of the*



- 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017, pages 590–603, 2017.
- 27 Matthias Krause, Christoph Meinel, and Stephan Waack. Separating the eraser turing machine classes  $le$ ,  $nle$ ,  $co-nle$  and  $pe$ . In *Mathematical Foundations of Computer Science 1988*, pages 405–413. Springer, 1988.
  - 28 James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 567–576. ACM, 2015. doi: 10.1145/2746539.2746599.
  - 29 R. Lipton and A. Viglas. Time-space tradeoffs for sat. In *Proceedings 40th FOCS*, pages 459–464, 1999.
  - 30 David Liu. Pebbling arguments for tree evaluation. *CoRR*, abs/1311.0293, 2013. arXiv: 1311.0293.
  - 31 Edward I Nechiporuk. On a boolean function. *Doklady Akademii Nauk SSSR*, 169(4):765–+, 1966.
  - 32 EA Okolnishnikova. On lower bounds for branching programs. *Siberian Advances in Mathematics*, 3(1):152–166, 1993.
  - 33 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. doi: 10.1007/s004930050062.
  - 34 Johan Håstad. The shrinkage exponent of de morgan formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998.
  - 35 Avishay Tal. Shrinkage of de morgan formulae by spectral techniques. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 551–560. IEEE, 2014.
  - 36 Ryan Williams. Better time-space lower bounds for sat and related problems. In *Computational Complexity, 2005. Proceedings. Twentieth Annual IEEE Conference on*, pages 40–49. IEEE, 2005.

## A Proofs

Proof of lemma 7: For  $k > 2^{42h}$  and  $\epsilon = \frac{9h}{\log k}$ ,  $\Pr_{\vec{F}}[Bad(\vec{F})] \leq \frac{1}{10}$ .

**Proof.** We will choose a random  $\vec{F}$  somewhat indirectly as follows. First, we sample a random vector  $\vec{F} \in \vec{\mathcal{F}}$ . Secondly, we choose a random permutation  $\Pi$  of the values  $[k]$ , and let  $\Pi(\vec{F})$  be the same as  $\vec{F}$  except that the root values have been permuted by  $\Pi$ . (This requires only changing the outputs of the function at the root.) Note that this distribution on  $\vec{F}$  is identical to the uniform distribution over  $\vec{\mathcal{F}}$ . It follows that  $\Pr_{\vec{F}}[Bad(\vec{F})] = \Pr_{\langle \vec{F}, \Pi \rangle}[Bad(\Pi(\vec{F}))]$ . We will consider the worst case value of  $\vec{F}$  in order to bound the above probability. Observe that

$$\Pr_{\langle \vec{F}, \Pi \rangle} [Bad(\Pi(\vec{F}))] \leq \text{Max}_{\vec{F}} \Pr_{\Pi} [Bad(\Pi(\vec{F})) \mid \vec{F}].$$

Fix such a worst case  $\vec{F}$ . For this  $\vec{F}$ , for each value  $v \in [k]$  let  $q_v$  denote the fraction of leaf values  $\xi$  that give value  $v$  at the root. Note  $\sum_v q_v = 1$  and  $\text{Avg}_v q_v = \frac{1}{k}$ .

Because the permutation  $\Pi$  is randomly chosen,  $\Pi^{-1}([k^{1-\epsilon}])$  is a random subset of  $[k]$  of size  $k^{1-\epsilon}$ . Therefore via linearity of expectation,

$$\text{Exp} \left( \frac{|S_{yes}|}{|\{\xi\}|} \right) = \text{Exp} \left( \sum_{v \in \Pi^{-1}([k^{1-\epsilon}])} q_v \right) = \frac{k^{1-\epsilon}}{k} = k^{-\epsilon}.$$

We want to bound the probability that the size of  $S_{yes}$  is significantly smaller than its expected value of  $k^{1-\epsilon}$ . But first, the lemma below proves that  $0 \leq q_v \leq \frac{4^{h-1}}{k}$ .

► **Lemma 19.**  $\forall v \in [k], q_v \leq \frac{4^{h-1}}{k}$ .

**Proof.** Fix  $\vec{F}$ . Fix all of the leaf values as in  $\vec{\xi}$ , except for the left most leaf. Working down from the root, for any value  $v$  at the root one can see that there are at most  $4^{h-1}$  values in  $[k]$  for this left most leaf that can lead to value  $v$  at the root of  $\vec{F}$ . This is because each internal function is 4-invertible and for any fixed value of an internal node, given the value of two of its children(subtree evaluations) there are at most 4 possible values the other child can take. ◀

We select a uniformly random set of size  $k^{1-\epsilon}$  to be mapped to  $[k^{1-\epsilon}]$  as follows. Flip a biased coin for each point 'v' in  $[k]$  to be selected with probability  $k^{-\epsilon}$ . Given a vector of  $q_v$  describing the fraction of inputs that map to  $v$ , let  $Q_v$  be a vector of random variables associated with corresponding coin flips with each of them taking value  $q_v$  with probability  $k^{-\epsilon}$  and 0 with the remaining  $1 - k^{-\epsilon}$ . The expected number of points selected is  $k^{1-\epsilon}$ . The experiment repeats until the number of points selected is within some standard deviations say  $c \cdot k^{\frac{1-\epsilon}{2}}$  of the mean  $k^{1-\epsilon}$ . Let's first analyze the number of inputs selected corresponding to the points selected in the process without the size requirement on number of points.

We are interested in the fraction of inputs that get to be Yes inputs as a result of being selected during the coin flipping process. Let  $Q_{Yes} = \sum_v Q_v$ . So

$$E[Q_{Yes}] = \sum_v E[Q_v] = \sum_v q_v k^{-\epsilon} = k^{-\epsilon}. \quad (1)$$

In this experiment  $Q_v$  are independent (but not necessarily identically distributed) non-negative random variables. Consequently  $Q_{Yes}$  obeys the following concentration bound[7] around its mean

$$Prob [ (E[Q_{Yes}] - \sum_v Q_v) \geq t ] \leq e^{\left( \frac{-t^2}{2 \sum_v E[Q_v^2]} \right)} \quad (2)$$

Since by the regularity property from Lemma 19 we have  $q_v \leq \frac{4^{h-1}}{k}$  for all  $v \in [k]$

$$\begin{aligned} \sum_v E[Q_v^2] &= \sum_v q_v^2 k^{-\epsilon} = k^{-\epsilon} \sum_v q_v^2 \leq k^{-\epsilon} \sum_v \left( \frac{4^{h-1}}{k} \right)^2 = k^{-\epsilon} k \cdot \left( \frac{4^{h-1}}{k} \right)^2 = \frac{4^{2h-2}}{k^{1+\epsilon}} \\ \implies Prob [ (E[Q_{Yes}] - Q_{Yes}) \geq t ] &\leq e^{\left( \frac{-t^2}{2 \sum_v E[Q_v^2]} \right)} \leq e^{\left( \frac{-t^2}{2 \left( \frac{4^{2h-2}}{k^{1+\epsilon}} \right)} \right)} = e^{\frac{-t^2 k^{1+\epsilon}}{2 \cdot 4^{2h-2}}} \end{aligned}$$

Consequently,

$$Prob [Q_{Yes} \leq E[Q_{Yes}] - t] \leq e^{\frac{-t^2 k^{1+\epsilon}}{2 \cdot 4^{2h-2}}} \quad (3)$$

Set  $t = \frac{1}{2k^\epsilon}$  for the event  $Bad' = [Q_{Yes} \leq E[Q_{Yes}] - t] = [Q_{Yes} \leq \frac{1}{2k^\epsilon}]$ .

$$p_{Bad'} = Prob \left[ Q_{Yes} \leq \frac{1}{2k^\epsilon} \right] \leq e^{\frac{-k^{1-\epsilon}}{8 \cdot 4^{2h-2}}} \quad (4)$$

Now consider the following transformed process in which the experiment repeats until number of points selected is within some fixed deviation  $g$  from the mean. Let the set of points be  $A$ . Depending on the count of number of points in  $A$  selected, if the count falls

below  $k^{1-\epsilon}$  a few more points are uniformly randomly selected from  $[k] \setminus A$  to obtain a set of size  $k^{1-\epsilon}$  and likewise if the number is larger than  $k^{1-\epsilon}$  the required number of points are uniformly randomly discarded from the set. Clearly, this process doesn't discriminate against any point in  $[k]$  and so generates a uniformly random subset of size exactly  $k^{1-\epsilon}$  from  $[k]$ . Let call this set  $A''$ , it shall be our final set of size  $k^{1-\epsilon}$ . Let  $p_{Bad}$  be the probability that the fraction of inputs associated with the set of points in  $A''$  is less than  $\frac{1}{6k^\epsilon}$ . For the intermediate set  $A$  let  $U$  be the event  $[k^{1-\epsilon} - g \leq |A| \leq k^{1-\epsilon} + g]$ . Then,

$$Prob[Bad' | U] = \frac{Prob(Bad' \cap U)}{Prob(U)} \leq \frac{Prob(Bad')}{Prob(U)} \tag{5}$$

Since  $|A|$  is binomially distributed with  $(n, p) = (k, k^{-\epsilon})$ , seen as a sum of independent non-negative random variables, for a deviation  $g \approx 2k^{\frac{1-\epsilon}{2}}$  we have the following concentration guaranteed by (2)

$$Prob(U) = Prob\left[k^{1-\epsilon} - 2k^{\frac{1-\epsilon}{2}} \leq |A| \leq k^{1-\epsilon} + 2k^{\frac{1-\epsilon}{2}}\right] \geq 0.8 \tag{6}$$

By (4) it follows that  $Prob(Bad') \leq e^{\frac{-k^{1-\epsilon}}{8.4^{2h-2}}}$  and together with (6) and (5) this implies

$$Prob[Bad' | U] \leq \frac{5}{4} e^{\frac{-k^{1-\epsilon}}{8.4^{2h-2}}} \tag{7}$$

the chance that  $S_{Y_{es}}^A$  is small is exponentially small. Now consider the transformation of  $A$  to  $A''$ . Note that whenever new points are added to  $A$  or some points in  $A$  are discarded so as to obtain  $A''$  i.e a uniformly random choice of a set of exact size  $k^{1-\epsilon}$  the change from  $S_{Y_{es}}^A$  to  $S_{Y_{es}}^{A''}$  is at most  $g \cdot \max_v q_v$ . But by regularity property given by Lemma 19,  $q_v \leq \frac{4^h}{k}$ . So  $\left| |S_{Y_{es}}^{A''}| - |S_{Y_{es}}^A| \right| \leq g \cdot \frac{4^h}{k} \approx k^{\frac{1-\epsilon}{2}} \frac{4^h}{k} = \frac{4^h}{k^{\frac{1-\epsilon}{2} + \epsilon}} = \left( \frac{4^h}{k^{\frac{1-\epsilon}{2}}} \right) \frac{1}{k^\epsilon} \leq \frac{1}{3k^\epsilon}$  for  $k > 2^{42h}$  at  $\epsilon = \frac{9h}{\log k}$ . The resulting set  $A''$  will then always have size at least  $\frac{1}{2k^\epsilon} - \frac{1}{3k^\epsilon} = \frac{1}{6k^\epsilon}$  whenever  $Q_{Y_{es}}^A > \frac{1}{2k^\epsilon}$ . This implies  $p_{Bad} = Prob\left[Q_{Y_{es}}^{A''} \leq \frac{1}{6k^\epsilon}\right] \leq Prob\left[Q_{Y_{es}}^A \leq \frac{1}{2k^\epsilon}\right] = Prob[Bad' | U]$  and hence  $\leq \frac{5}{4} e^{\frac{-k^{1-\epsilon}}{8.4^{2h-2}}}$ .

For  $k > 2^{42h}$  and  $\epsilon = \frac{9h}{\log k}$  it can be seen that  $p_{Bad} \leq \frac{5}{4} e^{\frac{-k^{1-\epsilon}}{8.4^{2h-2}}} \leq \frac{5}{4} \left( \frac{1}{e} \right)^{\frac{2^{42h-9h}}{2^{4h-1}}} \leq \frac{1}{2^{28h}} \leq \frac{1}{10}, \forall h \geq 1$ . ◀

## B Nečiporuk via Function Composition

Consider the composition of two boolean functions  $f : \{0, 1\}^a \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^b \rightarrow \{0, 1\}$ . Let  $f$  be a hard function in the sense that any non-deterministic branching program computing  $f$  requires size at least  $2^{a/2}$ . Such functions are guaranteed to exist by a simple counting argument. Fix  $g$  to be any function such that it does not take a constant value when all but any one of its  $b$  input bits are set.

► **Lemma 20.** *Any non-deterministic branching program solving  $f \circ g$  has size at least  $b2^{a/2}$ .*

**Proof.** Let there be a non-deterministic branching program solving  $f \circ g$  of size  $s$ . For each of the  $a$  copies of  $g$  in the composition  $f \circ g$  pick the least queried input bit from amongst each group of  $b$  input bits that correspond to a single copy of  $g$ , then set all remaining  $b - 1$  variables in this input group to any value and reconnect the outgoing edges amongst the remaining states appropriately. The resulting collapsed branching program has size at most  $\frac{s}{b}$ . But recall that  $g$  has the property that fixing  $b - 1$  of its input bits doesn't make the

function a constant. Thus the resulting collapsed branching program has to have size at least that required for computing  $f$ , that is  $2^{a/2}$ . Therefore the original non-deterministic branching program must have size at least  $s \geq b2^{a/2}$ .  $\blacktriangleleft$

Let  $g = \oplus$  be the parity function on  $b$  bits. The input to  $f \circ \oplus$  is the description of  $f$ , plus a vector of  $ab$  bits (the input to  $f \circ \oplus$ ). The input length is  $2^a + ab$ . Setting  $a = \log n$  and  $b = \frac{n}{\log n}$ , the input length is  $2n$ . By the above lemma, the size of a branching program required to solve the composition  $f \circ \oplus$  is at least  $b2^{a/2} = \left(\frac{n}{\log n}\right) \left(2^{\frac{\log n}{2}}\right) = \frac{n^{3/2}}{\log n}$ . This lower bound is also known to be the best achievable by Nečiporuk as shown by Beame and McKenzie in [5].

By essentially similar means an  $\Omega\left(\frac{n^2}{\log^2 n}\right)$  lower bound can be shown for deterministic branching programs. Consider deterministic branching programs solving  $f \circ g$  where  $f$  is now a hard function in the sense that any deterministic branching program computing  $f$  requires size at least  $\frac{2^a}{a}$  (once again such functions are guaranteed to exist by counting argument). Just as before, fix  $g = \oplus$  to be the parity function (or any function that is not constant when all but any one of its input bits are set.) A similar argument shows that any branching program solving  $f \circ \oplus$  requires size at least  $b\frac{2^a}{a}$ . Set  $a = \log n$  and  $b = \frac{n}{\log n}$  to obtain an  $\Omega\left(\frac{n^2}{\log^2 n}\right)$  lower bound.

### **C** The lower bound holds for most $\vec{F}$

We now argue that for most vectors of 4-invertible functions  $\vec{F}$ ,  $Tree_{\vec{F}}$  does not have a small branching program. We show that the probability that a uniformly randomly chosen  $\vec{F}$  has a small branching program is at most  $p_{Bad} + \frac{1}{2^r} \leq \frac{1}{2^{27n}}$ . First, let  $\#L = 2^{|L_{\vec{F}}|}$  be the total number of labels. Recall that  $|L_{\vec{F}}|$  is the number of bits needed to encode a label and that the number of bits saved in our alternate encoding from the proof of Theorem 3 is  $(1 - p_{Bad})[\log(1/p) - |L_{\vec{F}}|] = (1 - p_{Bad}) \log\left(\frac{1}{p \cdot \#L}\right)$ .

Note that for a uniformly randomly chosen  $\vec{F}$  the probability that it has a small branching program is at most the chance that  $Bad(\vec{F})$  holds plus the chance that  $Bad(\vec{F})$  doesn't hold and there exists a label  $L$  that is consistent with  $\vec{F}$  (in other words a label obtained via lemma 17 as a guaranteed consequence of  $\vec{F}$  having a small branching program).

$$\begin{aligned}
 & \Pr_{\vec{F}}[\exists \text{ a small BP solving } Tree_{\vec{F}}] \\
 & \leq \Pr_{\vec{F}}[Bad(\vec{F}) \cup [\neg Bad(\vec{F}) \cap \exists \text{ a label } L \text{ consistent with } \vec{F}]] \\
 & \leq p_{Bad} + \Pr_{\vec{F}}[\neg Bad(\vec{F}) \cap \exists \text{ a label } L \text{ that is consistent with } \vec{F}] \quad (\text{by Union bound}) \\
 & \leq p_{Bad} + \Pr_{\vec{F}}[\exists \text{ a label } L \text{ that is consistent with } \vec{F}] \quad (\text{since } P(A \cap B) \leq \min\{P(A), P(B)\}) \\
 & \leq p_{Bad} + \#L \cdot \max_L \Pr_{\vec{F}}[\text{label } L \text{ is consistent with } \vec{F}] \quad (\text{by Union bound}) \\
 & \leq p_{Bad} + p \cdot \#L
 \end{aligned}$$

We have shown in the proof of theorem 3 that the number of bits saved in our alternate encoding is at least  $r$ . So,

$$(1 - p_{Bad}) \log\left(\frac{1}{p \cdot \#L}\right) \geq r \implies \frac{1}{p \cdot \#L} \geq 2^{r/(1-p_{Bad})} \geq 2^r \implies p \cdot \#L \leq \frac{1}{2^r}.$$

Consequently it follows that:

$$\Pr_{\vec{F}}[\exists \text{ a small BP solving } Tree_{\vec{F}}] \leq p_{Bad} + \frac{1}{2^r}$$

Now note that the proof of Lemma 7 (see Appendix A) actually shows that  $p_{Bad} \leq 2^{-28h}$ .

As a result,  $\Pr_{\vec{F}}[\exists \text{ a small BP solving } Tree_{\vec{F}}] \leq \frac{1}{2^{28n}} + \frac{1}{2^r} \leq \frac{1}{2^{27n}}$ . (the last inequality follows since  $r = \frac{2^{6h}}{\epsilon} = \frac{2^{6h} \log k}{9h} \geq 2^{6h+2}$ ). Thus we can conclude that most vectors of 4-invertible functions in fact do not have small branching programs.

# Reordering Rule Makes OBDD Proof Systems Stronger

**Sam Buss**

University of California, San Diego, La Jolla, CA, USA  
sbuss@ucsd.edu

**Dmitry Itsykson**

St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences, St. Petersburg, Russia  
dmitrits@pdmi.ras.ru

**Alexander Knop**

University of California, San Diego, La Jolla, CA, USA  
St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences, St. Petersburg, Russia  
aknop@ucsd.edu

**Dmitry Sokolov**

KTH Royal Institute of Technology, Stockholm, Sweden  
St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences, St. Petersburg, Russia  
sokolovd@kth.se

---

## Abstract

Atserias, Kolaitis, and Vardi showed that the proof system of Ordered Binary Decision Diagrams with conjunction and weakening,  $\text{OBDD}(\wedge, \text{weakening})$ , simulates  $\text{CP}^*$  (Cutting Planes with unary coefficients). We show that  $\text{OBDD}(\wedge, \text{weakening})$  can give exponentially shorter proofs than dag-like cutting planes. This is proved by showing that the Clique-Coloring tautologies have polynomial size proofs in the  $\text{OBDD}(\wedge, \text{weakening})$  system.

The reordering rule allows changing the variable order for OBDDs. We show that  $\text{OBDD}(\wedge, \text{weakening}, \text{reordering})$  is strictly stronger than  $\text{OBDD}(\wedge, \text{weakening})$ . This is proved using the Clique-Coloring tautologies, and by transforming tautologies using coded permutations and orification. We also give CNF formulas which have polynomial size  $\text{OBDD}(\wedge)$  proofs but require superpolynomial (actually, quasipolynomial size) resolution proofs, and thus we partially resolve an open question proposed by Groote and Zantema.

Applying dag-like and tree-like lifting techniques to the mentioned results, we completely analyze which of the systems among  $\text{CP}^*$ ,  $\text{OBDD}(\wedge)$ ,  $\text{OBDD}(\wedge, \text{reordering})$ ,  $\text{OBDD}(\wedge, \text{weakening})$  and  $\text{OBDD}(\wedge, \text{weakening}, \text{reordering})$  polynomially simulate each other. For dag-like proof systems, some of our separations are quasipolynomial and some are exponential; for tree-like systems, all of our separations are exponential.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Computational complexity and cryptography

**Keywords and phrases** Proof complexity, OBDD, Tseitin formulas, the Clique-Coloring principle, lifting theorems

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.16

**Funding** The research was supported by the Russian Science Foundation (project 16-11-10123)



© Sam Buss, Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov;

licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 16; pp. 16:1–16:24



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



## 1 Introduction

An Ordered Binary Decision Diagram (OBDD) is a branching program such that variables are queried in the same order on every path from the source to a sink. OBDDs were defined by Bryant [3] and have been shown to be useful in a variety of domains, such as hardware verification, model checking, and other CAD applications [4, 15]. Perhaps their most important property is that it is possible to carry out operations on OBDDs efficiently, including Boolean operations, projection, and testing satisfiability.

OBDDs have been used for several approaches to SAT-solving [17, 22]. The first such algorithms [22] worked by computing an OBDD for bigger and bigger subformulas of the input formula until obtaining an OBDD for the entire input formula, and then testing the resulting OBDD for satisfiability. A more attractive algorithm, called symbolic quantifier elimination, was proposed by Pan and Vardi [17]. Symbolic quantifier elimination loads clauses of the input formula into the current OBDD one by one and applies projection by a variables which do not appear in the remaining clauses. In contrast with DPLL algorithms, symbolic quantifier elimination can solve Tseitin formulas [11] and the pigeonhole principle [6] in polynomial time.

Atserias-Kolaitis-Vardi [1] defined a proof system based on OBDDs for proving unsatisfiability of CNFs, which is now called  $\text{OBDD}(\wedge, \text{weakening})$ . An  $\text{OBDD}(\wedge, \text{weakening})$  proof is a sequence of  $\pi$ -OBDDs with the ordering  $\pi$  of the variables held fixed. The initial lines are  $\pi$ -OBDDs expressing the input clauses; the final line is the constant false. Each step of the proof applies one of the two rules:

**Join (or  $\wedge$ ):** A conjunction of any two previously derived  $\pi$ -OBDDs is inferred;

**Weakening:** A  $\pi$ -OBDD is inferred that is semantically implied by some earlier derived  $\pi$ -OBDD.

The correctness of a proof step can be checked in polynomial time; in particular, checking if  $D_1$  is a weakening of  $D_2$  can be done by verifying that  $D_2 \wedge \neg D_1$  is unsatisfiable.

The paper [1] showed that Cutting Planes with unary coefficients ( $\text{CP}^*$ ) is simulated by  $\text{OBDD}(\wedge, \text{weakening})$ . This was proved by showing that any linear inequality has a short  $\pi$ -OBDD representation (under any ordering  $\pi$ ) and that addition of two inequalities may be simulated by join and weakening. Hence,  $\text{OBDD}(\wedge, \text{weakening})$  is strictly stronger than resolution; however, Segerlind [19] showed that tree-like  $\text{OBDD}(\wedge, \text{weakening})$  does not simulate (dag-like) resolution. Additionally, [1] showed that any unsatisfiable system of linear equation modulo two has a short refutation in  $\text{OBDD}(\wedge, \text{weakening})$ , while it is open, whether linear systems have short CP refutations. It is still open whether CP is strictly stronger than  $\text{CP}^*$ , and correspondingly it is open whether  $\text{OBDD}(\wedge, \text{weakening})$  simulates CP.

Krajíček [14] proved the first exponential lower bound for  $\text{OBDD}(\wedge, \text{weakening})$ . His lower bound consisted of two parts.

1. If a function  $f$  is computed by a  $\pi$ -OBDD  $D$ , the communication complexity of  $f$  under a partition  $\Pi_0, \Pi_1$  of the variables where the variables in  $\Pi_0$  precede (in the sense of  $\pi$ ) the variables from  $\Pi_1$  is at most  $\lceil \log |D| \rceil + 1$ . Since every proof system that operates with proof lines with small communication complexity admits monotone feasible interpolation [13], there is an ordering  $\pi$  of the variables so that any  $\pi$ - $\text{OBDD}(\wedge, \text{weakening})$  proof of the Clique-Coloring principle has exponential size. (This was already proven by Atserias et al. [1]).
2. Formulas which are hard for  $\text{OBDD}(\wedge, \text{weakening})$  in *some* order can be transformed into formulas that are hard for  $\text{OBDD}(\wedge, \text{weakening})$  in *all* orders. This transformation behaves well for constant width formulas.

In the paper we use another transformation due to Segerlind [19]; we use it to prove Lemma 1 and Theorem 10. This transformation behaves well for formulas which grow polynomially under “orification”.

Theorem 8, proved in Section 6, gives short (polynomial size)  $\text{OBDD}(\wedge, \text{weakening})$  proofs of the Clique-Coloring principle. Since any CP proof of the Clique-Coloring principle has exponential size [18], it follows that CP does not simulate  $\text{OBDD}(\wedge, \text{weakening})$  and moreover, that  $\text{OBDD}(\wedge, \text{weakening})$  is strictly stronger than  $\text{CP}^*$ . The existence of the small proofs of the Clique-Coloring principle implies that  $\text{OBDD}(\wedge, \text{weakening})$  does not have the feasible interpolation property. This is very curious, because the monotone feasible interpolation property nonetheless helps to prove lower bounds for this system.

Our short proofs of the Clique-Coloring principles are based on Grigoriev et. al [9], who gave short proofs of Clique-Coloring in  $\text{LS}^4$ , a proof system that uses inequalities of degree 4. Unfortunately, even inequalities of degree 2 do not have short OBDD representation, in contrast to inequalities of degree 1. Nevertheless, the proof of [9] may be simulated in  $\text{OBDD}(\wedge, \text{weakening})$  in some order over the variables.

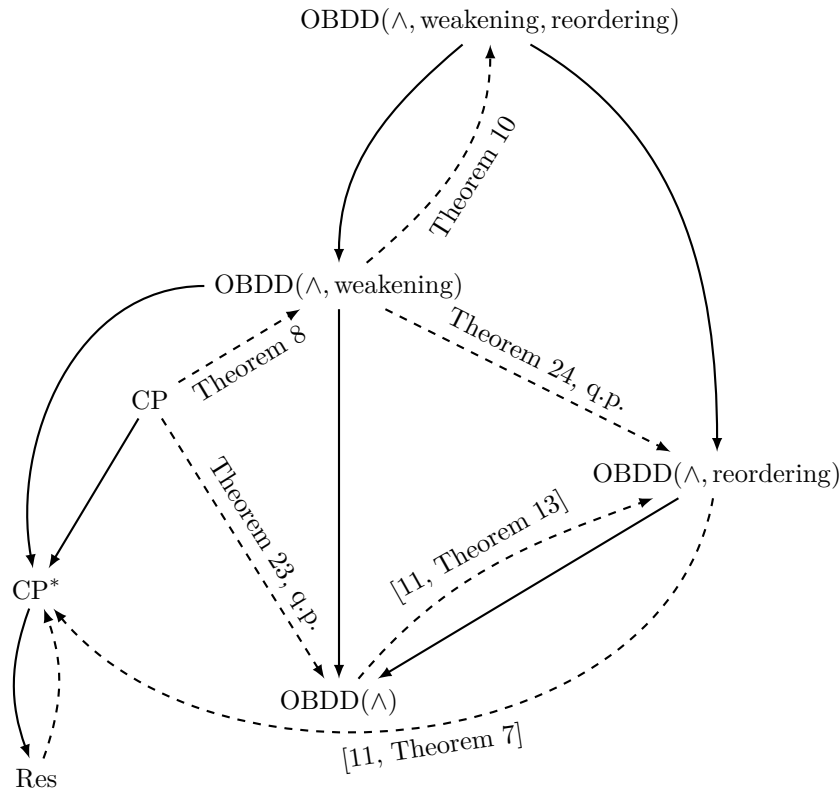
An interesting subsystem of  $\text{OBDD}(\wedge, \text{weakening})$  is the system  $\text{OBDD}(\wedge)$  that uses only the join rule; this system is connected with early OBDD algorithms for SAT-solving [22]. Tveretina et al. [21] proved that  $\text{PHP}_n^{n+1}$  is hard for  $\text{OBDD}(\wedge)$ . Grut and Zantema [10] showed that there is an unsatisfiable formula (not in CNF) such that it has an efficient construction in OBDDs and any resolution proof of its Tseitin transformation has exponential size. Because of the different translations, the question of an actual separation between  $\text{OBDD}(\wedge)$  and resolution was left open. In Corollary 12 and Lemma 13, we improve their result by giving CNF formulas which have polynomial size  $\text{OBDD}(\wedge)$  proofs but require superpolynomial (actually, quasipolynomial size) resolution proofs.

Järvisalo [12] claimed an exponential separation between tree-like resolution proofs and (dag-like)  $\text{OBDD}(\wedge)$  proofs. Unfortunately, as is discussed in Section 5, the proof for the last claim was erroneous. We correct the proof and establish an even stronger result: the proof of Theorem 32 shows that there is a formula  $\psi_n$  such that in some order  $\pi$  any tree-like  $\pi\text{-OBDD}(\wedge, \text{weakening})$  proof of  $\psi_n$  has exponential size, but there is a short  $\text{OBDD}(\wedge)$  proof of  $\psi_n$  in another order. Note that tree-like  $\pi\text{-OBDD}(\wedge, \text{weakening})$  simulates tree-like resolution for any order  $\pi$ .

So far, we have only discussed OBDD proof systems for which proofs consists of  $\pi\text{-OBDDs}$  in the same fixed order  $\pi$ . This constraint is somewhat artificial since there is an algorithm to transform an OBDD in one order into an OBDD in another order which runs in time polynomially bounded by the combined sizes of the input and output OBDDs. Accordingly, Itsykson et al. [11] introduced the proof system  $\text{OBDD}(\wedge, \text{reordering})$ . This system includes a *reordering* rule which allows changing an OBDD to a different variable ordering. It also includes the join ( $\wedge$ ) rule, but with the condition that the two conjoined OBDDs use the same variable ordering. They showed that  $\text{OBDD}(\wedge, \text{reordering})$  does not have short proofs of  $\text{PHP}_n^{n+1}$  or of Tseitin formulas based on expanders. Additionally, they showed that  $\text{OBDD}(\wedge, \text{reordering})$  is strictly stronger than  $\text{OBDD}(\wedge)$ . In Theorem 10, we resolve an open question of [11] by showing that  $\text{OBDD}(\wedge, \text{weakening}, \text{reordering})$  is strictly stronger than  $\text{OBDD}(\wedge, \text{weakening})$ .

Theorem 24 constructs formulas that have tree-like  $\text{OBDD}(\wedge, \text{reordering})$  proofs of small size but require superpolynomially larger size (dag-like)  $\text{OBDD}(\wedge, \text{weakening})$  proofs. The proof uses a result of [7] and formulas that have short  $\text{OBDD}(\wedge)$  refutations but require superpolynomial size resolution proofs. This method also allows constructing formulas





■ **Figure 1**  $C_1 \longrightarrow C_2$  denotes  $C_1$   $p$ -simulates  $C_2$ , and  $C_1 \dashrightarrow C_2$  denotes  $C_1$  does not  $p$ -simulate  $C_2$ . The results are for the dag-like versions of the systems. New results are labelled with the relevant theorem. All the separations on the picture are exponential, except the two separations labeled by “q,p” for “quasipolynomial”.

that are hard for CP but easy for OBDD(∧), see Theorem 23. In Theorem 32, we give CNF formulas which have polynomial size tree-like OBDD(∧, reordering) proofs but require exponential size for tree-like OBDD(∧, weakening) proofs.

A summary of the (non-)simulation results for dag-like systems is shown in Figure 1. There are still a few questions left open about the systems shown there. First, it is a long-standing open problem whether CP\* simulates CP. Second, it is open whether OBDD(∧, weakening) simulates CP. Third, we do not know whether resolution is simulated by OBDD(∧, reordering). In fact, we do not know whether resolution is simulated by OBDD(∧). A couple of earlier papers have claimed that resolution is not simulated by OBDD(∧), see Theorem 5 of [21] and Corollary 4 of [12], but we have been unable to verify their proofs.<sup>1</sup>

All the other missing arrows in Figure 1 follow from the arrows shown. For instance, OBDD(∧) does not simulate CP\*, since OBDD(∧, reordering) does not simulate CP\*.

<sup>1</sup> The difficult point in the proofs is in Lemma 8 of [21] and in Lemma 4 of [12]. In the former, it is shown that two distinct nodes in an OBDD  $B(F, \prec)$  correspond to two distinct nodes in another OBDD  $B(F \cup G, \prec)$ ; however, it does not follow from this that  $n$  distinct nodes in  $B(F, \prec)$  correspond to  $n$  distinct nodes in  $B(F \cup G, \prec)$ . A similar technique is implicitly used in the latter paper, and it is possible to give a counterexample to Lemma 4 of [12].



### Further research

Seegerind showed [19] that dag-like resolution does not polynomially simulate tree-like OBDD( $\wedge$ , weakening), hence dag-like OBDD( $\wedge$ , weakening) is strictly stronger than tree-like OBDD( $\wedge$ , weakening). It is open whether OBDD( $\wedge$ ), OBDD( $\wedge$ , reordering) and OBDD( $\wedge$ , weakening, reordering) are simulated by their tree-like versions.

It is interesting open question, whether resolution quasipolynomially simulates OBDD( $\wedge$ ). Any improving of our separation will automatically improve separations between CP vs. OBDD( $\wedge$ ) and OBDD( $\wedge$ , weakening) vs. OBDD( $\wedge$ , reordering).

The major open question is to prove a superpolynomial lower bound on the size of OBDD( $\wedge$ , weakening, reordering) refutations.

## 2 Preliminaries

### 2.1 Ordered Binary Decision Diagrams

An ordered binary decision diagram (OBDD) is used to represent a Boolean function [3]. Let  $\Gamma = \{x_1, \dots, x_n\}$  be a set of propositional variables. A binary decision diagram (BDD) is a directed acyclic graph with one source. Each vertex of the graph is labeled by a variable from  $\Gamma$  or by a constant 0 or 1. If a vertex is labeled by a constant, then it is a sink (has out-degree 0). If a vertex is labeled by a variable, then it has exactly two outgoing edges: one edge is labeled by 0 and the other edge is labeled by 1. Every binary decision diagram defines a Boolean function  $\{0, 1\}^n \rightarrow \{0, 1\}$ . The value of the function for given values of  $x_1, \dots, x_n$  is computed as follows: we start a path at the source and at every step follow the edge that corresponds to the value of the variable labelling the current vertex. Every such path reaches a sink, which is labelled either 0 or 1: this constant is the value of the function.

Let  $\pi$  be a permutation of the set  $[n] = \{1, \dots, n\}$ . A  $\pi$ -ordered binary decision diagram ( $\pi$ -OBDD) is a binary decision diagram such that on every path from the source to a sink every variable has at most one occurrence and the variable  $x_{\pi(i)}$  can not appear before  $x_{\pi(j)}$  if  $i > j$ . An ordered binary decision diagram (OBDD) is a  $\pi$ -ordered binary decision diagram for some permutation  $\pi$ . By convention, every OBDD is associated with a single fixed permutation  $\pi$ . This  $\pi$  puts a total order on all the variables, even if the OBDD does not query all variables.

OBDDs have a number of nice properties. Size of an OBDD is the number of vertices in it, and for a fixed ordering  $\pi$  of variables, every Boolean function has a unique minimal  $\pi$ -OBDD. Furthermore, the minimal  $\pi$ -OBDD of a function  $f$  may be constructed in polynomial time from any  $\pi$ -OBDD for the same  $f$ . There are also polynomial-time algorithms which act on  $\pi$ -OBDDs and efficiently perform the operations of conjunction, negation, disjunction, and projection [16]. (Projection is the operation that maps a  $\pi$ -OBDD  $D$  computing the Boolean function  $f(x, y_1, \dots, y_n)$  to a  $\pi$ -OBDD  $D'$  computing the Boolean function  $\exists x f(x, y_1, \dots, y_n)$ .) In addition, there is an algorithm running in time polynomial in the combined sizes of the input and the output which takes as input a  $\pi$ -OBDD  $D$  and a permutation  $\rho$ , and returns the minimal  $\rho$ -OBDD that represents the same function as  $D$  [16].

## 2.2 Proof Systems

### 2.2.1 Resolution

For an unsatisfiable CNF formula  $\varphi$ , a resolution refutation of  $\varphi$  (often called a “resolution proof”) is a sequence of clauses with the following properties: the last clause is an empty clause; and every clause is either a clause of the initial formula  $\varphi$ , or can be obtained from

previous ones by the resolution rule. The resolution rule allows inferring a clause  $(B \vee C)$  from clauses  $(x \vee B)$  and  $(\neg x \vee C)$ . The size of a resolution refutation is the number of clauses in it. It is well known that the resolution proof system is sound and complete. Soundness means that if a formula has a resolution refutation then it is unsatisfiable. Completeness means that every unsatisfiable CNF formula has a resolution refutation. If every clause is used as a premise of the inference rule at most once, then the proof is *tree-like*.

### 2.2.2 Cutting Planes

Before we give a definition of this proof system let us define the translation of clauses into linear inequalities by the following rule: if  $C = \bigvee_{i=1}^n x_i^{b_i}$ , then  $L(C)$  is the following inequality

$\sum_{i=1}^n (-1)^{1-b_i} x_i \geq 1 - \sum_{i=1}^n (1 - b_i)$  where  $x^0$  denotes  $\neg x$  and  $x^1$  denotes  $x$ . For an unsatisfiable CNF formula  $\varphi$  over the variables  $x_1, \dots, x_n$ , a Cutting Planes refutation of  $\varphi$  is a sequence of inequalities  $I_1, \dots, I_t$  of the type  $\sum_{i=1}^n a_i x_i \geq c$  (where  $a_i, c \in \mathbb{Z}$ ) such that  $I_t$  is an inequality  $0 \geq 1$  and every inequality  $I_j$  either is  $L(C)$  where  $C$  is some clause of the initial formula  $\varphi$  or can be obtained from previous inequalities by the following rules:

**Linear Combination:**  $I_j$  is an inequality  $\sum_{i=1}^n (\alpha \cdot a_i + \beta \cdot b_i) x_i \geq \alpha c + \beta d$  where for some  $\alpha, \beta > 0$  and  $1 \leq k, \ell < j$ ,  $I_k$  is an inequality  $\sum_{i=1}^n a_i x_i \geq c$  and  $I_\ell$  is an inequality  $\sum_{i=1}^n b_i x_i \geq d$ ;

**Division:**  $I_j$  is an inequality  $\sum_{i=1}^n a_i x_i \geq \lceil c/d \rceil$ , where for some  $k < j$ ,  $I_k$  is an inequality  $\sum_{i=1}^n d a_i x_i \geq c$ .

The size of such a refutation is the number of inequalities.

Additionally, we say that an unsatisfiable CNF formula  $\varphi$  has CP\* refutation of size  $S$  iff there is a CP refutation of  $\varphi$  such that the sum of absolute values of coefficients in the inequalities in this proof is at most  $S$ .<sup>2</sup>

We say that an unsatisfiable CNF formula  $\varphi$  has a semantic CP refutation (semantic CP\* refutation) of size  $S$  if there is a CP refutation of  $\varphi$  of size  $S$  such that instead of these rules we allow deriving any semantic implication of at most two previously derived inequalities. Note that semantic CP (semantic CP\*) is not a Cook–Reckhow proof system since it is NP-hard to check the correctness of the semantic rule. A proof is *tree-like* if every inequality is used as a premise of an inference at most once.

### 2.2.3 OBDD-based Proof Systems

Let  $\varphi$  be an unsatisfiable CNF formula. An OBDD proof of  $\varphi$  is a sequence  $D_1, D_2, \dots, D_t$  of OBDDs and permutations  $\pi_1, \dots, \pi_t$  such that  $D_t$  is a  $\pi_t$ -OBDD that represents the constant false function, and such that each  $D_i$  is either a  $\pi_i$ -OBDD which represents a clause of  $\varphi$  or can be obtained from previous OBDDs by one of the following inference rules:

**Join (or  $\wedge$ ):**  $D_i$  represents the Boolean function  $D_k \wedge D_\ell$  for  $1 \leq \ell, k < i$ , where  $D_i, D_k, D_\ell$  have the same order  $\pi_i = \pi_k = \pi_\ell$ ;

<sup>2</sup> Many authors define CP\* differently, by bounding the coefficients by a polynomial of the size of the formula. All the results for CP\* stated in the present paper hold under both definitions.

**Weakening:** there exists a  $1 \leq j < i$  such that  $D_i$  and  $D_j$  have the same order  $\pi_i = \pi_j$ , and  $D_j$  semantically implies  $D_i$ . The latter means that every assignment that satisfies  $D_j$  also satisfies  $D_i$ ;

**Reordering:**  $D_i$  is a  $\pi_i$ -OBDD that is equivalent to a  $\pi_j$ -OBDD  $D_j$  with  $1 \leq j < i$ .

Note that although we use terminology “OBDD proof”, it is actually a *refutation* of  $\varphi$ . By the discussion in the previous section, there is a polynomial time algorithm which recognizes whether a given  $D_1, \dots, D_t$  and  $\pi_1, \dots, \pi_t$  is a valid OBDD proof of a given  $\varphi$ . The size of this proof is equal to  $\sum_{i=1}^t |D_i|$ .

We use several different OBDD proof systems with different sets of allowed rules. For example, the  $\text{OBDD}(\wedge, \text{weakening})$  proof system uses conjunction and weakening rules; hence, all OBDDs in such a proof have the same order  $\pi$ . We use the notation  $\pi\text{-OBDD}(\wedge)$  proof and  $\pi\text{-OBDD}(\wedge, \text{weakening})$  proof to explicitly indicate the ordering. If every  $D_i$  is used as a premise of the inference rule at most once, then the proof is *tree-like*.

### 3 OBDD( $\wedge$ , weakening, reordering) is Strictly Stronger Than OBDD( $\wedge$ , weakening)

This section constructs formulas which are easy for  $\text{OBDD}(\wedge, \text{weakening, reordering})$  and hard for  $\text{OBDD}(\wedge, \text{weakening})$ . For this, we construct a transformation  $\mathcal{T} = \mathcal{T}(\varphi)$  such that

- If a formula  $\varphi$  is hard for  $\pi\text{-OBDD}(\wedge, \text{weakening})$  for some order  $\pi$ , then  $\mathcal{T}(\varphi)$  is hard for  $\text{OBDD}(\wedge, \text{weakening})$ ; i.e.,  $\mathcal{T}(\varphi)$  is hard for any order.
- If a formula  $\varphi$  is easy for  $\pi\text{-OBDD}(\wedge, \text{weakening})$  for some order  $\pi$ , then  $\mathcal{T}(\varphi)$  is easy  $\text{OBDD}(\wedge, \text{weakening, reordering})$ .

Then we construct a formula  $\varphi$  such that there are two orders  $\pi_1$  and  $\pi_2$  such that  $\varphi$  is hard for  $\pi_1\text{-OBDD}(\wedge, \text{weakening})$  but easy for  $\pi_2\text{-OBDD}(\wedge, \text{weakening})$ . As a corollary, we get that  $\mathcal{T}(\varphi)$  separates  $\text{OBDD}(\wedge, \text{weakening, reordering})$  and  $\text{OBDD}(\wedge, \text{weakening})$ .

We will apply this transformation to a formula  $\varphi$  expressing the Clique-Coloring principle ( $\text{Clique-Coloring}_{n,m}$ ) that any  $(m - 1)$ -colorable graph on  $n$  vertices does not contain a clique of size  $m$  for  $m \approx \sqrt{n}$ . Atserias, Kolaitis, and Vardi [1] proved (see also Krajíček [14]) that  $\text{Clique-Coloring}_{n,m}$  is hard for  $\pi\text{-OBDD}(\wedge, \text{weakening})$  for some order  $\pi$ . However, in Section 6 we show that there is an order  $\pi$  such that  $\text{Clique-Coloring}_{n,m}$  has a  $\pi\text{-OBDD}(\wedge, \text{weakening})$  proof of size polynomially bounded by  $n$  and  $m$ .

#### 3.1 Construction of $\mathcal{T}$

The transformation  $\mathcal{T}$  is the same as a construction of Segerlind [19]. We develop the definition of  $\mathcal{T}$  in stages. As a first approximation, we define how to transform a formula  $\varphi(x_1, \dots, x_n)$  into a formula  $\text{perm}_{S_n}(\varphi)(z_1, \dots, z_\ell, x_1, \dots, x_n)$  where  $\ell = \lceil \log(n!) \rceil$ . Fix an injective map  $\text{rep} : S_n \rightarrow \{0, 1\}^\ell$  that maps the set of permutations of  $[n]$  into binary strings of length  $\ell$ . The formula  $\text{perm}_{S_n}(\varphi)$  is defined by:

$$\begin{aligned} \text{perm}_{S_n}(\varphi)(z_1, \dots, z_\ell, x_1, \dots, x_n) = & \bigwedge_{\sigma \in S_n} \left[ \left( \bigwedge_{i=1}^{\ell} z_i = \text{rep}(\sigma)_i \right) \rightarrow \varphi(x_{\sigma(1)}, \dots, x_{\sigma(n)}) \right] \\ & \wedge \bigwedge_{t \in \{0,1\}^\ell \setminus \text{rep}(S_n)} \neg(z_1 = t_1 \wedge z_2 = t_2 \wedge \dots \wedge z_\ell = t_\ell). \end{aligned}$$

Note that it is easy to convert  $\text{perm}_{S_n}(\varphi)$  into a formula in CNF. We just add to each clause of  $\varphi(x_{\sigma(1)}, \dots, x_{\sigma(n)})$  the literals  $z_1^{1-\text{rep}(\sigma)_1}, z_1^{1-\text{rep}(\sigma)_2}, \dots, z_\ell^{1-\text{rep}(\sigma)_\ell}$ , where  $z_i^0$

denotes  $\neg z_i$ , and  $z_i^1$  denotes  $z_i$ , and also add the clauses  $\neg(z_1 = t_1 \wedge z_2 = t_2 \wedge \dots \wedge z_\ell = t_\ell)$ . It is easy to see that the formula  $\text{perm}_{S_n}(\varphi)$  is unsatisfiable since if a substitution to variables  $z_1, z_2, \dots, z_\ell$  does not correspond to a representation of some permutation, then this substitution falsifies the constraint  $\neg(z_1 = t_1 \wedge z_2 = t_2 \wedge \dots \wedge z_\ell = t_\ell)$  and if a substitution to the variables  $z_1, z_2, \dots, z_\ell$  corresponds to a permutation  $\sigma$ , then the formula  $\left(\bigwedge_{i=1}^{\ell} z_i = \text{rep}(\sigma)_i\right) \rightarrow \varphi(x_{\sigma(1)}, \dots, x_{\sigma(n)})$  is falsified by this substitution, since  $\varphi$  is unsatisfiable.

Applying the partial substitution  $z_i := \text{rep}(\sigma)_i$  for all  $i$  to  $\text{perm}_{S_n}(\varphi)(z_1, \dots, z_\ell, x_1, \dots, x_n)$  yields the formula  $\varphi(x_{\sigma(1)}, \dots, x_{\sigma(n)})$ . This implies that if  $\varphi$  requires a  $\pi$ -OBDD( $\wedge$ , weakening) proof of size  $S$  for some order  $\pi$ , then  $\text{perm}_{S_n}(\varphi)$  requires an OBDD( $\wedge$ , weakening) proof of size  $S$  in any order. Indeed, let  $\tau$  be an order on the variables  $z_1, z_2, \dots, z_\ell, x_1, x_2, \dots, x_n$  and let  $\sigma$  be the order on the variables  $x_1, \dots, x_n$  induced by  $\tau$ . The substitution  $z_1 z_2 \dots z_\ell := \text{rep}(\pi\sigma^{-1})$  transforms a  $\tau$ -OBDD( $\wedge$ , weakening) proof of  $\text{perm}_{S_n}(\varphi)$  to a  $\pi$ -OBDD( $\wedge$ , weakening) proof of  $\varphi$  with no increase in size. Hence the size of the minimal OBDD( $\wedge$ , weakening) proof of  $\text{perm}_{S_n}(\varphi)$  is at least  $S$ .

The problem with the transformation  $\text{perm}_{S_n}$  is that  $\text{perm}_{S_n}(\varphi)$  can be exponentially big. So the next idea for a transformation is to consider a small “good” set of permutations  $\Pi \subseteq S_n$  instead of all of  $S_n$ . Letting  $\ell = \lceil \log |\Pi| \rceil$  and letting  $\text{rep}$  now be some injective map  $\text{rep} : \Pi \rightarrow \{0, 1\}^\ell$ , we define analogously

$$\begin{aligned} \text{perm}_\Pi(\varphi)(z_1, \dots, z_\ell, x_1, \dots, x_n) = & \bigwedge_{\sigma \in \Pi} \left[ \left( \bigwedge_{i=1}^{\ell} z_i = \text{rep}(\sigma)_i \right) \rightarrow \varphi(x_{\sigma(1)}, \dots, x_{\sigma(n)}) \right] \\ & \wedge \bigwedge_{t \in \{0,1\}^\ell \setminus \text{rep}(\Pi)} \neg(z_1 = t_1 \wedge z_2 = t_2 \wedge \dots \wedge z_\ell = t_\ell). \end{aligned}$$

The problem with this is that it is possible that  $\pi\sigma^{-1}$  does not belong to  $\Pi$ .

To solve this problem we orify variables: each variable  $x_i$  is replaced by the disjunction of  $m$  fresh variables  $y_{i,1}, \dots, y_{i,m}$ ; i.e., instead of  $\varphi(x_1, x_2, \dots, x_n)$  we consider  $\varphi^{\vee m}(y_{1,1}, \dots, y_{n,m}) = \varphi\left(\bigvee_{j=1}^m y_{1,j}, \dots, \bigvee_{j=1}^m y_{n,j}\right)$ . Now let  $\Pi \subseteq S_{mn}$  and consider  $\text{perm}_\Pi(\varphi^{\vee m})$ . As in previous case we want to substitute variables to a proof of  $\text{perm}_\Pi(\varphi^{\vee m})$  in some order and get a proof of  $\varphi$  in order  $\pi$ . However, in this case we substitute not only for the variables  $z_1, \dots, z_\ell$ , but also for each  $k \in [n]$  we substitute zero for all variables  $y_{k,i}$  except one. This increases the number of different permutations of the variables  $x_1, \dots, x_n$  that we can obtain. The only problem with this transformation is that for some formulas  $\varphi$ , size of  $\varphi^{\vee m}$  may be exponentially bigger than size of  $\varphi$ . However, if each clause of  $\varphi$  there is only  $O(1)$  negated literals, then size of  $\varphi^{\vee m}$  will be polynomially bounded.

Our “good” set of permutations is a set of pairwise independent permutations. Let  $t = \lceil \log(n) \rceil$  and  $N = 2^t$ , and  $\mathbb{F}$  be the field  $\text{GF}(N)$ . Define  $\Pi_n$  to be the set of all mappings given by  $x \mapsto ax + b$  with  $a, b \in \mathbb{F}$  and  $a \neq 0$ . Elements of  $\Pi_n$  may be represented by binary strings of length  $\ell = 2t$  such that the first  $t$  bits are not all zero. Note that  $\Pi_n \subseteq S_N$  so we have to add new variables,  $x_{n+1}, \dots, x_N$  and assume that  $\varphi$  does not depend on them. Then define

$$\text{perm}(\varphi)(z_1, \dots, z_\ell, x_1, \dots, x_N) = \bigwedge_{\sigma \in \Pi_n} \left[ \left( \bigwedge_{i=1}^{\ell} z_i = \text{rep}(\sigma)_i \right) \rightarrow \varphi(x_{\sigma(1)}, \dots, x_{\sigma(N)}) \right] \wedge \bigvee_{i=1}^t z_i.$$

Now we can define the transformation  $\mathcal{T}$ . Let  $\varphi$  be a formula on  $n$  variables and  $m$  be the least integer such that  $\frac{2n^3}{m} + \frac{n^2}{mn-1} < 1$ , so  $m = O(n^3)$ . Then  $\mathcal{T}(\varphi) = \text{perm}(\varphi^{\vee m})$ . The first property of  $\mathcal{T}$  given at the beginning of Section 2.2 was established by Segerlind [19]:

► **Lemma 1** ([19]). *Let  $\varphi$  be an unsatisfiable formula in CNF on the variables  $x_1, \dots, x_n$ . Suppose there is an OBDD( $\wedge$ , weakening) proof (respectively, an OBDD( $\wedge$ ) proof) of the formula  $\mathcal{T}(\varphi)$  of size  $S$ . Then for every order  $\pi$  on  $x_1, \dots, x_n$  there is a  $\pi$ -OBDD( $\wedge$ , weakening) proof (respectively, a  $\pi$ -OBDD( $\wedge$ ) proof) of  $\varphi$  of size at most  $S$ .*

The idea of the proof of lemma is as follows. Suppose  $\tau \in \Pi_n$  is an order on  $z_1, \dots, z_\ell, x_1, \dots, x_n$ , and let  $\pi$  be an order on  $x_1, \dots, x_n$ . Then there are  $j_1, \dots, j_n$  such the order  $\tau$  restricted to  $y_{1,j_1}, \dots, y_{n,j_n}$  is the same as the order  $\pi$  on  $x_1, \dots, x_n$ . Replacing the variables  $z_i$  with the constants  $\text{rep}(\tau)_i$ , renaming the variables  $y_{i,j_i}$  to  $x_i$ , and replacing all other variables  $y_{i,j}$  with 0 thus transforms the OBDD( $\wedge$ , weakening) or OBDD( $\wedge$ ) proof of  $\mathcal{T}(\varphi)$  into a proof of  $\varphi$ . For details, consult Segerlind [19].

The second property of  $\mathcal{T}$  states that if  $\varphi$  is easy for OBDD( $\wedge$ , weakening) in some order, then  $\mathcal{T}(\varphi)$  is easy for OBDD( $\wedge$ , weakening, reordering). Its proof consists of two parts: First, Lemma 2 shows that if  $\varphi$  is easy for OBDD( $\wedge$ , weakening), then  $\text{perm}(\varphi)$  is easy for OBDD( $\wedge$ , weakening, reordering); then Section 3.2 shows that if  $\varphi$  is easy for OBDD( $\wedge$ , weakening), then  $\varphi^{\vee m}$  is easy for OBDD( $\wedge$ , weakening).

► **Lemma 2.** *Let  $\varphi_n(x_1, x_2, \dots, x_n)$  be a family of unsatisfiable formulas such that for each  $n$ , there is an order  $\tau$  so that  $\varphi_n$  has a  $\tau$ -OBDD( $\wedge$ , weakening) proof  $P_1$  of size  $t(n)$ . Then the formula  $\text{perm}(\varphi_n)$  has an OBDD( $\wedge$ , weakening, reordering) proof  $P_2$  of size  $t(n)\text{poly}(n)$ . If  $P_1$  is tree-like, then so is  $P_2$ . In addition, if  $P_1$  does not use the weakening rule, then neither does  $P_2$ .*

**Proof.** Suppose  $P_1$  is a  $\tau$ -OBDD( $\wedge$ , weakening) proof of  $\varphi_n(x_1, x_2, \dots, x_n)$  of size  $t(n)$  using the order  $\tau$  on  $x_1, x_2, \dots, x_n$ . We describe an OBDD( $\wedge$ , weakening, reordering) proof  $P_2$  of  $\text{perm}(\varphi_n)$ . For  $\sigma$  a permutation in  $\Pi_n$ , let  $\mu_\sigma$  be the order on  $z_1, z_2, \dots, z_\ell, x_1, x_2, \dots, x_n$  such that  $x_1, x_2, \dots, x_n$  are ordered by  $\tau\sigma^{-1}$  and follow the variables  $z_1, z_2, \dots, z_\ell$ . In other words,  $\mu_\sigma$  orders variables as follows:  $z_1, z_2, \dots, z_\ell, x_{\tau\sigma^{-1}(1)}, x_{\tau\sigma^{-1}(2)}, \dots, x_{\tau\sigma^{-1}(n)}$ .

For  $\sigma \in \Pi_n$ , it is easy to transform the proof  $P_1$  into a  $\mu_\sigma$ -OBDD( $\wedge$ ) derivation  $P_{1,\sigma}$  of a diagram that represents  $\neg \left( \bigwedge_{i=1}^\ell z_i = \text{rep}(\sigma)_i \right)$  from the CNF formula  $\left( \bigwedge_{i=1}^\ell z_i = \text{rep}(\sigma)_i \right) \rightarrow \varphi_n(x_{\sigma(1)}, \dots, x_{\sigma(n)})$ . Namely each diagram  $D$  of  $P_1$  is replaced by the diagram  $D_\sigma \vee \neg \left( \bigwedge_{i=1}^\ell z_i = \text{rep}(\sigma)_i \right)$ , where  $D_\sigma$  is  $D$  with the variables  $x_i$  permuted according to  $\sigma$ . Since the variables  $z_1, z_2, \dots, z_\ell$  precede the variables  $x_1, \dots, x_n$  in the order  $\mu_\sigma$ , each diagram  $D_\sigma \vee \neg \left( \bigwedge_{i=1}^\ell z_i = \text{rep}(\sigma)_i \right)$  has size  $|D| + O(\ell)$ , where  $|D|$  is the size of  $D$ . Hence,  $|P_{1,\sigma}|$  is  $t(n) \cdot (1 + O(\ell))$ .

For  $\sigma \in \Pi_n$ , the hypotheses of  $P_{1,\sigma}$  are clauses of  $\text{perm}(\varphi_n)$ . Therefore combining the derivations  $P_{1,\sigma}$  gives immediately a derivation of the diagrams which represent  $\neg \left( \bigwedge_{i=1}^\ell z_i = \text{rep}(\sigma)_i \right)$  for  $\sigma \in \Pi_n$  and a diagram encoding  $\bigvee_{i=1}^\ell z_i$ . Formally, these diagrams use different orders  $\mu_\sigma$  but these differ only in how they order the variables  $x_1, \dots, x_n$  that do not occur in the derived diagrams. Thus, the reordering rule can be used to change the orders in all of these diagrams to some “standard” one, without changing the diagrams. Repeatedly applying the conjunction rule to these diagrams yields the constant false diagram since  $z_1 z_2 \dots z_\ell$  is equal to  $\text{rep}(\sigma)$  for some  $\sigma \in \Pi_n$  or  $z_1 = z_2 = \dots = z_t = 0$ . All intermediate diagrams use only  $\ell$  variables and thus have size at most  $O(2^\ell)$ . The overall size of the proof  $P_2$  is  $|\Pi_n| \cdot t(n)(1 + O(\ell)) + O(2^\ell |\Pi_n|) = t(n)\text{poly}(n)$  since  $\ell = 2t = 2\lceil \log n \rceil$ .

The construction preserves the tree-like property, and whether the weakening rule is used, so Lemma 2 is proved. ◀

### 3.2 Complexity of Composition

We now prove that if  $\varphi$  has a small OBDD( $\wedge$ , weakening) proof, then  $\varphi^{\vee m}$  has a small OBDD( $\wedge$ , weakening) proof. In fact, we prove more a general statement. Let  $\varphi$  be a CNF formula with  $n$  variables, and  $g : \{0, 1\}^k \rightarrow \{0, 1\}$  be a Boolean function. Then  $\varphi \circ g$  denotes a CNF formula on  $kn$  variables that represents  $\varphi(g(\vec{x}_1), g(\vec{x}_2), \dots, g(\vec{x}_n))$ , where  $\vec{x}_i$  denotes a vector of  $k$  new variables.  $\varphi \circ g$  is constructed by applying the substitution to every clause  $C$  of  $\varphi$  and converting the resulting function  $C \circ g$  to CNF in some fixed way.

We need the following technical definition. Consider a CNF formula  $\varphi = \bigwedge_{i=1}^m C_i$ . We say  $\varphi$  is  $S$ -constructible with respect to (w.r.t.) the order  $\pi$  if there is a binary tree with vertices labeled by  $\pi$ -OBDDs such that: (1) the root is labeled by a  $\pi$ -OBDD representation of  $\varphi$ , (2) the tree contains  $m$  leaves labeled by  $\pi$ -OBDD representations of the clauses  $C_i$ , each clause appears in exactly one leaf, (3) each vertex is labelled by a  $\pi$ -OBDD that represents the conjunction of labels of its children, and (4) the size of each label is at most  $S$ .

► **Remark.** If  $\varphi$  is  $S$ -constructible CNF w.r.t. the order  $\pi$ , then there is a tree-like  $\pi$ -OBDD( $\wedge$ ) derivation of size  $(2m - 1)S$  of a  $\pi$ -OBDD that represents  $\varphi$  from the clauses of  $\varphi$ .

► **Proposition 3.** *Let  $F = G_1 \vee G_2$ , where  $G_1$  and  $G_2$  are Boolean functions that depend on disjoint sets of variables. If the variables of  $G_1$  precede variables of  $G_2$  in the order  $\pi$ , then the smallest size of a  $\pi$ -OBDD representation of  $F$  is at most the sum of sizes of the smallest  $\pi$ -OBDD representations of  $G_1$  and  $G_2$ .*

**Proof.** This is obvious. The  $\pi$ -OBDD for  $F$  can be obtained by the identifying the source of the  $\pi$ -OBDD for  $G_2$  with the sink of the  $\pi$ -OBDD for  $G_1$  labeled by 0. ◀

► **Lemma 4.** *Let  $F_1, F_2, \dots, F_k$  be CNF formulas with disjoint sets of variables, where  $F_j = \bigwedge_{i \in I_j} C_i$  for all  $j \in [k]$ . Let  $\pi_1, \dots, \pi_k$  be orders such that each  $F_j$  is  $S$ -constructible w.r.t.  $\pi_j$ . Define the order  $\pi$  to order the variables of each  $F_i$  according to  $\pi_i$  and so that all the variables of  $F_i$  precede all the variables of  $F_{i+1}$ . Let  $F$  be the CNF representation of the function  $F_1 \vee F_2 \vee \dots \vee F_k$ , namely,  $F = \bigwedge_{i_1 \in I_1, \dots, i_k \in I_k} \bigvee_{j=1}^k C_{i_j}$ . Then  $F$  is  $kS$ -constructible w.r.t.  $\pi$ .*

**Proof.** We prove this lemma by induction on  $k$ . The basis case is trivial: if  $k = 1$ , then  $F = F_1$ , hence  $F$  is  $S$ -constructible. For the induction hypothesis, let  $G = F_1 \vee F_2 \vee \dots \vee F_{k-1}$ . By the induction hypothesis  $G$  is  $(k-1)S$ -constructible w.r.t.  $\pi$ . For each clause  $D$  of  $G$  and each  $i \in I_k$ , the clause  $D \vee C_i$  is a clause of  $F$ . The formula  $F_k$  is  $S$ -constructible w.r.t.  $\pi$  by a tree  $T_k$  with  $|I_k|$  leaves which are labeled by  $C_i$  for  $i \in I_k$ . We wish to replace each leaf of  $T_k$  labelled with a  $C_i$  with a tree for  $G \vee C_i$ . Since  $G$  is  $(k-1)S$ -constructible and since the variables of  $C_i$  are disjoint from those of  $G$ , Proposition 3 implies that  $G \vee C_i$  is  $kS$ -constructible w.r.t.  $\pi$ , since we can incorporate the clause  $C_i$  into all clauses of the tree giving the  $(k-1)S$ -constructibility of  $G$ . In addition, replace all the diagrams  $D$  labelling vertices in the tree  $T_k$  by  $D \vee G$ ; by Proposition 3 the size of the updated diagrams is at most  $kS$ . This gives a tree witnessing the  $kS$ -constructibility of  $F_1 \vee \dots \vee F_k$  as desired. ◀

► **Theorem 5.** *Let  $\pi$  be an order on  $z_1, \dots, z_m$ . Let  $f$  and  $g$  be Boolean functions of  $z_1, \dots, z_m$  such that  $f = \neg g$  and that both  $f$  and  $g$  have  $S$ -constructible CNF representations w.r.t.  $\pi$ . If  $\varphi(x_1, \dots, x_n)$  is a CNF formula that has an OBDD( $\wedge$ , weakening) proof of size  $L$ , then  $\varphi \circ g$  has an OBDD( $\wedge$ , weakening) proof of size  $\text{poly}(|\varphi \circ g|, S, L)$ .*

*The statement is also true for OBDD( $\wedge$ ), tree-like OBDD( $\wedge$ ), and tree-like OBDD( $\wedge$ , weakening).*

The basic idea of Theorem 5 is that each line of a proof of  $\varphi$  can be composed with  $g$  to form a proof of  $\varphi \circ g$ ; Lemma 4 is used to handle initial clauses.

**Proof.** Let  $\varphi$  have an OBDD( $\wedge$ , weakening) proof of size  $L$  using the order  $\sigma$  on  $x_1, \dots, x_n$ . Define the order  $\tau$  on the variables  $z_{i,j}$  as follows. The variables are grouped into blocks, the  $i$ -th block is  $z_{i,1}, \dots, z_{i,m}$ . The blocks are ordered according to  $\sigma$  so all variables of block  $i$  precede those of block  $j$  iff  $x_i$  precedes  $x_j$  according to  $\sigma$ . Within the  $i$ -th block, the variables  $z_{i,1}, \dots, z_{i,m}$  are ordered according to the order  $\pi$ . We construct the desired OBDD( $\wedge$ , weakening) proof using the order  $\tau$ .

Lemma 4 implies that, for any clause  $C$ , the CNF  $C \circ g$  is  $S|C|$ -constructible in order  $\tau$ . Note that we need that both  $g$  and  $\neg g$  are  $S$ -constructible to apply Lemma 4, since variables can appear both positively and negatively in  $C$ .

Consider the following  $\tau$ -OBDD( $\wedge$ , weakening) proof of  $\varphi \circ g$ : First we create  $\tau$ -OBDDs that represent the functions  $C \circ g$  for each clause  $C$  of the formula  $\varphi$ . Then we repeat the OBDD( $\wedge$ , weakening) proof for  $\varphi$ , but we do it for  $\varphi \circ g$ . Each a diagram  $D$  from the proof of  $\varphi$  is replaced by a diagram for  $D \circ g$ . It is not hard to see that the definition of  $\tau$  allows us to replace a splitting over a variable  $x_i$  in the diagram  $D$  by a subdiagram splitting over the value of the function  $g(\vec{z}_i)$ , where  $\vec{z}_i$  is the vector of the variables  $z_{i,1}, \dots, z_{i,m}$ . This increases the proof size by at most a factor of  $S$ . The resulting proof is a correct OBDD( $\wedge$ , weakening) proof and its size is at most  $L \cdot S + |\varphi \circ g| \cdot S$ .  $\blacktriangleleft$

The clause  $\bigvee_{i=1}^m y_i$  and the CNF  $\bigwedge_{i=1}^m \neg y_i$  are both  $m$ -constructible, thus we obtain:

► **Corollary 6.** *If there is a short OBDD( $\wedge$ , weakening) proof (tree-like OBDD( $\wedge$ ) proof) of a formula  $\varphi$ , then there is a short OBDD( $\wedge$ , weakening) proof (tree-like OBDD( $\wedge$ ) proof) of the formula  $\varphi^{\vee m}$ .*

### 3.3 Separation

We have shown that if a formula  $\varphi$  is hard for OBDD( $\wedge$ , weakening) in one order, but is easy for OBDD( $\wedge$ , weakening) in another, then  $\mathcal{T}(\varphi)$  is hard for OBDD( $\wedge$ , weakening) but it is easy for OBDD( $\wedge$ , weakening, reordering). We will prove this holds for  $\varphi$  the Clique-Coloring principle.

► **Definition 7.** The Clique-Coloring principle is a formula encoding the statement that it is impossible that a graph both is  $(m-1)$ -colorable and has a  $m$ -clique. The Clique-Coloring principle uses the variables  $\{p_{i,j}\}_{i \neq j \in [n]}$ ,  $\{r_{i,l}\}_{i \in [n], l \in [m-1]}$ , and  $\{q_{k,i}\}_{k \in [m], i \in [n]}$ . Informally  $p_{i,j} = 1$  if there is an edge between vertices  $i$  and  $j$ ,  $r_{i,l} = 1$  if vertex  $i$  has color  $l$ , and  $q_{k,i} = 1$  if vertex  $i$  is the  $k$ -th vertex in the clique.

More formally, the Clique-Coloring principle is the conjunction of the following statements written as clauses. For technical reasons we also express the clauses as inequalities with integer coefficients:

1.  $\bigvee_{i=1}^n q_{k,i}$  ( $\sum_{i=1}^n q_{k,i} \geq 1$ ) for any  $k \in [m]$ . This states that the clique has a vertex with number  $k$ .
2.  $\neg q_{k,i} \vee \neg q_{k',j} \vee p_{i,j}$  ( $q_{k,i} + q_{k',j} \leq p_{i,j} + 1$ ) for all  $i \neq j \in [n]$  and  $k \neq k' \in [m]$ . This states that there is an edge between the  $i$ -th and  $j$ -th vertices of the clique.
3.  $\neg q_{k,i} \vee \neg q_{k,j}$  ( $q_{k,i} + q_{k,j} \leq 1$ ) for any  $k \in [m]$  and  $i \neq j \in [n]$ . This states that at most one element in the clique with number  $k$ .



4.  $\neg q_{k,i} \vee \neg q_{k',i}$  ( $q_{k,i} + q_{k',i} \leq 1$ ) for all  $i \in [n]$  and  $k \neq k' \in [m]$ . This states that the  $n$  vertices in clique are distinct.
  5.  $\bigvee_{l=1}^{m-1} r_{i,l}$  ( $\sum_{l=1}^{m-1} r_{i,l} \geq 1$ ) for all  $i \in [n]$ . This states that the  $i$ -th vertex has a color.
  6.  $\neg p_{i,j} \vee \neg r_{i,l} \vee \neg r_{j,l}$  ( $p_{i,j} + r_{i,l} + r_{j,l} \leq 2$ ) for all  $i \neq j$  and  $l$ . This states that if vertices  $i$  and  $j$  have the same color  $l$ , there is no edge between them.
- Clique-Coloring** $_{n,m}$  denotes the Clique-Coloring principle for  $n$  and  $m$ . This formula has size polynomially bounded by  $m$  and  $n$ .

Note that, usually Clique-Coloring principle is defined without constraints 3. We prove the next theorem in Section 6.

► **Theorem 8.** *There is an OBDD( $\wedge$ , weakening) proof of the **Clique-Coloring** $_{n,m}$  principle of size polynomial in  $n$  and  $m$ .*

An exponential lower bound on the size of proofs of the formula **Clique-Coloring** $_{n,m}$  has been given by Atserias–Kolaitis–Vardi and by Krajíček. Their proofs hold even with the addition of the constraints 3.

► **Theorem 9** ([1, 14]). *There is an order  $\pi$  such that any OBDD( $\wedge$ , weakening) proof of **Clique-Coloring** $_{n,\sqrt{n}}$  has size at least  $2^{n^{1/5}}$ .*

These two theorems let us separate the OBDD( $\wedge$ , weakening, reordering) and OBDD( $\wedge$ , weakening) proof systems.

► **Theorem 10.** *There are a family of CNF formulas  $\varphi_n$  and a constant  $c > 0$  such that:*

- $\varphi_n$  has size  $\text{poly}(n)$ ;
- there is an OBDD( $\wedge$ , weakening, reordering) proof of  $\varphi_n$  of size  $\text{poly}(n)$ ;
- any OBDD( $\wedge$ , weakening) proof of  $\varphi_n$  has size  $\Omega(2^{n^c})$ .

**Proof.** Let us consider  $\psi_n = \text{Clique-Coloring}_{n,\sqrt{n}}$ . By Theorem 9 there is an order  $\pi$  such that any  $\pi$ -OBDD( $\wedge$ , weakening) proof of the formula  $\psi_n$  has size at least  $2^{n^c}$ . Since all clauses of **Clique-Coloring** $_{n,\sqrt{n}}$  that contain a negation have constant width, the CNF encoding of **Clique-Coloring** $_{n,\sqrt{n}}^{\vee m}$  has size  $\text{poly}(n, m)$ . By Lemma 1, any OBDD( $\wedge$ , weakening) proof of the formula  $\mathcal{T}(\psi_n)$  has size  $2^{n^c}$ . In the definition of  $\mathcal{T}(\psi_n)$ , we choose  $m$  that is polynomially bounded in the number of variables in **Clique-Coloring** $_{n,\sqrt{n}}$ . Hence, by Theorem 8 and Theorem 5, there is an OBDD( $\wedge$ , weakening) proof of  $\psi_n^{\vee m}$  of size polynomial in  $n$ . As a result, by Lemma 2, there is an OBDD( $\wedge$ , weakening, reordering) proof of  $\mathcal{T}(\psi_n) = \text{perm}(\psi_n^{\vee m})$  of size  $\text{poly}(n, m)$ . Thus, we can use the formula  $\mathcal{T}(\psi_n)$  as  $\varphi_n$ . ◀

## 4 Quasipolynomial Separations for Dag-like Case

### 4.1 Resolution Does Not Polynomially Simulate OBDD( $\wedge$ )

In this section we prove that resolution does not polynomially simulate OBDD( $\wedge$ ). After that we will apply to this result a lifting technique recently developed by Garg et al. [7] and get as a corollary that Cutting Planes does not polynomially simulate OBDD( $\wedge$ ), and that OBDD( $\wedge$ , weakening) does not polynomially simulate OBDD( $\wedge$ , reordering).

A Tseitin formula  $\text{TS}_{G,c}$  is based on an undirected graph  $G(V, E)$  and a labelling function  $c : V \rightarrow \{0, 1\}$ . In this formula for every edge  $e \in E$  there is the corresponding propositional variable  $p_e$ . For every vertex  $v \in V$  we write down a formula in CNF encoding



$\sum_{u \in V: (u,v) \in E, u \neq v} p(u,v) \equiv c(v) \pmod{2}$ . The conjunction of the formulas described above is called a Tseitin formula. If  $\sum_{v \in U} c(v) \equiv 1 \pmod{2}$  for some connected component  $U \subseteq V$ , then the Tseitin formula is unsatisfiable. Indeed, if we sum up (modulo 2) all equalities corresponding to the vertices from  $U$  we get  $0 \equiv 1 \pmod{2}$  since each variable has exactly 2 occurrences. If  $\sum_{v \in U} c(v) \equiv 0 \pmod{2}$  for every connected component  $U$ , then the Tseitin formula is satisfiable ([23, Lemma 4.1]).

Tseitin formulas based on constant degree expanders are known to be hard for resolution [23]. Itsykson et al. [11] showed that they are also hard for OBDD( $\wedge$ , reordering) by giving a  $2^{\Omega(|V|)}$  lower bound. There are, of course, resolution refutations of size  $O(2^{|E|})$  since there are  $|E|$  many variables. Accordingly, we consider Tseitin formulas based on the complete graph  $K_{\log n}$  on  $\lceil \log n \rceil$  vertices, so as to have  $|V| = o(|E|)$ .

By the definition of a Tseitin formula,  $\text{TS}_{K_{\log n}, c}$  is a system of  $\lceil \log n \rceil$  linear equations and every equation depends on  $\lceil \log n \rceil - 1$  variables. Hence,  $\text{TS}_{K_{\log n}, c}$  is a  $(\lceil \log n \rceil - 1)$ -CNF formula with  $O(\log^2 n)$  variables and  $O(n \log n)$  clauses.

► **Lemma 11.** *Let  $F$  be a canonical CNF representation of an unsatisfiable linear system  $A$  over  $\mathbb{F}_2$  that contains  $m$  equations and  $n$  variables. Then for every order of variables,  $F$  has a tree-like OBDD( $\wedge$ ) proof of size at most  $8m|F|^2 + mn2^m + 2m$ .*

**Proof.** First of all, for every linear equation of  $A$  we deduce an OBDD representing this equation. Assume that a linear equation contains  $r$  variables, then its canonical CNF representation contains  $2^{r-1}$  clauses, hence  $|F| \geq 2^{r-1}$ . We deduce an OBDD representation of the equation by joining all the clauses that represent this equation. The conjunction of several clauses that represent the equation is a Boolean function from  $r$  variables, hence it has an OBDD representation of size at most  $2^{r+1} + 1$  (this is the size of an OBDD that corresponds to the complete decision tree). Hence, the size of the derivation is at most  $8|F|^2$ . And the size of the derivation of all OBDDs for all equations is at most  $8m|F|^2$ .

Finally, we join all OBDDs representing linear equations one by one and we get the constant false OBDD. The size of the described derivation may be estimated using the following claim.

► **Claim.** *For any order over the variables there is an OBDD of size at most  $n2^m + 2$  that represents the system of  $m$  linear equations over  $\mathbb{F}_2$  with  $n$  variables.*

Let us fix some order on the variables. The described OBDD will have  $n$  levels. Nodes on the  $i$ -th level are labeled with  $i$ -th variable in the chosen order.

Assume that we already tested the values of the first  $i - 1$  variables. For every equation we compute the sum modulo 2 of the values of these  $i - 1$  variables that occur in the equation. So we will have a vector of  $m$  parities. The  $i$ -th level of the OBDD contains  $2^m$  nodes corresponding to all the possible values of the vector of parities that we get after the reading of the first  $i - 1$  edges. Each node on the  $i$ -th level has two outgoing edges to nodes on the  $(i + 1)$ -th level corresponding to the way how values of variables change the partial sum. The node on the first level corresponding to all zero values of parities is the source of the OBDD (all nodes that are not reachable from the source should be removed). Outgoing edges for every node on the last level lead to a sink labelled 1 or 0 depending whether or not all the equations are satisfied. This proves the claim, and hence Lemma 11. ◀

► **Corollary 12.** *If  $\text{TS}_{K_{\log n}, c}$  is unsatisfiable Tseitin formula, then there is a tree-like OBDD( $\wedge$ ) proof of  $\text{TS}_{K_{\log n}, c}$  of size at most  $\text{poly}(n)$ .*

► **Lemma 13.** *Every resolution proof of  $\text{TS}_{K_{\log n},c}$  has size at least  $2^{\Omega(\log^2 n)}$ .*

The proof of Lemma 13 is based on the width based lower bound by Ben-Sasson and Wigderson [2]. The *width* of a clause is the number of literals in it. For a CNF formula  $\varphi$ , the *width*  $w(\varphi)$  of  $\varphi$  is the maximum width of its clauses. The *width of a resolution refutation* is a width of the largest used clause.  $w(\vdash \varphi)$  denotes the minimum width of any resolution proof of  $\varphi$ .

► **Theorem 14** ([2]). *The size of the shortest resolution refutation of any CNF formula  $\varphi$  with  $n$  variables is at least  $2^{\Omega((w(\vdash \varphi) - w(\varphi))^2/n)}$ .*

► **Theorem 15** ([2]). *The minimal width of a resolution proof of a Tseitin formula based on a graph  $G(V, E)$  is at least  $e(G)$ , where  $e(G)$  is the minimal number of edges between  $U$  and  $V \setminus U$  over all set of vertices  $U$  of size between  $|V|/3$  and  $2|V|/3$ .*

► **Corollary 16.** *If  $\text{TS}_{K_{\log n},c}$  is an unsatisfiable Tseitin formula, then  $w(\vdash \text{TS}_{K_{\log n},c}) = \Omega(\log^2 n)$ .*

**Proof.** It is straightforward that  $e(K_{\log n}) = \Omega(\log^2 n)$ . So by Theorem 13,  $w(\vdash \text{TS}_{K_{\log n},c}) = \Omega(\log^2 n)$ . ◀

**Proof of Lemma 13.** It is easy to see that  $w(\text{TS}_{K_{\log n},c}) = O(\log n)$  and  $\text{TS}_{K_{\log n},f}$  contains  $O(\log^2 n)$  variables. Thus, by Theorem 14 and by Corollary 16, size of the shortest resolution proof of  $\text{TS}_{K_{\log n},f}$  is at least  $2^{\Omega(\log^2 n)}$ . ◀

Corollary 12 and Lemma 13 give a superpolynomial separation between resolution and tree-like OBDD( $\wedge$ ). The next sections describe how to lift this to separate cutting planes and tree-like OBDD( $\wedge$ ).

## 4.2 Lifting from Resolution Width

This subsection briefly describes the results by Garg et al. [7] that allows mapping formulas with large resolution width to formulas that are hard for several stronger proof systems.

Let  $\mathcal{G}$  be a family of functions  $\{0, 1\}^n \rightarrow \{0, 1\}$  and  $\varphi$  be an unsatisfiable formula over  $n$  variables. The  $\mathcal{G}$ -refutation of  $\varphi$  is a directed acyclic graph of fan-out at most 2 with each node  $v$  labeled by a function  $g_v \in \mathcal{G}$  such that the following constraints are satisfied.

**Source:** There is a distinguished source node  $r$  with fan-in 0, and  $g_r$  is constant 0 function.

**Non-sinks:** For each non-sink node  $v$  with children  $u_1$  and  $u_2$ , we have  $g_v^{-1}(0) \subseteq g_{u_1}^{-1}(0) \cup g_{u_2}^{-1}(0)$ . And if  $v$  has only one child  $u$ , then  $g_v^{-1}(0) \subseteq g_u^{-1}(0)$ .

**Sinks:** Each sink node  $v$  is labeled by a clause  $C$  of  $\varphi$  such that  $g_v^{-1}(0) \subseteq C^{-1}(0)$  (i.e. every assignment that satisfies  $C$  also satisfies  $g_v$ ).

The size of a  $\mathcal{G}$ -refutation is the size of the graph.

The notion of  $\mathcal{G}$ -refutation extends several proof systems including resolution (if functions from  $\mathcal{G}$  are represented by clauses), Cutting Planes (if functions from  $\mathcal{G}$  are represented by linear inequalities) and OBDD( $\wedge$ , weakening) (if functions from  $\mathcal{G}$  are represented by OBDDs).  $\mathcal{G}$ -refutations are commonly called “semantic refutations”.

Let  $\Pi = (X, Y)$  be a partition of  $[n]$  into two disjoint parts. We say that  $\mathcal{G}$  is  $\Pi$ -rectangular if for every function  $g \in \mathcal{G}$ , the set  $g^{-1}(0)$  is a rectangle, i.e.  $g^{-1}(0) = A \times B$ , where  $A \subseteq \{0, 1\}^X$  and  $B \subseteq \{0, 1\}^Y$ . We say that  $\mathcal{G}$  has  $\Pi$ -communication complexity at most  $c$  iff for every  $g \in \mathcal{G}$  the communication complexity of  $g$  with respect to the partition  $\Pi$  is at most  $c$ . Notice that if  $\mathcal{G}$  is  $\Pi$ -rectangular, then it has  $\Pi$ -communication complexity at most 2.

► **Lemma 17** ([20]). *Let  $\varphi$  be an unsatisfiable CNF formula with  $n$  variables and  $\Pi = (X, Y)$  be a partition of  $[n]$  into two disjoint parts. Assume that  $\pi$  has a  $\mathcal{G}$ -refutation of size  $S$  and  $\mathcal{G}$  has  $\Pi$ -communication complexity at most  $c$ . Then there is a  $\Pi$ -rectangular set  $\mathcal{G}'$  such that  $\varphi$  has a  $\mathcal{G}'$ -refutation of size at most  $2^{3c}S$ .*

Notice that the set of all clauses is  $\Pi$ -rectangular for every partition  $\Pi$ . The set of  $\pi$ -OBDDs of size  $S$  has  $\Pi$ -communication complexity  $\log S + 1$  for partitions  $\Pi = (X, Y)$  where the variables of  $X$  precede the variables of  $Y$  in the order  $\pi$ .

In order to capture Cutting Planes we say that  $\mathcal{G}$  is  $\Pi$ -triangular if for every  $g \in \mathcal{G}$  there are functions  $a : \{0, 1\}^X \rightarrow \mathbb{R}$  and  $b : \{0, 1\}^Y \rightarrow \mathbb{R}$  such that  $g^{-1}(0) = \{x \in \{0, 1\}^X, y \in \{0, 1\}^Y \mid a(x) < b(y)\}$ . Note that the set of all linear inequalities with integer coefficients over Boolean variables is  $\Pi$ -triangular for every partition  $\Pi$ .

Let  $\text{Ind}_m : \{0, 1\}^{\lceil \log m \rceil} \times \{0, 1\}^m \rightarrow \{0, 1\}$  be a Boolean function such that  $\text{Ind}_m(z_1, \dots, z_{\lceil \log m \rceil}, y_1, \dots, y_m) = y_b$ , where  $b$  is the integer with binary representation  $z_1 \dots z_{\lceil \log m \rceil}$ .

► **Theorem 18** ([7]). *Let  $\varphi$  be an unsatisfiable CNF formula  $\varphi$  with  $n$  variables. Let  $m = n^\delta$ , where  $\delta$  is some global constant. Let  $\Pi = (X, Y)$  be the following partition of variables of  $\varphi \circ \text{Ind}_m$ : all  $z$ -variables go to  $X$ , all  $y$ -variables go to  $Y$ . If  $\mathcal{G}$  is  $\Pi$ -rectangular or  $\mathcal{G}$  is  $\Pi$ -triangular, then every  $\mathcal{G}$ -refutation of  $\varphi \circ \text{Ind}_m$  has size at least  $n^{\Omega(w(\varphi))}$ .*

► **Corollary 19.** *Under the conditions of Theorem 18, if  $\mathcal{G}$  has  $\Pi$ -communication complexity at most  $c$ , then every  $\mathcal{G}$ -refutation of  $\varphi \circ \text{Ind}_m$  has size at least  $2^{-3c}n^{\Omega(w(\varphi))}$ .*

**Proof.** By Lemma 17, if there is a  $\mathcal{G}$ -refutation of  $\varphi \circ \text{Ind}_m$  of size  $S$ , there exists a  $\mathcal{G}'$ -refutation of  $\varphi \circ \text{Ind}_m$  of size at most  $2^{3c}S$  such that  $\mathcal{G}'$  is  $\Pi$ -rectangular. By Theorem 18,  $2^{3c}S \geq n^{\Omega(w(\varphi))}$ , hence  $S \geq 2^{-3c}n^{\Omega(w(\varphi))}$ . ◀

► **Corollary 20.** *Under the conditions of Theorem 18, every Cutting Planes proof of  $\varphi \circ \text{Ind}_m$  has size at least  $n^{\Omega(w(\varphi))}$ .*

**Proof.** The statement follows from Theorem 18, since the set of linear inequalities is  $\Pi$ -triangular for every partition  $\Pi$ . ◀

### 4.3 Cutting Planes Does Not Polynomially Simulates OBDD( $\wedge$ )

► **Lemma 21.** *Both functions  $\text{Ind}_m$  and  $\neg \text{Ind}_m$  have  $\text{poly}(m)$ -constructible CNF representations.*

**Proof.** Let us consider the following formula for  $\text{Ind}_m$ ,

$$\bigwedge_{i=1}^m (\text{bin}(z_1, \dots, z_{\lceil \log m \rceil}) = i) \rightarrow y_i,$$

where  $\text{bin}(z_1, \dots, z_{\lceil \log m \rceil}) = i$  is the conjunction of literals stating that  $z_1, \dots, z_{\lceil \log m \rceil}$  is the binary representation of  $i$ . For  $\ell \in [m]$ , let  $\varphi_\ell$  be the formula  $\bigwedge_{i=1}^\ell (\text{bin}(z_1, \dots, z_{\lceil \log m \rceil}) = i) \rightarrow y_i$ , and let  $\varphi_m = \text{Ind}_m$ . We claim that for all  $\ell \in [m]$  the formula  $\varphi_\ell$  has an OBDD representation of size  $\text{poly}(m)$  in the order  $z_1, \dots, z_{\lceil \log m \rceil}, y_1, \dots, y_m$ . Indeed, such an OBDD has the following structure: it starts with the complete decision tree over all the variables  $z_i$ ; consider a leaf of this decision tree that corresponds to a number  $i$ . If  $i \leq \ell$ , then we add to this leaf a node of OBDD labeled with  $y_i$  and the outgoing edge labeled with 0 going to the

0-sink and the outgoing edge labeled with 1 going to the 1-sink. If  $i > \ell$ , then we identify this leaf with 1-sink. Hence, there is a  $\text{poly}(m)$ -constructible CNF representation of  $\text{Ind}_m$ .

The same argument works also for  $\neg\text{Ind}_m$ , since  $\neg\text{Ind}_m(z_1, \dots, z_{\lfloor \log m \rfloor}, y_1, y_2, \dots, y_m) = \text{Ind}_m(z_1, \dots, z_{\lfloor \log m \rfloor}, \neg y_1, \neg y_2, \dots, \neg y_m)$ .  $\blacktriangleleft$

► **Lemma 22.** *The formula  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  has at most  $m^{O(\log n)}$  clauses of size  $O(\log n \log m)$  and  $O(m \log^2 n)$  variables.*

**Proof.** Each clause of  $\text{TS}_{K_{\log n}, c}$  consists of  $\lfloor \log n \rfloor - 1$  literals and by Lemma 21 there is CNF representations of  $\text{Ind}_m$  and  $\neg\text{Ind}_m$  with  $m$  clauses. Hence, for each clause  $C$  of  $\text{TS}_{K_{\log n}, c}$ , the formula  $C \circ \text{Ind}_m$  has  $m^{\lfloor \log n \rfloor - 1}$  clauses each of length  $(\lfloor \log n \rfloor - 1)(\lfloor \log m \rfloor + 1)$ .  $\blacktriangleleft$

► **Theorem 23.** *Let  $\text{TS}_{K_{\log n}, c}$  be unsatisfiable Tseitin formula based on a complete graph  $K_{\log n}$  on  $\lfloor \log n \rfloor$  vertices.*

*Let  $m = (\log n)^{2\delta}$ , where  $\delta$  is the constant from Theorem 18. Then*

1.  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  has a tree-like OBDD( $\wedge$ ) proof of size  $(\log n)^{O(\log n)}$  and
2. every Cutting Planes proof of  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  has size at least  $(\log n)^{\Omega(\log^2 n)}$ .

**Proof.**

1. By Lemma 21, both  $\text{Ind}_m$  and  $\neg\text{Ind}_m$  are  $\text{poly}(m)$ -constructible. By Corollary 12, there is a tree-like OBDD( $\wedge$ ) refutation of  $\text{TS}_{K_{\log n}, c}$  of size  $\text{poly}(n)$ . By Lemma 22, the size of the formula  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  is at most  $m^{O(\log n)}$ . Hence, by Theorem 5, there is a tree-like OBDD( $\wedge$ ) refutation of  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  of size  $\text{poly}(\text{poly}(n), m^{O(\log n)}, \text{poly}(n)) = (\log n)^{O(\log n)}$ .
2. By Corollary 16,  $w(\vdash \text{TS}_{K_{\log n}, c}) = \Omega(\log^2 n)$ . Hence, by Corollary 20, every Cutting Planes proof of  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  has size at least  $(\log^2 n)^{\Omega(\log^2 n)} = (\log n)^{\Omega(\log^2 n)}$ .  $\blacktriangleleft$

#### 4.4 OBDD( $\wedge$ , weakening) Does Not Polynomially Simulate OBDD( $\wedge$ , reordering)

► **Theorem 24.** *There is a family of formulas  $\varphi_n$  such that:*

- *the size of  $\varphi_n$  is  $(\log n)^{O(\log n \log \log n)}$  and number of variables in  $\varphi_n$  is  $\text{poly}(\log n)$ ;*
- *there is a tree-like OBDD( $\wedge$ , reordering) proof of  $\varphi_n$  of size  $(\log n)^{O(\log n \log \log n)}$ ;*
- *every OBDD( $\wedge$ , weakening) proof of  $\varphi_n$  has size at least  $(\log n)^{\Omega(\log^2 n)}$ .*

► **Lemma 25.** *Let  $\text{TS}_{K_{\log n}, c}$  be an unsatisfiable Tseitin formula. Let  $m = (\log n)^{2\delta}$ , where  $\delta$  is the constant from Theorem 18.*

*There is a family of orders  $\{\pi_n\}_{n \in \mathbb{N}}$  over the variables of the formulas  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  such that every  $\pi_n$ -OBDD( $\wedge$ , weakening) proof of  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  has size at least  $(\log n)^{\Omega(\log^2 n)}$ .*

**Proof.** Let  $\pi_n$  be an order on variables of  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$ , where all  $z$ -variables precedes all  $y$ -variables. Consider some  $\pi_n$ -OBDD( $\wedge$ , weakening) proof of  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$ ; let  $S$  denote its total size. Hence, the number of proof lines and sizes of all OBDDs are at most  $S$ . Consider a partition  $\Pi = (X, Y)$  of the variables of  $\text{TS}_{K_{\log n}, c} \circ \text{Ind}_m$  such that  $X$  contains all  $z$ -variables and  $Y$  contains all  $y$ -variables. The communication complexity of computing an OBDD of size  $S$  w.r.t. the partition  $\Pi$  is at most  $\log S + 1$ . Therefore, the  $\pi_n$ -OBDD( $\wedge$ , weakening) proof can be viewed as a  $\mathcal{G}$ -refutation, where  $\mathcal{G}$  has  $\Pi$ -communication complexity at most  $\log S + 1$ . Hence, by Corollary 19,  $S \geq 2^{-3 \log S - 3} (\log^2 n)^{\Omega(\log^2 n)}$ . Thus,  $S \geq (\log^2 n)^{\Omega(\log^2 n)} = (\log n)^{\Omega(\log^2 n)}$ .  $\blacktriangleleft$

**Proof of Theorem 24.** Let  $\text{TS}_{K_{\log n, c}}$  be an unsatisfiable Tseitin formula. Let  $m = (\log n)^{2\delta}$ , where  $\delta$  is the constant from Theorem 18.

Let us consider  $\varphi_n = \mathcal{T}(\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m)$ , where  $\mathcal{T}$  is the transformation defined in Section 3.1. By Corollary 12, there is a tree-like OBDD( $\wedge$ ) proof of  $\text{TS}_{K_{\log n, c}}$  of size  $\text{poly}(n)$ . By Lemma 22,  $\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m$  has  $\log n^{O(\log n)}$  clauses of size  $O(\log n \log \log n)$  and  $\text{poly}(\log n)$  variables. By Lemma 21,  $\text{Ind}_m$  is  $\text{poly}(m)$ -constructible; hence, by Theorem 5, there is a tree-like OBDD( $\wedge$ ) proof of  $\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m$  of size  $\log n^{O(\log n)}$ .

Recall that  $\varphi_n = \mathcal{T}(\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m) = \text{perm}((\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m)^{\vee k})$ , where  $k = \text{poly}(\log n)$ .

The formula  $(\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m)^{\vee k}$  has size  $(\log n)^{O(\log n \log \log n)}$ ; by Theorem 5 there is a tree-like OBDD( $\wedge$ ) proof of  $(\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m)^{\vee k}$  of size  $(\log n)^{O(\log n \log \log n)}$ .

Thus, by Lemma 2, there is a tree-like OBDD( $\wedge$ , reordering) proof of  $\mathcal{T}(\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m)$  of size  $(\log n)^{O(\log n \log \log n)}$ .

Note that, by Lemma 25 and Lemma 1, every OBDD( $\wedge$ , weakening) proof of  $\mathcal{T}(\text{TS}_{K_{\log n, c}} \circ \text{Ind}_m)$  has size at least  $(\log^2 n)^{\Omega(\log^2 n)} = (\log n)^{\Omega(\log^2 n)}$ .  $\blacktriangleleft$

## 5 Exponential Separations for Tree-like Case

In this section we exhibit a formula which is hard for tree-like OBDD( $\wedge$ , weakening) and easy for tree-like OBDD( $\wedge$ , reordering) in another order. An example of such a formula can be obtained from a construction of Göös and Pitassi [8]. We use a pebbling contradiction as the base of our example.

► **Definition 26.** Let  $G$  be a directed acyclic graph with one sink  $t$ . The CNF formula  $\text{Peb}_G$  (pebbling contradiction for a graph  $G$ ), uses a variable  $x_v$  for each vertex  $v$  of  $G$  and has the following clauses:

- $\neg x_t$ ;
- for each vertex  $v$ , the clause  $x_v \vee \bigvee_{i=1}^d \neg x_{p_i}$  where  $p_1, \dots, p_d$  are all the immediate predecessors of  $v$  ( $d = 0$  if  $v$  is a source).

It is not hard to see that  $\text{Peb}_G$  has short tree-like OBDD( $\wedge$ ) proofs:

► **Theorem 27.** For any directed acyclic graph  $G(V, E)$  with  $n$  vertices and maximum in-degree  $d$  there is a tree-like OBDD( $\wedge$ ) proof of  $\text{Peb}_G$  of size  $\text{poly}(n)$ .

**Proof.** For a vertex  $v \in V$ , we let  $p_{v,1}, \dots, p_{v,l_v}$  be the immediate predecessors of  $v$ . For any set  $S \subseteq V$  such that if  $v \in S$ , then  $p_{v,1}, \dots, p_{v,l_v}$  are also in  $S$  (we call such a set closed under predecessors), the formula  $\bigwedge_{v \in S} \left( x_v \vee \bigvee_{i=1}^{l_v} \neg x_{p_{v,i}} \right)$  is equivalent to  $\bigwedge_{v \in S} x_v$ . Thus

$\bigwedge_{v \in S} \left( x_v \vee \bigvee_{i=1}^{l_v} \neg x_{p_{v,i}} \right)$  has an OBDD representation of size  $\text{poly}(n, d)$ .

Let  $v_1, \dots, v_n$  be a topological ordering of vertices of  $G$ . Consider an order  $\pi$  and a sequence  $D_1, \dots, D_{n+1}$  of  $\pi$ -OBDDs such that  $D_i$  represents the formula  $\bigwedge_{j=1}^i \left( x_{v_j} \vee \bigvee_{k=1}^{l_{v_j}} \neg x_{p_{v_j,k}} \right)$  for all  $1 \leq i \leq n$  and  $D_{n+1}$  is the constant false diagram. We claim that, together with  $\pi$ -OBDDs representing the initial clauses,  $D_1, \dots, D_{n+1}$  is an OBDD( $\wedge$ ) refutation of  $\text{Peb}_G$  of total size  $O(n^2)$ . Indeed, since for all  $i \in [n]$  the set  $\{v_1, v_2, \dots, v_i\}$  is closed under predecessors,  $D_i = \bigwedge_{j=1}^i x_{v_j}$  has size  $2i + 2$ . It is easy to see that  $D_{i+1}$  is equal to

$$D_i \wedge \left( x_{v_{i+1}} \vee \bigvee_{i=1}^{l_{v_{i+1}}} \neg x_{p_{v_{i+1},i}} \right). \quad \blacktriangleleft$$

► **Corollary 28** (Lemma 2, [12]). *For any directed acyclic graph  $G(V, E)$  with  $n$  vertices and maximum in-degree  $d$  there is a tree-like OBDD( $\wedge$ ) proof of  $\text{Peb}_G^{\vee 2}$  of size  $\text{poly}(n, 2^d)$ .*

**Proof.** Since  $\text{Peb}_G$  is a formula in  $(d + 1)$ -CNF, size of the formula  $\text{Peb}_G^{\vee 2}$  is at most  $O(|\text{Peb}_G|2^d)$ . The Corollary follows from Theorem 27 and Theorem 5. ◀

Corollary 28 was presented earlier as [12, Lemma 2], however, there was a flaw in previous proof. The proof of [12, Lemma 2] was based on the following statement ([12, Lemma 1]): Let  $G$  be a dag on  $n$  nodes, and  $j$  be a node in  $G$  with parents  $i_1, \dots, i_k$  where  $k = O(\log n)$ . Consider the clauses  $(x_{i_1,0} \vee x_{i_1,1}), \dots, (x_{i_k,0} \vee x_{i_k,1})$  and  $(\neg x_{i_1,a_1} \vee \dots \vee \neg x_{i_k,a_k} \vee x_{j,0} \vee x_{j,1})$  for all  $(a_1, \dots, a_k) \in \{0, 1\}^k$ . For any variable order  $\pi$ , there is a polynomial-size  $\pi$ -OBDD( $\wedge$ ) derivation of  $x_{j,0} \vee x_{j,1}$  from these clauses. However, [12, Lemma 1] is incorrect, for example for  $k = 1$  it claims that it is possible to derive  $(a \vee b)$  from  $A = \{(\neg x \vee a \vee b), (\neg y \vee a \vee b), (x \vee y)\}$  in OBDD( $\wedge$ ). Assume that  $(a \vee b)$  is the conjunction of clauses from  $B \subseteq A$ . Notice that  $(x \vee y) \notin B$ , since otherwise it would be possible to satisfy  $(a \vee b)$  by substitution  $x := 0, y := 0$ . It is easy to see that  $B$  can not be empty, hence  $B$  is non empty subset of  $\{(\neg x \vee a \vee b), (\neg y \vee a \vee b)\}$ . In this case it should be possible to satisfy  $a \vee b$  by substitution  $x := 0, y := 0$ . Thus, [12, Lemma 1] is incorrect.

Järvisalo [12] used Corollary 28 in order to give a family of formulas that are easy for OBDD( $\wedge$ ) but hard for tree-like Resolution. The lower bound was proved by Buresh-Oppenheimer and Pitassi [5], who proved that there is a family of graphs  $\{G_n\}_{n \in \mathbb{N}}$  with  $n$  vertices and maximum in-degree 2 such that any tree-like resolution proof of  $\varphi_n = \text{Peb}_{G_n}^{\vee 2}$  has size at least  $2^{\Omega(n/\log(n))}$ .

Let  $\varphi(x_1, \dots, x_n, y_1, \dots, y_n) = \bigwedge_{i=1}^m C_i(x_1, \dots, x_n, y_1, \dots, y_n)$ . The relation  $\text{Search}_\varphi \subseteq \{0, 1\}^n \times \{0, 1\}^n \times [m]$  is defined by

$$(x, y, i) \in \text{Search}_\varphi \text{ iff } C_i(x_1, \dots, x_n, y_1, \dots, y_n) = 0.$$

Consider the following communication game: Alice knows values of variables  $x_1, x_2, \dots, x_n$  and Bob knows variables  $y_1, y_2, \dots, y_n$ . The goal of the communication game is to compute some  $i \in [m]$  such that  $(x_1, \dots, x_n, y_1, \dots, y_n, i) \in \text{Search}_\varphi$ .

Göös and Pitassi [8] proved the following theorem:

► **Theorem 29** ([8]). *There are a family of directed acyclic graphs  $\{G_n\}_{n \in \mathbb{N}}$  with constant degree such that  $G_n$  has  $n$  vertices, and a CNF formula  $g$  on variables  $x_1, x_2, y_1, y_2$  such that the deterministic communication complexity of  $\text{Search}_{\text{Peb}_{G_n} \circ g}$  is at least  $\Omega(\sqrt{n})$  if Alice knows variables  $\{x_{1,1}, x_{1,2}, \dots, x_{n,1}, x_{n,2}\}$  and Bob knows variables  $\{y_{1,1}, y_{1,2}, \dots, y_{n,1}, y_{n,2}\}$ .*

In fact Theorem 29 is true even for randomized communication complexity, but the deterministic version is enough for our applications.

► **Lemma 30.** *Let a function  $f$  be computed by a  $\pi$ -OBDD  $D$ , the communication complexity of  $f$  under a partition  $\Pi_0, \Pi_1$  of the variables where the variables in  $\Pi_0$  precede (in the sense of  $\pi$ ) the variables from  $\Pi_1$  is at most  $\lceil \log |D| \rceil + 1$ .*

**Proof.** Alice starts the computation of  $f$  according  $D$  using her variables. Finally Alice reaches vertex  $v$  of  $D$  reading all her variables. Alice sends to Bob number of the vertex  $v$ , it has at most  $\lceil \log |D| \rceil$  bits. Bob continues computing  $f$  starting from  $v$  using his variables and sends the result of the computation (it is 1 bit) to Alice. ◀

► **Theorem 31.** *Let  $\varphi(x_1, \dots, x_n, y_1, \dots, y_n)$  be an unsatisfiable CNF formula. Suppose the the communication complexity of the relation  $\text{Search}_\varphi$  is equal to  $t$  if Alice knows the values*

of variables  $x_i$  and Bob knows the variables  $y_i$ . Let  $\pi$  be an ordering of the variables of  $\varphi$  such that variables  $x_i$  precede variables  $y_i$ . Then the size of any tree-like  $\pi$ -OBDD( $\wedge$ , weakening) refutation of  $\varphi$  is at least  $2^{O(\sqrt{\ell})}$ .

**Proof.** Consider a tree-like  $\pi$ -OBDD( $\wedge$ , weakening) proof  $D_1, \dots, D_\ell$  of the formula  $\varphi$  of size  $S$ . Based on this proof we construct a communication protocol for  $\text{Search}_\varphi$  of complexity at most  $O(\log^2 S)$ . The protocol consists of  $\ell = O(\log S)$  steps. At each step we consider some tree  $T_i$  that is known by both players. The inner vertices of the tree are labelled with  $\pi$ -OBDDs and the leaves are labelled with clauses of  $\varphi$  or with trivially satisfied clauses. In the first step, the tree  $T_1$  is the tree of our tree-like proof.  $T_i \subseteq T_{i-1}$ . At each step, the two players know that the clause at the root of  $T_i$  is falsified by the input assignment, and that there exists some clause at a leaf of  $T_i$  that is falsified. In the end, the tree  $T_\ell$  consists of a single vertex; hence it provides clause of  $\varphi$  that is falsified by the input assignment.

Now we describe how we obtain the tree  $T_{i+1}$  from the tree  $T_i$ . Let  $v$  be a vertex of  $T_i$  such that a subtree  $T'$  with root  $v$  satisfies the following condition:  $\frac{1}{3}|T_i| \leq |T'| \leq \frac{2}{3}|T_i|$  (such a vertex  $v$  players can find without communication). Let  $D$  be the OBDD labelling  $v$ ; if the input assignment evaluates diagram  $D$  to zero, then  $T_{i+1}$  equals  $T'$ . The players can evaluate the  $\pi$ -OBDD  $D$  on the input assignment with at most  $\lceil \log |D| \rceil + 1 \leq 2 \log S$  bits of communication by Lemma 30. Otherwise,  $T_{i+1} := T_i \setminus T'$ .

It is easy to see that if the value of  $D$  equals zero then there is a leaf with falsified clause in the tree  $T'$ . Otherwise there is a leaf with falsified clause in the tree  $T_i \setminus T'$ . Also, at each step the players use at most  $2 \log(S)$  bits of communication and there are at most  $O(\log(S))$  steps (since  $|T_i| \leq \frac{2}{3}|T_{i+1}|$ ). Hence, the players use at most  $O(\log^2 S)$  bits of communication. Therefore  $S = 2^{\Omega(\sqrt{\ell})}$ .  $\blacktriangleleft$

As a result we obtain the following separation.

- **Theorem 32.** *There are a family of formulas  $\varphi_n$  in CNF and a constant  $c > 0$  such that:*
- *size of  $\varphi_n$  and number of variables in  $\varphi_n$  are polynomially bounded by  $n$ ;*
  - *there is a tree-like OBDD( $\wedge$ , reordering) proof of  $\varphi_n$  of size polynomial in  $n$ ;*
  - *any tree-like OBDD( $\wedge$ , weakening) proof of  $\varphi_n$  has size at least  $2^{\Omega(n^{1/4})}$ .*

**Proof.** Let  $g$  be a CNF formula on the variables  $x_1, x_2, y_1, y_2$  and let  $\{G_n\}_{n \in \mathbb{N}}$  be a family of graphs so that Theorem 29 holds. Consider the formula  $\psi_n = \text{Peb}_{G_n} \circ g$ . By Theorem 29 and Theorem 31 there exists an order  $\pi$  such that the size of every tree-like  $\pi$ -OBDD( $\wedge$ , weakening) refutation of  $\psi_n$  has size at least  $2^{O(n^{1/4})}$ . By Lemma 1 any tree-like OBDD( $\wedge$ , weakening) proof of the formula  $\varphi_n := \mathcal{T}(\psi_n)$  has size  $2^{\Omega(n^{1/4})}$ .

By Theorems 27 and 5,  $\psi_n$  has a tree-like OBDD( $\wedge$ ) proof of size  $\text{poly}(n)$ . Then, by Lemma 2, there is a OBDD( $\wedge$ , reordering) proof of  $\mathcal{T}(\psi_n)$  of size  $\text{poly}(n)$ .  $\blacktriangleleft$

## 6 Clique-Coloring is Easy for OBDD( $\wedge$ , weakening)

In this section we prove Theorem 8. Let  $\pi$  be the following order on the variables of  $\text{Clique-Coloring}_{n,m}$ :

$$p_{1,1}, \dots, p_{n,n}, q_{1,1}, \dots, q_{m,1}, r_{1,1}, \dots, r_{1,m}, \\ q_{1,2}, \dots, q_{m,2}, r_{2,1}, \dots, r_{2,m}, \dots, q_{1,n}, \dots, q_{m,n}, r_{n,1}, \dots, r_{n,m}.$$

This order places at the beginning the variables encoding a graph, after them the variables encoding the number of the first vertex in clique, after them the variables encoding the color of the first vertex and so on. All OBDDs used in this section are  $\pi$ -OBDDs.

► **Lemma 33.** *For any integer constants  $c, c_q, c_r$ , and sets  $I \subseteq [n]$ ,  $K \subseteq [m]$ , and  $L \subseteq [m-1]$  the inequality*

$$\sum_{i \in I} \left( \sum_{k \in K} q_{k,i} - c_q \right) \left( \sum_{l \in L} r_{i,l} - c_r \right) \geq c \quad (1)$$

has a  $\pi$ -OBDD representation of size polynomial in  $c_r, c_q, m$ , and  $n$ .

**Proof.** The order  $\pi$  was picked to make it convenient to evaluate the left hand side of (1) with a  $\pi$ -OBDD. The OBDD is constructed in levels, one level per variable. Each level has vertices corresponding to the values of partial sums used to compute the left hand side of (1). Specifically, let  $Q_{i,k} = \sum_{k' \in K, k' \leq k} (q_{k',i} - c_q)$ , let  $R_{i,l} = \sum_{l' \in L, l' \leq l} (r_{i,l'} - c_r)$ , and let  $S_i = \sum_{i' \in I, i' < i} Q_{i,m+1} R_{i,m}$ . Note  $S_{1+\max(I)}$  equals the left hand side of (1).

The vertices of the OBDD at the level corresponding to a variable  $q_{k,i}$  encode the values of  $S_i$  and  $Q_{i,k}$ . The vertices at the level corresponding to a variable  $r_{i,l}$  encode the values of  $S_i, Q_{i,m+1}$ , and  $R_{i,l}$ . The number of possible values at each level is polynomially bounded by  $c_r, c_q, m, n$ . To finalize the  $\pi$ -OBDD for evaluating (1), the vertices in the final level that correspond to a value  $\geq c$  are sinks labeled with 1, and the remaining vertices in the final level are sinks with label 0. ◀

**Proof of Theorem 8.** The idea of the proof is to first derive a  $\pi$ -OBDD which represents the inequality  $\sum_{k,i,l} q_{k,i} r_{i,l} \geq m$ , stating that every vertex of clique is colored, and second to derive a  $\pi$ -OBDD which represents the inequality  $\sum_{k,i,l} q_{k,i} r_{i,l} \leq m-1$  stating roughly that there is at most one vertex per color. Combining these with conjunction derives a contradiction.

1. We first describe the derivation of the OBDD representing  $\sum_{k,i,l} q_{k,i} r_{i,l} \geq m$ . For  $i \in [n]$ ,

the derivation starts with an OBDD representing the inequality  $\sum_{l=1}^{m-1} r_{i,l} \geq 1$ ; note that **Clique-Coloring** $_{n,m}$  has such a clause. For each  $k \in m$ , using the weakening rule (in fact multiplying the inequality by  $q_{k,i}$ ) gives an OBDD that represents the inequality

$$\sum_{l=1}^{m-1} q_{k,i} r_{i,l} \geq q_{k,i}. \quad (2)$$

Since this is equivalent to  $q_{k,i} \sum_{l=1}^{m-1} (r_{i,l} - 1) \geq 0$ , Lemma 33 implies that the OBDD representing (2) has polynomial size. Summing the inequalities (2) for all  $i \in [n]$  gives

$$\sum_{i=1}^n \sum_{l=1}^{m-1} q_{k,i} r_{i,l} \geq \sum_{i=1}^n q_{k,i}. \quad (3)$$

To derive an OBDD representation of the inequality (3) for a fixed value of  $k$ , we add the inequalities (2) for  $i \in [n]$  one by one. The addition of two inequalities may be expressed by a conjunction followed by a weakening rule. The intermediate inequalities can be expressed as  $\sum_{i=1}^u q_{k,i} \sum_{l=1}^{m-1} (r_{i,l} - 1) \geq 0$ ; hence by Lemma 33, they have OBDD representations of size  $\text{poly}(n, m)$ . This allows the derivation of polynomial size OBDDs representing (3) for each  $k$ .



The inequality  $\sum_{i=1}^n q_{k,i} \geq 1$  is expressed by a clause of **Clique-Coloring** $_{n,m}$ ; combining this with the inequality (3) using the conjunction and weakening rules gives an OBDD representing

$$\sum_{i=1}^n \sum_{l=1}^{m-1} q_{k,i} r_{i,l} \geq 1. \quad (4)$$

The size of an OBDD representation of (4) is polynomially bounded, again by Lemma 33. Finally, to get the desired inequality  $\sum_{k,i,l} q_{k,i} r_{i,l} \geq m$  we sum the inequalities (4) for all  $k \in [m]$ . As in the previous cases, we do this iteratively, combining the inequalities (4) one by one with the conjunction and weakening rules. The intermediate OBDDs are  $\sum_{k < u} \sum_{i,l} q_{k,i} r_{i,l} \geq u$  and are polynomially bounded by Lemma 33.

2. The second part derives an OBDD representation of the inequality  $\sum_{k,i,l} q_{k,i} r_{i,l} \leq m - 1$ .

If we derive

$$\sum_{k=1}^m \sum_{i=1}^n q_{k,i} r_{i,l} \leq 1 \quad (5)$$

for each  $l \in [m - 1]$  and sum them as we do earlier we get the desired inequality. All intermediate inequalities have small OBDD representations by Lemma 33.

For each  $l$ , the inequality (5) will be derived from the inequalities (6) and (9) as described below. For  $k \in [m]$ , we derive (an OBDD representing) the inequality (6)

$$\sum_{i=1}^n q_{k,i} r_{i,l} \leq 1. \quad (6)$$

stating that there is at most one vertex with number  $k$  in clique which has color  $l$ . The inequality (6) follows by weakening from the inequality

$$\sum_{i=1}^n q_{k,i} \leq 1. \quad (7)$$

To derive (7), we derive inequalities  $\sum_{i=1}^u q_{k,i} \leq 1$  for all  $u \in [n]$ . For  $u = n$  this inequality is the same as (7). For  $u = 1$  this inequality is the constant true statement. For  $u + 1$  it is a weakening of the conjunction of  $\sum_{i=1}^u q_{k,i} \leq 1$  and

$$\bigwedge_{i=1}^u (q_{k,i} + q_{k,u+1} \leq 1). \quad (8)$$

Each inequality  $q_{k,i} + q_{k,u+1} \leq 1$  is a clause of **Clique-Coloring** $_{n,m}$  but we need to check that their  $u$ -fold conjunctions (8) have polynomial size OBDD derivations. For this, we iteratively derive  $\bigwedge_{i=1}^t (q_{k,i} + q_{k,u+1} \leq 1)$  for all  $t \in [u]$ . For each  $t$ , this inequality has

a small OBDD representation since it is equivalent to  $\left( \bigvee_{i=1}^t q_{k,i} \right) \rightarrow \neg q_{k,u+1}$ ; the latter clearly has a polynomial size OBDD representation. Thus there are short refutations of constraints (8) and as a result, of inequalities (7) and (6).

To derive (5), we also need

$$\sum_{i=1}^n q_{k,i} r_{i,l} + \sum_{i=1}^n q_{k',i} r_{i,l} \leq 1 \quad (9)$$

for all  $k \neq k' \in [m]$ . Before deriving inequality (9) we show how to derive (5) from (6) and (9). This derivation is similar to derivation of (7) but it is slightly more complicated to show that all intermediate inequalities have polynomial size OBDD representations. To derive (5), we derive successively the inequalities

$$\sum_{k=1}^u \sum_{i=1}^n q_{k,i} r_{i,l} \leq 1. \quad (10)$$

for all  $u \in [n]$ . Each inequality (10) has a polynomial size OBDD representation by Lemma 33. For  $u = 1$ , (10) is the same as (6). Let us show how to derive inequality (10) for  $u + 1$  from the inequality (10) for  $u$ . For this, it suffices to derive the inequality

$$\bigwedge_{k=1}^u \left( \sum_{i=1}^n q_{k,i} r_{i,l} + \sum_{i=1}^n q_{u+1,i} r_{i,l} \leq 1 \right) \quad (11)$$

and then use the conjunction and weakening rules. Each inequality from the conjunction is an instance of inequality (9). We must show the conjunction (11) has a small derivation.

To derive (11), we iteratively derive  $\bigwedge_{k=1}^t \left( \sum_{i=1}^n q_{k,i} r_{i,l} + \sum_{i=1}^n q_{u+1,i} r_{i,l} \leq 1 \right)$  for all  $t \in [u]$ .

This conjunction is equal to  $\bigvee_{k=1}^t \bigvee_{i=1}^n q_{k,i} \wedge r_{i,l} \rightarrow \neg \bigvee_{i=1}^n q_{u+1,i} \wedge r_{i,l}$ . Hence it has a small OBDD representation by the choice of  $\pi$ .

We conclude the proof of Theorem 8 by proving the inequality (9) for  $k$  and  $k'$ . For this we will first derive the inequalities

$$\begin{aligned} \sum_{i=1}^t q_{k,i} r_{i,l} = 0 \vee \sum_{i=1}^t q_{k',i} r_{i,l} = 0 \vee \\ \bigvee_{i \in [t]} \left( q_{k,i} r_{i,l} = q_{k',i} r_{i,l} = 1 \wedge \bigwedge_{j \in [n] \setminus \{i\}} (q_{k,j} r_{j,l} = q_{k',j} r_{j,l} = 0) \right) \end{aligned} \quad (12)$$

for all  $t \in [n]$ . The inequality (12) for  $t = n$  and the conjunction  $\bigwedge_{i=1}^n \neg q_{k,i} \vee \neg q_{k',i}$  implies

$$\sum_{i=1}^n q_{k,i} r_{i,l} = 0 \vee \sum_{i=1}^n q_{k',i} r_{i,l} = 0. \quad (13)$$

Each clause in the conjunction  $\bigwedge_{i=1}^n \neg q_{k,i} \vee \neg q_{k',i}$  is a clause of **Clique-Coloring** $_{n,m}$ . The conjunction derived iteratively using the conjunction and weakening rules; all intermediate constraints have polynomial sized  $\pi$ -OBDD representations since  $\pi$  orders the variables  $q_{k,i}$  first by  $i$  and second by  $k$ .

The constraint (13) and the two inequalities (6) for  $k, l$  and for  $k', l$  imply (9). The constraint (12) is derived from the inequalities

$$q_{k,i} r_{i,l} + q_{k',j} r_{j,l} \leq 1 \quad (14)$$

for  $i \neq j \in [n]$ .

The inequality (12) is equivalent to the conjunction of inequalities (14) for all  $i \neq j \in [t]$ , and it is clear that these have polynomial size  $\pi$ -OBDD representations. We show there is a small OBDD derivation of this conjunction, that is, of (12), by deriving it for successive values of  $t$ . For  $t = 0$ , (12) is the constant true statement. We claim there is a short derivation of (12) for  $t = u + 1$  from (12) for  $t = u$ . Indeed, (14) together with (12) for  $t = u$  implies  $\bigwedge_{i=1}^u (q_{k,i}r_{i,l} + q_{k',u+1}r_{u+1,l} \leq 1)$ . It is easy to see that this latter inequality has a small OBDD representation since it is equivalent to the constraint  $\left(\bigvee_{i=1}^u q_{k,i}r_{i,l} = 1\right) \rightarrow q_{k',u+1}r_{u+1,l} = 0$ .

Now the only thing left to derive is the inequality (14).  $\text{Clique-Coloring}_{n,m}$  contains the clauses  $\neg q_{k,i} \vee \neg q_{k',j} \vee p_{i,j}$  and  $\neg p_{i,j} \vee \neg r_{i,l} \vee \neg r_{j,l}$ . From these, we can derive (14) using the conjunction rule and the weakening rules.  $\blacktriangleleft$

---

## References

- 1 Albert Atserias, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint propagation as a proof system. In Mark Wallace, editor, *Principles and Practice of Constraint Programming - CP 2004*, volume 3258 of *Lecture Notes in Computer Science*, pages 77–91. Springer, 2004. doi:10.1007/978-3-540-30201-8\_{ }9.
- 2 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.
- 3 Randal E. Bryant. Symbolic manipulation of boolean functions using a graphical representation. In Hillel Ofek and Lawrence A. O’Neill, editors, *Proceedings of the 22nd ACM/IEEE conference on Design automation, DAC 1985, Las Vegas, Nevada, USA, 1985.*, pages 688–694. ACM, 1985. doi:10.1145/317825.317964.
- 4 Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. Symbolic model checking:  $10^{20}$  states and beyond. *Inf. Comput.*, 98(2):142–170, 1992. doi:10.1016/0890-5401(92)90017-A.
- 5 Joshua Buresh-Oppenheimer and Toniann Pitassi. The complexity of resolution refinements. *J. Symb. Log.*, 72(4):1336–1352, 2007. doi:10.2178/jsl/1203350790.
- 6 Wei Chen and Wenhui Zhang. A direct construction of polynomial-size OBDD proof of pigeon hole problem. *Inf. Process. Lett.*, 109(10):472–477, 2009. doi:10.1016/j.ipl.2009.01.006.
- 7 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:175, 2017. URL: <https://ecc.ecc.weizmann.ac.il/report/2017/175>.
- 8 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 847–856. ACM, 2014. doi:10.1145/2591796.2591838.
- 9 Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. Complexity of semi-algebraic proofs. In Helmut Alt and Afonso Ferreira, editors, *STACS 2002, 19th Annual Symposium on Theoretical Aspects of Computer Science, Antibes - Juan les Pins, France, March 14-16, 2002, Proceedings*, volume 2285 of *Lecture Notes in Computer Science*, pages 419–430. Springer, 2002. doi:10.1007/3-540-45841-7\_34.
- 10 Jan Friso Groote and Hans Zantema. Resolution and binary decision diagrams cannot simulate each other polynomially. *Discrete Applied Mathematics*, 130(2):157–171, 2003. doi:10.1016/S0166-218X(02)00403-1.

- 11 Dmitry Itsykson, Alexander Knop, Andrei E. Romashchenko, and Dmitry Sokolov. On obdd-based algorithms and proof systems that dynamically change order of variables. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPICs*, pages 43:1–43:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi: 10.4230/LIPICs.STACS.2017.43.
- 12 Matti Järvisalo. On the relative efficiency of DPLL and obdds with axiom and join. In Jimmy Ho-Man Lee, editor, *Principles and Practice of Constraint Programming - CP 2011 - 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, volume 6876 of *Lecture Notes in Computer Science*, pages 429–437. Springer, 2011. doi:10.1007/978-3-642-23786-7\_33.
- 13 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.
- 14 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/js1/1208358751.
- 15 Kenneth L. McMillan. *Symbolic model checking*. Kluwer, 1993.
- 16 Christoph Meinel and Anna Slobodova. On the complexity of constructing optimal ordered binary decision diagrams. In *Proceedings of Mathematical Foundations of Computer Science*, volume 841, pages 515–524, 1994.
- 17 Guoqiang Pan and Moshe Y. Vardi. Search vs. symbolic techniques in satisfiability solving. In *7th International Conference on Theory and Applications of Satisfiability Testing, SAT 2004, Revised Selected Papers*, volume 3542, pages 235–250, 2005. doi: 10.1007/11527695\_{\_}19.
- 18 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997.
- 19 Nathan Segerlind. On the relative efficiency of resolution-like proofs and ordered binary decision diagram proofs. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA*, pages 100–111. IEEE Computer Society, 2008. doi:10.1109/CCC.2008.34.
- 20 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.
- 21 Olga Tveretina, Carsten Sinz, and Hans Zantema. Ordered binary decision diagrams, pigeonhole formulas and beyond. *JSAT*, 7(1):35–58, 2010. URL: [http://jsat.ewi.tudelft.nl/content/volume7/JSAT7\\_3\\_Tveretina.pdf](http://jsat.ewi.tudelft.nl/content/volume7/JSAT7_3_Tveretina.pdf).
- 22 Tomás E. Uribe and Mark E. Stickel. Ordered binary decision diagrams and the davisputnam procedure. In Jean-Pierre Jouannaud, editor, *Constraints in Computational Logics, First International Conference, CCL'94, Munich, Germany, September 7-9, 1994*, volume 845 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 1994. doi:10.1007/BFb0016843.
- 23 Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987. doi: 10.1145/7531.8928.

# Testing Linearity against Non-Signaling Strategies

**Alessandro Chiesa**

UC Berkeley, Berkeley (CA), USA  
alexch@berkeley.edu

**Peter Manohar**

UC Berkeley, Berkeley (CA), USA  
manohar@berkeley.edu

**Igor Shinkar**

UC Berkeley, Berkeley (CA), USA  
igors@berkeley.edu

---

## Abstract

Non-signaling strategies are collections of distributions with certain non-local correlations. They have been studied in Physics as a strict generalization of quantum strategies to understand the power and limitations of Nature’s apparent non-locality. Recently, they have received attention in Theoretical Computer Science due to connections to Complexity and Cryptography.

We initiate the study of Property Testing against non-signaling strategies, focusing first on the classical problem of *linearity testing* (Blum, Luby, and Rubinfeld; JCSS 1993). We prove that any non-signaling strategy that passes the linearity test with high probability must be close to a *quasi-distribution* over linear functions.

Quasi-distributions generalize the notion of probability distributions over global objects (such as functions) by allowing negative probabilities, while at the same time requiring that “local views” follow standard distributions (with non-negative probabilities). Quasi-distributions arise naturally in the study of Quantum Mechanics as a tool to describe various non-local phenomena.

Our analysis of the linearity test relies on Fourier analytic techniques applied to quasi-distributions. Along the way, we also establish general equivalences between non-signaling strategies and quasi-distributions, which we believe will provide a useful perspective on the study of Property Testing against non-signaling strategies beyond linearity testing.

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

**Keywords and phrases** property testing, linearity testing, non-signaling strategies, quasi-distributions

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.17

**Related Version** Full version is available on the Electronic Colloquium on Computational Complexity as TR18-067, <https://eccc.weizmann.ac.il/report/2018/067/>.

**Funding** This work was supported by the UC Berkeley Center for Long-Term Cybersecurity.

**Acknowledgements** We are grateful to Aneesh Manohar for helpful discussions on the prior uses of quasi-distributions in quantum mechanics. We thank Tom Gur and Thomas Vidick for useful discussions and suggestions that have improved the presentation in this paper. We thank anonymous reviewers who brought [47, 42, 2] to our attention, encouraged us to also explore statements for non-signaling players, and provided other valuable feedback.



© Alessandro Chiesa, Peter Manohar, and Igor Shinkar;  
licensed under Creative Commons License CC-BY  
33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 17; pp. 17:1–17:37

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



## 1 Introduction

Property Testing studies sublinear-time algorithms for approximate decision problems. A *tester* is an algorithm that receives oracle access to an input, samples a small number of locations, queries the input at these locations, and then decides whether to accept or reject. If the input has a certain property, the tester must accept with high probability; if instead the input is far from all inputs having this property, then the tester must reject with high probability.

Seminal works in Property Testing include those of Blum, Luby, and Rubinfeld [15], who studied the problem of deciding whether the input is the evaluation table of a linear function or is far from any such table, and of Rubinfeld and Sudan [45], who studied the analogous problem for low-degree functions. Property Testing for general decision problems was introduced in the foundational work of Goldreich, Goldwasser, and Ron [26].

We initiate the study of Property Testing when the input is a *non-signaling strategy* [35, 43, 40, 41], which means that the input belongs to a certain class of probabilistic oracles that answer a tester’s queries by sampling from a distribution that may depend on all queries. This setting stands in stark contrast to the standard one, where each query’s answer is *fixed* before queries are sampled. We provide a first analysis of linearity testing against non-signaling strategies, establishing general statements and techniques about non-signaling strategies along the way.

Non-signaling strategies have been studied in Physics for over 30 years as a strict generalization of quantum strategies, in order to understand the power and limitations of Nature’s apparent non-locality.<sup>1</sup> Informally, Quantum Mechanics is a very accurate description of Nature but it may also be an incomplete one: it has not been successfully combined with General Relativity to get a quantum theory of gravity. Nevertheless, there is wide agreement that Nature forbids instantaneous communication despite its apparent non-locality, so this *non-signaling* property must be part of *any* ultimate theory of Nature. Non-signaling strategies exactly capture this minimal requirement, thus (purportedly) capturing any physically-realizable strategy.

Non-signaling strategies also have strong connections to Complexity Theory and Cryptography. Property Testing against non-signaling strategies is likely to strengthen these connections (see Section 4 for details), and thus we believe that it should be explicitly studied.

### 1.1 Linearity testing

A boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  is *linear* if  $f(x) + f(y) = f(x + y)$  for all  $x, y \in \{0, 1\}^n$ , where bits are added modulo two and vectors are added component-wise. The problem of *linearity testing* is to decide whether a given arbitrary boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  is linear or is far from all linear functions. Blum, Luby, and Rubinfeld [15] suggest a very simple 3-query tester: sample uniform and independent  $x, y \in \{0, 1\}^n$ , and check that  $f(x) + f(y) = f(x + y)$ . Perhaps surprisingly, analyzing this tester is far from simple, and a tight characterization of its acceptance probability is still an open problem. Nevertheless, upper and lower bounds on the acceptance probability are known, which is sufficient for applications. Bellare, Coppersmith, Håstad, Kiwi, and Sudan [12] have shown that the acceptance probability is at most  $1 - \Delta(f)$ , where  $\Delta(f)$  is the fractional Hamming distance

---

<sup>1</sup> “Non-locality” refers to correlations in Nature that appear non-local when interpreted using classical physics.

of  $f$  to the closest linear function. Many other works have studied this problem and closely related ones [50, 13, 14, 21]. Finally, Ito and Vidick [30, 52] analyzed linearity testing against quantum strategies. Fixed functions and quantum strategies are both special cases of non-signaling strategies, the subject of this work.

## 1.2 Non-signaling strategies

A non-signaling strategy is a collection of distributions, one per set of queries, that jointly satisfy certain restrictions. There are two distinct definitions, corresponding to whether the strategy is meant to represent a function or players in a game. Throughout most of this paper, we consider *non-signaling functions*, because the functional view fits better the setting of Property Testing; nevertheless, we also consider *non-signaling players*, and show that our results about non-signaling functions imply corresponding results about non-signaling players (see full version for details).

A  $k$ -non-signaling function  $\mathcal{F}$  extends the notion of a function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  as follows: it is a collection  $\{\mathcal{F}_S\}_{S \subseteq \{0, 1\}^n, |S| \leq k}$  where each  $\mathcal{F}_S$  is a *distribution* over functions  $f_S: S \rightarrow \{0, 1\}$  and, for every two subsets  $S$  and  $T$  each of size at most  $k$ , the restrictions of  $\mathcal{F}_S$  and  $\mathcal{F}_T$  to  $S \cap T$  are equal as distributions. We sometimes write “ $\mathcal{F}(S) = \vec{b}$ ”, for a subset  $S \subseteq \{0, 1\}^n$  and string  $\vec{b} \in \{0, 1\}^S$ , to denote the event that the function sampled from  $\mathcal{F}_S$  equals  $\vec{b}$ .

Observe that, given any  $k \in \{1, \dots, 2^n\}$ , every function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  naturally induces a  $k$ -non-signaling function  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq \{0, 1\}^n, |S| \leq k}$ , namely the one where each  $\mathcal{F}_S$  equals the constant distribution that outputs the restriction of  $f$  to  $S$  with probability 1. More generally, every distribution over functions induces a corresponding  $k$ -non-signaling function in a similar way.

However, the set of non-signaling functions is richer, because consistency between local distributions need *not* imply a global distribution, as the following example shows. For  $n = 2$  and  $k = 2$ , consider the non-signaling function  $\{\mathcal{F}_S\}_{S \subseteq \{0, 1\}^2, |S| \leq 2}$  defined as follows:  $\mathcal{F}_{\{00, 11\}}$  is uniform over the two functions  $\left\{ \begin{array}{l} 00 \rightarrow 0 \\ 11 \rightarrow 1 \end{array} , \begin{array}{l} 00 \rightarrow 1 \\ 11 \rightarrow 0 \end{array} \right\}$  and, for every  $\{x, y\} \neq \{00, 11\}$ ,  $\mathcal{F}_{\{x, y\}}$  is uniform over  $\left\{ \begin{array}{l} x \rightarrow 0 \\ y \rightarrow 0 \end{array} , \begin{array}{l} x \rightarrow 1 \\ y \rightarrow 1 \end{array} \right\}$ . No distribution over functions can explain the above strategy, as any  $f$  in the support of such a distribution would have to satisfy  $f(00) \neq f(11)$  and  $f(x) = f(y)$  for every  $\{x, y\} \subseteq \{0, 1\}^2 \setminus \{00, 11\}$ , which is impossible.

## 1.3 The problem and challenges

We study linearity testing against non-signaling functions, which is the following problem.

► **Question 1.1** (informal). *Let  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq \{0, 1\}^n, |S| \leq k}$  be a  $k$ -non-signaling function. Suppose that with probability at least  $1 - \varepsilon$  (for sufficiently small  $\varepsilon \geq 0$ ) it holds that  $f(x) + f(y) = f(x + y)$ , where  $x$  and  $y$  are sampled uniformly and independently from  $\{0, 1\}^n$  and  $f: \{x, y, x + y\} \rightarrow \{0, 1\}$  is sampled from the distribution  $\mathcal{F}_{\{x, y, x + y\}}$ . Can we deduce any global properties about  $\mathcal{F}$ ?*

In order to build intuition about this question, we temporarily put aside the case when  $\varepsilon > 0$ , and focus on the case  $\varepsilon = 0$ , which already turns out to be quite subtle. In other words, let us assume for now that for every  $x, y \in \{0, 1\}^n$  and every  $f$  in the support of  $\mathcal{F}_{\{x, y, x + y\}}$  it holds that  $f(x) + f(y) = f(x + y)$ . What global properties, if any, can we deduce about  $\mathcal{F}$ ?

Ideally, we would like to characterize the set of *all* non-signaling functions that pass the linearity test with probability 1 and say that this set is related to linear functions. If  $\mathcal{F}$

## 17:4 Testing Linearity against Non-Signaling Strategies

is restricted to answer according to a single fixed function  $f: \{0,1\}^n \rightarrow \{0,1\}$  (as in the standard setting) then  $f$  passing the linearity test with probability 1 is *equivalent* to  $f$  being linear by definition. On the other extreme, if  $\mathcal{F}$  is allowed to answer queries arbitrarily without any non-signaling property then no interesting conclusion is possible. The case of  $\mathcal{F}$  being a non-signaling function sits somewhere in between these two extremes:  $\mathcal{F}$  is neither a fixed function nor completely arbitrary. We present two examples to highlight the challenges that arise when seeking an answer.

► **Example 1.2.** Consider the following 3-non-signaling function  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq \{0,1\}^n, |S| \leq 3}$ . For every subset  $\{x, y, x+y\} \subseteq \{0,1\}^n \setminus \{0^n\}$ , the random variable  $f \leftarrow \mathcal{F}_{\{x,y,x+y\}}$  is such that  $(f(x), f(y), f(x+y))$  is uniform over  $\{(0,0,0), (1,1,0), (1,0,1), (0,1,1)\}$ ; for every subset  $\{x, y, z\} \subseteq \{0,1\}^n \setminus \{0^n\}$  with  $z \neq x+y$ , the random variable  $f \leftarrow \mathcal{F}_{\{x,y,z\}}$  is such that  $(f(x), f(y), f(z))$  is uniform over  $\{0,1\}^3$ . For every set  $S \subseteq \{0,1\}^n$  containing  $0^n$ ,  $\mathcal{F}$  samples  $f \leftarrow \mathcal{F}_{S \setminus \{0^n\}}$ , and outputs the function  $g$  where  $g(x) = f(x)$  for  $x \in S \setminus \{0^n\}$  and  $g(0^n) = 0$ . Note that  $\mathcal{F}$  is 3-non-signaling because for every  $S \subseteq \{0,1\}^n \setminus \{0^n\}$  with  $|S| = 3$  the restriction of  $\mathcal{F}_S$  to any two coordinates  $\{x, y\} \subseteq S$  induces a uniformly boolean random function over  $f: \{x, y\} \rightarrow \{0,1\}$ . In particular, for distinct  $x, y \in \{0,1\}^n \setminus \{0^n\}$  it holds that  $\mathcal{F}_{\{0^n, x, y\}}$  outputs 0 on  $0^n$ , and random bits on  $x$  and  $y$ .

Clearly,  $\mathcal{F}$  passes the linearity test with probability 1. Observe that we can alternatively describe its answers according to the following procedure: upon receiving a subset  $S \subseteq \{0,1\}^n$ ,  $\mathcal{F}$  samples a uniformly random *linear* function  $f: \{0,1\}^n \rightarrow \{0,1\}$  (independent of  $S$ ) and returns the restriction of  $f$  to  $S$ . We can thus explain  $\mathcal{F}$  via the uniform distribution over linear functions.

Generalizing from the above example, any non-signaling function that is induced by sampling a linear function from *any* distribution (not just the uniform one) and answering accordingly will pass the linearity test with probability 1. Note that a distribution over linear functions is given by non-negative real numbers  $(p_\alpha)_{\alpha \in \{0,1\}^n}$  such that  $\sum_{\alpha \in \{0,1\}^n} p_\alpha = 1$ , where  $p_\alpha$  is the probability of sampling the function  $\langle \alpha, \cdot \rangle$ . If  $\mathcal{F}$  answers according to  $(p_\alpha)_{\alpha \in \{0,1\}^n}$ , then  $\Pr[\mathcal{F}(x) = b] = \sum_{\alpha: \langle \alpha, x \rangle = b} p_\alpha$  for every  $x \in \{0,1\}^n$  and  $b \in \{0,1\}$ ; a similar formula holds for more inputs.

The above discussion suggests a natural conjecture: every non-signaling function that passes the linearity test with probability 1 can be explained by some distribution over linear functions. In fact, this conjecture *is* true if the non-signaling strategy is restricted to be a quantum strategy [30, 52]. But the set of non-signaling strategies is strictly larger. Below we show that, perhaps surprisingly, these additional strategies make this conjecture false.

► **Example 1.3.** Consider the following 3-non-signaling function  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq \{0,1\}^n, |S| \leq 3}$ . For every subset  $\{x, y, x+y\} \subseteq \{0,1\}^n \setminus \{0^n\}$ ,  $\mathcal{F}_{\{x,y,x+y\}}$  is the following distribution

$$\Pr_{f \leftarrow \mathcal{F}_{\{x,y,x+y\}}} [f(x, y, x+y) = (a_1, a_2, a_3)] = \begin{cases} 1/7 & \text{if } (a_1, a_2, a_3) = (0, 0, 0) \\ 2/7 & \text{if } (a_1, a_2, a_3) \in \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\} \end{cases}$$

for every subset  $\{x, y, z\} \subseteq \{0,1\}^n \setminus \{0^n\}$  with  $z \neq x+y$ ,  $\mathcal{F}_{\{x,y,z\}}$  is the following distribution

$$\Pr_{f \leftarrow \mathcal{F}_{\{x,y,z\}}} [f(x, y, z) = (a_1, a_2, a_3)] = \begin{cases} 0 & \text{if } (a_1, a_2, a_3) = (0, 0, 0) \\ 1/7 & \text{if } (a_1, a_2, a_3) \neq (0, 0, 0) \end{cases}.$$

If an input set  $S$  contains  $0^n$ ,  $\mathcal{F}_S$  assigns  $0^n$  to 0 and answers the rest according to  $\mathcal{F}_{S \setminus \{0^n\}}$ . Note that  $\mathcal{F}$  is 3-non-signaling because for distinct and non-zero  $x$  and  $y$ , the distribution of



$\mathcal{F}_{\{x,y\}}$  is

$$\Pr_{f \leftarrow \mathcal{F}_{\{x,y\}}} [f(x,y) = (a_1, a_2)] = \begin{cases} 1/7 & \text{if } (a_1, a_2) = (0, 0) \\ 2/7 & \text{if } (a_1, a_2) \neq (0, 0) \end{cases} .$$

In particular, for distinct and non-zero  $x$  and  $y$ , the distribution of  $\mathcal{F}_{\{x,y,0^n\}}$  is

$$\Pr_{f \leftarrow \mathcal{F}_{\{x,y,0^n\}}} [f(x,y,0^n) = (a_1, a_2, 0)] = \begin{cases} 1/7 & \text{if } (a_1, a_2) = (0, 0) \\ 2/7 & \text{if } (a_1, a_2) \in \{(1, 0), (0, 1), (1, 1)\} \end{cases} .$$

Observe that  $\mathcal{F}$  passes the linearity test with probability 1. However, unlike before, a distribution over linear functions that explains  $\mathcal{F}$  *does not exist*. Namely, there is no probability vector  $(p_\alpha)_{\alpha \in \{0,1\}^n}$  with non-negative entries and  $\sum_{\alpha \in \{0,1\}^n} p_\alpha = 1$  such that  $\Pr[\mathcal{F}(x) = b] = \sum_{\alpha: \langle \alpha, x \rangle = b} p_\alpha$  for all  $x \in \{0,1\}^n$  and  $b \in \{0,1\}$ . In fact, when trying to solve this linear system of equations with  $(p_\alpha)_{\alpha \in \{0,1\}^n}$  as the variables, we obtain a solution vector in which some of the entries are *negative*.

The above example is problematic because it seems to suggest that a clean characterization of the set of all non-signaling functions passing the linearity test does not exist. Indeed, it shows that this set is strictly richer than the set of all distributions over linear functions.

### 1.4 Negative probabilities and quasi-distributions

In order to resolve the difficulty encountered in Example 1.3, we *embrace* negative probabilities (and probabilities greater than 1), and consider the notion of a *quasi-distribution* over boolean functions.

► **Definition 1.4** (informal). A *quasi-distribution* is defined as a vector of real numbers  $\mathcal{Q} = \{q_f\}_{f: \{0,1\}^n \rightarrow \{0,1\}}$  such that  $\sum_{f: \{0,1\}^n \rightarrow \{0,1\}} q_f = 1$ . Similarly, a *quasi-distribution over linear functions* is a quasi-distribution  $\mathcal{Q} = \{q_f\}_{f: \{0,1\}^n \rightarrow \{0,1\}}$  such that  $q_f = 0$  for all  $f$  that are not linear functions; in this case, we also allow ourselves to represent the quasi-distribution by a vector  $(q_\alpha)_{\alpha \in \{0,1\}^n}$ , where each  $q_\alpha$  is associated with the linear function  $\langle \alpha, \cdot \rangle$ .

A function  $f$  in a quasi-distribution  $\mathcal{Q} = \{q_f\}_f$  is thus “sampled” with “probability”  $q_f$ , which means that for every subset  $S \subseteq \{0,1\}^n$  and string  $\vec{b} \in \{0,1\}^S$  the event “ $\mathcal{Q}(S) = \vec{b}$ ” has *quasi-probability* given by  $\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] := \sum_{f \text{ s.t. } f(S) = \vec{b}} q_f$ .

This may seem nonsensical, because quasi-probabilities are not restricted to be in  $[0, 1]$ . But this shall soon make sense. In the words of Paul Dirac [22, p.8]: “*Negative energies and probabilities should not be considered as nonsense. They are well-defined concepts mathematically, like a negative sum of money, since the equations which express the important properties of energies and probabilities can still be used when they are negative. Thus negative energies and probabilities should be considered simply as things which do not appear in experimental results.*”

This viewpoint, which plays a central role in our work, is borrowed from Physics, where it is used to describe many physical phenomena [22, 25], including non-signaling ones [2].

While the non-signaling function  $\mathcal{F}$  in Example 1.3 cannot be explained by any distribution over linear functions, it *can* be explained by a *quasi-distribution* over linear functions. Concretely, letting  $q_\alpha$  represent the probability of “sampling” the function  $\langle \alpha, \cdot \rangle$ , we solve the following system of linear equations in the variables  $(q_\alpha)_{\alpha \in \{0,1\}^n}$ :

$$\sum_{\alpha \in \{0,1\}^n} q_\alpha = 1 \quad \text{and} \quad \forall x \in \{0,1\}^n \quad \forall b \in \{0,1\} \quad \sum_{\alpha: \langle \alpha, x \rangle = b} q_\alpha = \Pr[\mathcal{F}(x) = b] .$$

The solution to this system is  $q_{\vec{0}} = 1 - \frac{8}{7} \frac{2^n - 1}{2^n} < 0$  and  $q_\alpha = \frac{8}{7} \cdot \frac{1}{2^n}$  for all  $\alpha \neq \vec{0}$ . We stress that the solution has a negative entry. One can then verify that the quasi-distribution obtained above not only matches  $\mathcal{F}$  on events involving one input (which is by construction) but also on events involving two inputs:  $\Pr[\mathcal{F}(x_1) = b_1, \mathcal{F}(x_2) = b_2] = \sum_{\alpha: \langle \alpha, x_1 \rangle = b_1, \langle \alpha, x_2 \rangle = b_2} q_\alpha$  for all  $x_1, x_2 \in \{0, 1\}^n$  and  $b_1, b_2 \in \{0, 1\}$ . Similarly, the same holds for events involving three inputs.

Crucially, the quasi-probabilities of events that involve a small enough set of inputs “magically” add up to *non-negative* probabilities because, in particular, they describe distributions of  $\mathcal{F}$ . In other words, like in Dirac’s observation above, the negative probabilities “do not appear in experimental results”; in our case the experiment is querying  $\mathcal{F}$ , and a quasi-distribution is merely a convenient mathematical abstraction to describe it.

The foregoing considerations directly lead to the following observation.

► **Observation 1.5.** *If  $\mathcal{Q} = (q_f)_{f: \{0,1\}^n \rightarrow \{0,1\}}$  is a quasi-distribution that induces a probability distribution on every event of at most  $k$  inputs, then  $\mathcal{Q}$  induces a  $k$ -non-signaling function.*

*Furthermore, if  $\mathcal{Q}$  is supported on linear functions only, then the corresponding  $k$ -non-signaling function passes the linearity test with probability 1.*

The first part of the observation suggests using  $k$  as a measure of a quasi-distribution’s locality: we say that a quasi-distribution  $\mathcal{Q} = (q_f)_f$  is  *$k$ -local* if for every  $k$  inputs  $x_1, \dots, x_k \in \{0, 1\}^n$  and  $k$  outputs  $b_1, \dots, b_k \in \{0, 1\}$  it holds that  $\sum_{f: f(x_1)=b_1, \dots, f(x_k)=b_k} q_f \geq 0$ . Thus  $\mathcal{Q}$  behaves like a collection of (standard) distributions on all events that involve at most  $k$  inputs and, moreover, these distributions jointly satisfy the  $k$ -non-signaling property.

The second part of the observation shows the existence of a class of non-signaling functions that pass the linearity test with probability 1 that is *much richer* than the class of distribution over linear functions. Are there any other types of non-signaling functions that pass the linearity test with probability 1, or are these all of them? Moreover, how does this answer change when we merely require that a non-signaling function pass the linearity test with probability at least  $1 - \varepsilon$ ? We now discuss our results, which will provide answers to these questions.

## 2 Our results

Quasi-distributions arose rather naturally when reasoning about non-signaling functions. First, we show that this is not a coincidence by proving that the two notions are equivalent.

► **Theorem 2.1 (informal).** *Local quasi-distributions and non-signaling functions are equivalent:*

1. *every  $k$ -local quasi-distribution induces a corresponding  $k$ -non-signaling function; conversely,*
2. *every  $k$ -non-signaling function has a  $k$ -local quasi-distribution that describes it. (In fact, this quasi-distribution is not unique: the set of all such quasi-distributions is an affine subspace.)*

See Section 8 (specifically, Theorem 8.1 and Theorem 8.2) for precise statements of the two items.

The first item is just Observation 1.5. The second item is proved via Fourier analytic techniques applied to a quasi-probability vector. Informally, the Fourier coefficients of quasi-probability vectors are indexed by subsets of  $\{0, 1\}^n$ , and can be grouped into *levels* according to their size. We prove that the only coefficients that matter for the  $k$ -non-signaling

function are those in the levels for sizes at most  $k$ , while all others change the weights in the quasi-probability vector but do not affect the induced  $k$ -non-signaling function.

The foregoing equivalence can be viewed as the “functional analogue” of an equivalence proved in [2] for the (incomparable) case of non-signaling players. The Fourier analytic techniques that we use are novel and, moreover, can be adapted to the case of non-signaling players in order to strengthen [2]’s result to find *all* quasi-distributions (rather than just one) that describe a given set of non-signaling players (see full version for details). We believe that the mathematical structure uncovered by our Fourier analytic techniques is of independent interest.

Having established the equivalence of local quasi-distributions and non-signaling functions, we return to the problem of linearity testing against non-signaling functions. Our first theorem in this direction is a characterization of the set of non-signaling functions that pass the linearity test with probability 1: this set consists of local quasi-distributions over linear functions (essentially).

► **Theorem 2.2** (informal). *Let  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq \{0,1\}^n, |S| \leq k}$  be a  $k$ -non-signaling function such that*

$$\Pr_{\substack{x, y \leftarrow \{0,1\}^n \\ f \leftarrow \mathcal{F}_{\{x, y, x+y\}}}} [f(x) + f(y) = f(x + y)] = 1 .$$

*There is a unique  $(k - 1)$ -local quasi-distribution  $\mathcal{L}$  over linear functions describing  $\mathcal{F}$  on all input sets of size  $\leq k - 1$  ( $\mathcal{L}_S$  and  $\mathcal{F}_S$  are equal as distributions for every set  $S \subseteq \{0, 1\}^n$  with  $|S| \leq k - 1$ ).*

See Theorem 10.1 in Section 10 for the precise statement. (A minor technicality of the theorem is that  $\mathcal{L}$  is only  $(k - 1)$ -local and only matches  $\mathcal{F}$  on at most  $k - 1$  inputs; the discussion after Theorem 10.1 explains why this is the best we can hope for.) To prove the theorem we define a quasi-distribution  $\mathcal{L}$  over linear functions by solving a certain system of linear equations that ensures that  $\mathcal{L}$  and  $\mathcal{F}$  match on single inputs, i.e., that  $\widetilde{\Pr}[\mathcal{L}(x) = b] = \Pr[\mathcal{F}(x) = b]$  for all  $x \in \{0, 1\}^n$  and  $b \in \{0, 1\}$ . We then need to establish that  $\mathcal{L}$  and  $\mathcal{F}$  match on all sets of at most  $k - 1$  inputs. We do so in two steps: we first use linearity to show that  $\mathcal{L}$  and  $\mathcal{F}$  match on all parity events (i.e.,  $\widetilde{\Pr}[\sum_{i \in T} \mathcal{L}(x_i) = b] = \Pr[\sum_{i \in T} \mathcal{F}(x_i) = b]$  for all  $x_1, \dots, x_s \in \{0, 1\}^n$  and  $b \in \{0, 1\}$  with  $s \leq k - 1$ ); then we use Fourier analysis to extend this claim to all allowed input sets.

We finally return to our original question (Question 1.1). Suppose that a non-signaling function  $\mathcal{F}$  passes the linearity test with probability  $1 - \varepsilon$  for sufficiently small  $\varepsilon \geq 0$  (possibly with  $\varepsilon > 0$  so Theorem 2.2 does not apply). What can we learn about  $\mathcal{F}$ ? Recall that if  $\mathcal{F}$  answers according to a fixed function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  (as in standard linearity testing), then we may conclude that  $f$  is  $\varepsilon$ -close to some linear function [15, 12]. The foregoing discussion for the case of  $\varepsilon = 0$  leads to a natural conjecture: *non-signaling functions that pass the linearity test with high probability are local quasi-distributions over functions that are close to linear*. Our next theorem implies that this conjecture is true, but in a non-interesting way. That is, it holds even without the hypothesis: *every  $k$ -non-signaling function can be expressed as a quasi-distribution over functions with support of size at most  $k$  (namely, over functions that are non-zero for at most  $k$  inputs)*.

► **Theorem 2.3** (informal). *Every  $k$ -non-signaling function  $\mathcal{F}$  can be expressed as a  $k$ -local quasi-distribution  $\mathcal{Q}$  over functions with support of size at most  $k$ .*

## 17:8 Testing Linearity against Non-Signaling Strategies

The above theorem is quite counterintuitive. On one hand, if  $\mathcal{F}$  is described by a distribution over functions that are close to linear, then  $\mathcal{F}$  passes the linearity test with high probability. But this simple fact does *not* extend to the case where  $\mathcal{F}$  is a *quasi*-distribution over functions that are close to linear. For example, the all-ones function never passes the linearity test, yet Theorem 2.3 implies that it can be expressed as a quasi-distribution over functions with support of size at most  $k$ , i.e., functions that are  $\frac{k}{2^n}$ -close to the all-zeros function (a linear function)!

We prove Theorem 2.3 via a greedy approach: given the non-signaling function  $\mathcal{F}$ , we iteratively consider small-support functions from heaviest to lightest and, in each iteration, assign to these functions certain quasi-probabilities computed from  $\mathcal{F}$ . See Theorem 9.1 (in Section 9) for details.

Since our last conjecture turned out to be false, we again look for inspiration in the standard setting in order to formulate another conjecture. Taking a different view, linearity testing tells us that if a function  $f: \{0,1\}^n \rightarrow \{0,1\}$  passes the linearity test with high probability then we know that there exists a linear function  $L$  such that for *every*  $x \in \{0,1\}^n$  it holds that  $L(x) = f(x+y) - f(y)$  with high probability over a random  $y \in \{0,1\}^n$ . Put another way, the answers to any given query (or, more generally, a set of queries) given by the self-correction of  $f$  and by  $L$  are close in statistical distance.

The foregoing observation suggests a conjecture: *if a non-signaling function passes the linearity test with high probability, then its self-correction is close to a quasi-distribution over linear functions.*

The self-correction  $\hat{\mathcal{F}}$  of a non-signaling function  $\mathcal{F}$  is naturally defined: on input  $x \in \{0,1\}^n$ ,  $\hat{\mathcal{F}}$  samples a random  $y \in \{0,1\}^n$  and outputs  $\mathcal{F}(x+y) - \mathcal{F}(y)$ ; a similar procedure applies if  $\hat{\mathcal{F}}$  receives multiple inputs. Note that if  $\mathcal{F}$  is  $k$ -non-signaling then  $\hat{\mathcal{F}}$  is  $\hat{k}$ -non-signaling with  $\hat{k} := \lfloor k/2 \rfloor$ .

The notion of distance is also naturally defined: the distance between two non-signaling functions is the maximum statistical distance between the distributions induced on every subset  $S$ ; the equivalence of non-signaling functions and quasi-distributions (Theorem 2.1) extends this definition to apply between two quasi-distributions, or between a non-signaling function and a quasi-distribution.

The following theorem shows that the conjecture above is in fact true.

► **Theorem 2.4** (informal). *Let  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq \{0,1\}^n, |S| \leq k}$  be a  $k$ -non-signaling function such that*

$$\Pr_{\substack{x,y \leftarrow \{0,1\}^n \\ f \leftarrow \mathcal{F}_{\{x,y,x+y\}}} [f(x) + f(y) = f(x+y)] \geq 1 - \varepsilon \quad \text{for some } \varepsilon \geq 0 .$$

*There is a  $(\hat{k} - 1)$ -local quasi-distribution  $\mathcal{L}$  over linear functions that is  $O_{\hat{k}}(\varepsilon)$ -close to  $\hat{\mathcal{F}}$  on all input sets of size  $\leq \hat{k} - 1$ . That is, the maximum statistical distance between  $\mathcal{L}_S$  and  $\hat{\mathcal{F}}_S$ , across all sets  $S \subseteq \{0,1\}^n$  with  $|S| \leq \hat{k} - 1$ , is  $O_{\hat{k}}(\varepsilon)$ .*

See Theorem 11.2 (in Section 11) for details. Our proof differs significantly from prior proofs of linearity testing in the standard setting. Informally, we start the proof by noting that  $\hat{\mathcal{F}}$  satisfies  $\Pr_{f \leftarrow \hat{\mathcal{F}}_{\{x,y,x+y\}}} \Pr[\hat{f}(x) + \hat{f}(y) = \hat{f}(x+y)] \geq 1 - \hat{\varepsilon}$  for *every*  $x, y \in \{0,1\}^n$  and  $\hat{\varepsilon} := 4\varepsilon$ . (By assumption,  $\mathcal{F}$  merely satisfies such a statement for *random*  $x, y \in \{0,1\}^n$ .) The next step is similar to a step in the proof of Theorem 2.2: we define a quasi-distribution  $\mathcal{L}$  over linear functions by solving a system of linear equations that ensures that  $\mathcal{L}$  and  $\hat{\mathcal{F}}$  match on single inputs, i.e., that  $\Pr[\mathcal{L}(x) = b] = \Pr[\hat{\mathcal{F}}(x) = b]$  for all  $x \in \{0,1\}^n$  and  $b \in \{0,1\}$ .

We are left to argue that  $\mathcal{L}$  and  $\hat{\mathcal{F}}$  *almost* match on all sets of at most  $\hat{k} - 1$  inputs, i.e., that the distributions  $\mathcal{L}_S$  and  $\hat{\mathcal{F}}_S$  are statistically close for  $|S| < \hat{k}$ . As before, we do so in two steps: we first use linearity to show that  $\mathcal{L}$  and  $\hat{\mathcal{F}}$  *almost* match on all parity events (i.e.,  $\Pr[\sum_{i \in T} \hat{\mathcal{F}}(x_i) = b] \approx \widehat{\Pr}[\sum_{i \in T} \mathcal{L}(x_i) = b]$  for all  $x_1, \dots, x_s \in \{0, 1\}$  for  $s \leq \hat{k} - 1$ ), and then we use a quantitative Fourier analytic claim (Lemma 5.1) to extend this claim to the remaining query sets.

Finally, we use the foregoing results about non-signaling *functions* to prove analogous statements about non-signaling *players*.

Recall that a  $k$ -non-signaling player  $\mathcal{P}$  extends the notion of  $k$  non-communicating players (possibly sharing randomness) as follows: it is a collection  $(\mathcal{P}_{(x_1, \dots, x_k)})_{(x_1, \dots, x_k) \in \{0, 1\}^{k \cdot n}}$  where each  $\mathcal{P}_{(x_1, \dots, x_k)}$  is a *distribution* over functions  $f: [k] \rightarrow \{0, 1\}$  (the players'  $k$  answers to the  $k$  inputs) and, for every two input vectors  $(x_1, \dots, x_k)$  and  $(y_1, \dots, y_k)$  that agree on a subset  $I \subseteq [k]$  of entries, the restrictions of  $\mathcal{P}_{(x_1, \dots, x_k)}$  and  $\mathcal{P}_{(y_1, \dots, y_k)}$  to entries in  $I$  are equal as distributions. Non-signaling players are a richer class than non-communicating players (and quantum-entangled ones) [40].

Now the linearity test, given a  $k$ -non-signaling player  $\mathcal{P} = (\mathcal{P}_{(x_1, \dots, x_k)})_{(x_1, \dots, x_k) \in \{0, 1\}^{k \cdot n}}$ , samples random vectors  $x, y \in \{0, 1\}^n$  and distinct players  $i_1, i_2, i_3 \in [k]$ , sends the three queries  $x, y, x + y$  to the players  $\mathcal{P}_{i_1}, \mathcal{P}_{i_2}, \mathcal{P}_{i_3}$ , and checks that  $\mathcal{P}_{i_1}(x) + \mathcal{P}_{i_2}(y) = \mathcal{P}_{i_3}(x + y)$ .

► **Theorem 2.5** (informal). *Let  $\mathcal{P} = (\mathcal{P}_{(x_1, \dots, x_k)})_{(x_1, \dots, x_k) \in \{0, 1\}^{k \cdot n}}$  be a  $k$ -non-signaling player.*

1. *Suppose that*

$$\Pr_{\substack{x, y \leftarrow \{0, 1\}^n \\ i_1, i_2, i_3 \leftarrow [k] \\ \mathcal{P}}} [\mathcal{P}_{i_1}(x) + \mathcal{P}_{i_2}(y) = \mathcal{P}_{i_3}(x + y)] = 1 .$$

*There exists a  $(k - 2)$ -local quasi-distribution  $\mathcal{L}$  over linear functions that describes  $\mathcal{P}$ .*

2. *Suppose that*

$$\Pr_{\substack{x, y \leftarrow \{0, 1\}^n \\ i_1, i_2, i_3 \leftarrow [k] \\ \mathcal{P}}} [\mathcal{P}_{i_1}(x) + \mathcal{P}_{i_2}(y) = \mathcal{P}_{i_3}(x + y)] \geq 1 - \varepsilon .$$

*There exists a  $(\hat{k} - 1)$ -local quasi-distribution  $\mathcal{L}$  over linear functions that is  $O_{\hat{k}}(\varepsilon)$ -close to  $\hat{\mathcal{P}}$ , where  $\hat{\mathcal{P}}$  is the (appropriately defined) self-correction of  $\mathcal{P}$ .*

See full version for details. The proof of these theorems show how to reduce to the case of non-signaling functions, which we have already established (in Theorems 2.2 and 2.4 respectively).

We conclude this section via a brief comparison to the case of quantum strategies. Ito and Vidick [30, 52] show that any quantum strategy that passes the linearity test with high probability is close to a *distribution* over linear functions. Our results instead show that, in our setting, we can only hope for a conclusion involving a *quasi-distribution* over linear functions. This qualitative difference is due to the fact that non-signaling strategies are a richer class than quantum strategies.

### 3 Techniques

We highlight some of the techniques that we use by providing proof sketches of some of our results. We first discuss the ideas behind the equivalence between non-signaling functions and local quasi-distributions (Section 3.1) and then how we analyze the linearity test (Section 3.2). After that, we explain how we derive corresponding results about non-signaling players (Section 3.3).

### 3.1 Non-signaling functions and local quasi-distributions are equivalent

Our Theorem 2.1 states that non-signaling functions and local quasi-distributions are equivalent. One direction of this equivalence, namely that every  $k$ -local quasi-distribution induces a corresponding  $k$ -non-signaling function, is a simple observation. Below we focus on the other, more interesting direction, which is: given a  $k$ -non-signaling function  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq \{0,1\}^n, |S| \leq k}$ , how do we construct a quasi-distribution  $\mathcal{Q} = \{q_f\}_{f: \{0,1\}^n \rightarrow \{0,1\}}$  that matches  $\mathcal{F}$  on all sets of at most  $k$  queries?

We construct  $\mathcal{Q}$  by specifying its *Fourier coefficients*. We view  $\mathcal{Q} = (q_f)_{f: \{0,1\}^n \rightarrow \{0,1\}}$  as a function  $q: \{0,1\}^{\{0,1\}^n} \rightarrow \mathbb{R}$  by setting  $q(f) := q_f \in \mathbb{R}$ , and then write  $\mathcal{Q}$  via its Fourier expansion:

$$q(\cdot) = \sum_{T \subseteq \{0,1\}^n} \hat{q}(T) \chi_T(\cdot) \quad \text{where} \quad \begin{cases} \chi_T(f) := (-1)^{\sum_{x \in T} f(x)} \\ \hat{q}(T) := \langle q, \chi_T \rangle = \frac{1}{2^{2^n}} \sum_{f: D \rightarrow \{0,1\}} q(f) \chi_T(f) \end{cases} .$$

We set the  $2^{2^n}$  Fourier coefficients as follows:

$$\hat{q}(T) := \begin{cases} \frac{1}{2^{2^n}} & \text{if } T = \emptyset \\ \frac{2}{2^{2^n}} (\Pr[\sum_{x \in T} \mathcal{F}(x) = 0] - \frac{1}{2}) & \text{if } 1 \leq |T| \leq k \\ 0 & \text{if } |T| > k \end{cases} .$$

We have to argue that the above choice of  $\mathcal{Q}$  does describe  $\mathcal{F}$ . First, we show that  $\mathcal{F}$  and  $\mathcal{Q}$  match on all *parity events* of size at most  $k$ , i.e., for all  $S \subseteq \{0,1\}^n$  with  $|S| \leq k$

$$\Pr \left[ \sum_{x \in S} \mathcal{F}(x) = 1 \right] = \sum_{f: \sum_{x \in S} f(x) = 1} q_f = \widetilde{\Pr} \left[ \sum_{x \in S} \mathcal{Q}(x) = 1 \right] .$$

Recall (see Section 1.4) that  $\widetilde{\Pr}[\cdot]$  denotes the quasi-probability for an event about a quasi-distribution.

Second, we prove that  $\Pr[\mathcal{F}(S) \in E] = \widetilde{\Pr}[\mathcal{Q}(S) \in E]$  for every subset  $S \subseteq \{0,1\}^n$  and event  $E \subseteq \{0,1\}^S$ . We build on the previous step by observing that any event can be expressed as a linear combination of parity events: there exist real numbers  $\{c_T\}_T$  depending on  $E$  such that

$$\Pr[\mathcal{Q}(S) \in E] = \sum_{T \subseteq S} c_T \cdot \widetilde{\Pr} \left[ \sum_{x \in T} \mathcal{Q}(x) = 0 \right] . \quad (1)$$

In fact, the real numbers  $\{c_T\}_T$  are closely related to the Fourier coefficients of the indicator function of  $E$ , and this relation is a consequence of the fact that the functions  $\{\chi_T(\cdot)\}_T$  depend only on the parities of their inputs. See Lemma 5.1 for details.

The above is merely one quasi-distribution that explains  $\mathcal{F}$ . We can find other such quasi-distributions by noting that changing  $\hat{q}(T)$  for  $|T| > k$  yields quasi-distributions that still match  $\mathcal{F}$ . Essentially, if  $|T| > k$  then  $\chi_T(\cdot)$  does not affect the induced distributions on sets of at most  $k$  inputs. We then argue that these are the only solutions possible. See Section 8 for details.

### 3.2 Testing linearity against non-signaling functions

We discuss the ideas behind our analysis of linearity test against non-signaling functions (that is, behind Theorem 2.2 and Theorem 2.4). We first explain why known proofs in the standard setting do not easily extend to our setting, and then we describe the approach that we took.

### 3.2.1 Difficulties of prior approaches

We begin with a helpful exercise for which difficulties do *not* arise: consider the task of analyzing the linearity test against a *distribution*  $\mathcal{D}$  over boolean functions. Namely, if  $\Pr[f(x) + f(y) = f(x + y)] \geq 1 - \varepsilon$  for  $f \leftarrow \mathcal{D}$  and  $x, y \leftarrow \{0, 1\}^n$  then what can we conclude about  $\mathcal{D}$ ? This case is not hard to analyze: we separately apply known results on linearity testing to each function in the support of  $\mathcal{D}$ , and conclude that most of  $\mathcal{D}$  is concentrated on nearly-linear functions. Indeed, by Markov's inequality, with probability  $1 - \sqrt{\varepsilon}$  over a choice of  $f \leftarrow \mathcal{D}$  it holds that  $\Pr_{x,y}[f(x) + f(y) = f(x + y)] \geq 1 - \sqrt{\varepsilon}$  and thus that  $f$  is  $\sqrt{\varepsilon}$ -close to a linear function. This conclusion explains why  $\mathcal{D}$  passes the linearity test with high probability.

However, when considering the linearity test against a non-signaling function, the situation changes significantly, as we now explain.

**The Fourier analytic approach.** One of the classical proofs of linearity testing in the standard setting follows a Fourier analytic approach [12]. Unfortunately, we do not see how to use this approach directly on a non-signaling function  $\mathcal{F}$ , because computing Fourier coefficients requires access to an entire function while  $\mathcal{F}$  only provides local views. We could instead rely on the equivalence between non-signaling functions and local quasi-distributions, and apply Fourier analysis to the functions in a quasi-distribution  $\mathcal{Q} = (q_f)_{f: \{0,1\}^n \rightarrow \{0,1\}}$  that describes  $\mathcal{F}$ . Namely, we could rewrite the probability  $\Pr[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)]$  as  $\sum_f q_f \Pr[f(x) + f(y) = f(x + y)]$ , and then reason about the Fourier coefficients of every  $f$ . We do not see how to make this work either, because the coefficients  $\{q_f\}_f$  can be positive or negative (and even unbounded), which in particular forbids Markov-type arguments. It is also not clear what kind of conclusion we could expect about the Fourier coefficients about *all* functions.

**The combinatorial approach.** Another classical proof of linearity testing in the standard setting follows a combinatorial approach (e.g., [15, 13]): given the function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , define its correction  $g: \{0, 1\}^n \rightarrow \{0, 1\}$  to be  $g(x) := \text{maj}_{y \in \{0,1\}^n} f(x + y) - f(y)$ , and show that it is close to  $f$ ; then show that  $g$  is linear as, for every  $x \in \{0, 1\}^n$ , a vast majority of  $y$ 's yield  $g(x)$ . This approach also seems to fail in our setting: the foregoing correcting procedure relies on taking majority over *all*  $y \in \{0, 1\}^n$ , but a non-signaling function only accepts up to  $k$  inputs at a time.

It is not surprising that prior approaches do not seem to apply to our setting: they were developed to show that a function  $f$  passing the linearity test with high probability is nearly-linear. But we already know that every non-signaling function can be described by a quasi-distribution over nearly-linear functions, so we are not interested in conclusions about nearly-linear functions. Instead, we aim to show that (the self-correction of) a non-signaling function passing the linearity test with high probability is close to a quasi-distribution over *linear* functions. We next discuss our approach to establish such a conclusion.

### 3.2.2 Our approach

Let us once more first focus on the case where a  $k$ -non-signaling function  $\mathcal{F}$  passes the linearity test with probability 1, namely,  $\Pr[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] = 1$  for every  $x, y \in \{0, 1\}^n$ . Our first step is to show that there exists a quasi-distribution  $\mathcal{L}$  over linear functions that matches  $\mathcal{F}$  on single inputs, namely,  $\widetilde{\Pr}[\mathcal{L}(x) = b] = \Pr[\mathcal{F}(x) = b]$  for every  $x \in \{0, 1\}^n$  and

## 17:12 Testing Linearity against Non-Signaling Strategies

$b \in \{0, 1\}$ . Viewing  $\mathcal{L}$  as a vector  $(\ell_\alpha)_{\alpha \in \{0,1\}^n}$  where each  $\alpha$  is associated with the linear function  $\langle \alpha, \cdot \rangle$ , we know that  $\mathcal{L}$  must be a solution to the following system of linear equations:

$$\forall x \in \{0, 1\}^n, \quad \sum_{\alpha: \langle \alpha, x \rangle = 0} \ell_\alpha = \Pr[\mathcal{F}(x) = 0] .$$

Note that it suffices to consider constraints only involving  $\Pr[\mathcal{F}(x) = 0]$  because  $\Pr[\mathcal{F}(x) = 1] = 1 - \Pr[\mathcal{F}(x) = 0]$ . Also,  $\mathcal{L}$  is a quasi-distribution because  $\sum_\alpha \ell_\alpha = \Pr[\mathcal{F}(0^n) = 0] = \Pr_{x \leftarrow \{0,1\}^n}[\mathcal{F}(0^n) + \mathcal{F}(x) = \mathcal{F}(x)] = 1$  (as  $\mathcal{F}$  always passes the linearity test). This system has a unique solution, which thus defines the quasi-distribution  $\mathcal{L}$ . We remark that it is no coincidence that quasi-distributions supported on LIN are uniquely defined by their probabilities on sets of size 1: a quasi-distribution is supported on LIN if and only if all of its Fourier coefficients are determined by the coefficients only for sets of size 1 (see full version for details).

Next, we need to argue that  $\mathcal{L}$  and  $\mathcal{F}$  match on larger sets of inputs. We first argue that they match on all parity events, similarly to the idea behind the equivalence between non-signaling functions and quasi-distributions discussed above (in Section 3.1). Specifically, we use the assumption on linearity to show that for every subset  $S \subseteq \{0, 1\}^n$  with  $|S| < k$  it holds that

$$\widetilde{\Pr} \left[ \sum_{x \in S} \mathcal{L}(x) = 0 \right] = \Pr \left[ \sum_{x \in S} \mathcal{F}(x) = 0 \right] .$$

After that, using Eq. (1), we conclude that  $\mathcal{L}$  and  $\mathcal{F}$  match on all sets  $S$  of less than  $k$  inputs: we express each event  $E \subseteq \{0, 1\}^S$  as a linear combination of parity events for both  $\mathcal{F}$  and  $\mathcal{L}$ ,

$$\Pr[\mathcal{F}(S) \in E] = \sum_{T \subseteq S} c_T \cdot \Pr \left[ \sum_{x \in T} \mathcal{F}(x) = 0 \right] ,$$

and similarly

$$\widetilde{\Pr}[\mathcal{L}(S) \in E] = \sum_{T \subseteq S} c_T \cdot \widetilde{\Pr} \left[ \sum_{x \in T} \mathcal{L}(x) = 0 \right] .$$

The above shows that matching on parity events implies matching on all sets of less than  $k$  inputs.

Let us now relax the assumption that  $\mathcal{F}$  passes the linearity test with probability 1 to merely that it passes the test with high probability, say at least  $1 - \varepsilon$  for  $\varepsilon > 0$ . We first consider  $\hat{\mathcal{F}}$ , which is the  $\hat{k}$ -non-signaling self-correction of  $\mathcal{F}$  (with  $\hat{k} := k/2$ ), and observe that there exists  $\hat{\varepsilon} = 4\varepsilon$  such that  $\hat{\mathcal{F}}$  satisfies, for *every*  $x, y \in \{0, 1\}^n$ ,

$$\Pr_{\hat{f} \leftarrow \hat{\mathcal{F}}_{\{x, y, x+y\}}} [\hat{f}(x) + \hat{f}(y) = \hat{f}(x+y)] \geq 1 - \hat{\varepsilon} .$$

Note that, by assumption,  $\mathcal{F}$  merely satisfies such a statement for *random*  $x, y \in \{0, 1\}^n$ .

The next step is similar to the “ $\varepsilon = 0$ ” case discussed above: we define a quasi-distribution  $\mathcal{L}$  over linear functions by solving the system of linear equations that ensures that  $\mathcal{L}$  and  $\hat{\mathcal{F}}$  match on all single inputs, i.e., that  $\widetilde{\Pr}[\mathcal{L}(x) = b] = \Pr[\hat{\mathcal{F}}(x) = b]$  for all  $x \in \{0, 1\}^n$  and  $b \in \{0, 1\}$ .

We then argue that  $\mathcal{L}$  and  $\hat{\mathcal{F}}$  *almost* match on sets of less than  $\hat{k}$  inputs, i.e., that the distributions  $\mathcal{L}_S$  and  $\hat{\mathcal{F}}_S$  are statistically close for every  $S \subseteq \{0, 1\}^n$  with  $|S| < \hat{k}$ . We do so,



again, in two steps. First, we use the almost linearity of  $\hat{\mathcal{F}}$  to show that  $\mathcal{L}$  and  $\hat{\mathcal{F}}$  *almost* match on all parity events. Specifically, we show that for every subset  $T \subseteq \{0, 1\}^n$  with  $|T| < \hat{k}$  and  $b \in \{0, 1\}$ ,

$$\left| \Pr \left[ \sum_{x \in T} \hat{\mathcal{F}}(x) = b \right] - \widetilde{\Pr} \left[ \sum_{x \in T} \mathcal{L}(x) = b \right] \right| < (|T| - 1) \hat{\varepsilon} .$$

Then, we use Eq. (1) to extend this claim to all events on these query sets: for every subset  $S \subseteq \{0, 1\}^n$  with  $|S| < \hat{k}$  and event  $E \subseteq \{0, 1\}^S$

$$\left| \Pr[\hat{\mathcal{F}}(S) \in E] - \widetilde{\Pr}[\mathcal{L}(S) \in E] \right| < \sum_{T \subseteq S} |c_T| \cdot (|T| - 1) \cdot \hat{\varepsilon} .$$

Crucially, unlike the case of  $\varepsilon = 0$ , here we need *quantitative* bounds on the coefficients  $\{c_T\}_T$  in order to derive an upper bound. We prove such bounds in Lemma 5.1.

Finally, while  $\mathcal{L}$  is close to  $\hat{\mathcal{F}}$  (see Definition 7.5 for how to extend the notion of statistical distance to our setting), it is possible that  $\mathcal{L}$  does not induce a distribution on all subsets  $S \subseteq \{0, 1\}^n$  with  $|S| < \hat{k}$ , because it could be that  $\widetilde{\Pr}[\mathcal{L}(S) \in E]$  is negative for some  $S$  and  $E \subseteq \{0, 1\}^S$ . However, since  $\Pr[\hat{\mathcal{F}}(S) \in E]$  is a probability (i.e., a number between 0 and 1), for all subsets  $S \subseteq \{0, 1\}^n$  with  $|S| < \hat{k}$  it holds that  $\widetilde{\Pr}[\mathcal{L}(S) \in E] \in [-\varepsilon', 1 + \varepsilon']$  for  $\varepsilon' := (|S| - 1) \cdot \sqrt{|E|} \cdot \hat{\varepsilon}$ . We then show that  $\mathcal{L}$  can be corrected to obtain a  $(\hat{k} - 1)$ -local quasi-distribution  $\mathcal{L}'$  that is close to  $\mathcal{L}$  (see Corollary 7.9). By triangle inequality this implies that  $\mathcal{L}'$  is also close to  $\hat{\mathcal{F}}$ .

See Section 11 for details.

### 3.3 Extending the analysis to non-signaling players

We make a “black-box” use of our results on testing linearity against non-signaling *functions* to derive corresponding results on testing linearity against non-signaling *players*. Recall that, given a  $k$ -non-signaling player  $\mathcal{P} = (\mathcal{P}_{(x_1, \dots, x_k)})_{(x_1, \dots, x_k) \in \{0, 1\}^{k \cdot n}}$ , the linearity test is now as follows: sample  $x, y \in \{0, 1\}^n$  and (distinct)  $i_1, i_2, i_3 \in [k]$  uniformly at random, send the three queries  $x, y, x + y$  to the players  $\mathcal{P}_{i_1}, \mathcal{P}_{i_2}, \mathcal{P}_{i_3}$  respectively, and check that  $\mathcal{P}_{i_1}(x) + \mathcal{P}_{i_2}(y) = \mathcal{P}_{i_3}(x + y)$ .

We prove that if  $\mathcal{P}$  *always* passes the linearity test, then there exists a quasi-distribution  $\mathcal{L}$  over linear functions that matches  $\mathcal{P}$ .

We first argue that  $\mathcal{P}$  must be (almost) *symmetric*, that is,  $\mathcal{P}$ 's answers depend only on the set of asked queries but not also on which players answer these queries. In more detail, we show that, for every subset  $I \subseteq [k]$  of  $|I| = k - 1$  players, it holds that  $\Pr[\mathcal{P}(\vec{x}) = \vec{b}] = \Pr[\mathcal{P}(\pi(\vec{x})) = \pi(\vec{b})]$  for every permutation  $\pi: I \rightarrow I$ , inputs  $\vec{x} = (x_i)_{i \in I} \in (\{0, 1\}^n)^I$ , and answers  $\vec{b} = (b_i)_{i \in I} \in \{0, 1\}^I$ .

We then define a  $(k - 1)$ -non-signaling function  $\mathcal{F}$  that matches  $k - 1$  players of  $\mathcal{P}$  in the natural way (we define  $\Pr[\mathcal{F}(x_1) = b_1, \dots, \mathcal{F}(x_{k-1}) = b_{k-1}]$  to be  $\Pr[\mathcal{P}_1(x_1) = b_1, \dots, \mathcal{P}_{k-1}(x_{k-1}) = b_{k-1}]$ ). By the aforementioned symmetry of  $\mathcal{P}$ , it does not matter which  $k - 1$  players we use to define  $\mathcal{F}$ .

We then argue that  $\mathcal{F}$  always passes the linearity test. Our earlier results imply that there exists a quasi-distribution  $\mathcal{L}$  over linear functions that matches  $\mathcal{F}$  on all subsets of at most  $k - 2$  queries. By definition of  $\mathcal{F}$  this implies that  $\mathcal{L}$  *also* matches the players  $\mathcal{P}_1, \dots, \mathcal{P}_{k-2}$ , and, using the symmetry of  $\mathcal{P}$ , we conclude that  $\mathcal{L}$  also matches *every* subset of  $k - 2$  players.

We now relax the assumption that  $\mathcal{P}$  passes the linearity test with probability 1 to merely that it passes the test with probability  $1 - \varepsilon$  for a small enough  $\varepsilon > 0$ .

Similarly to the case of non-signaling functions, we define a self-correction  $\hat{\mathcal{P}}$  of  $\mathcal{P}$  in the natural way: it is a  $\hat{k}$ -non-signaling player (for  $\hat{k} := k/2$ ) that, given a query  $(x_1, \dots, x_{\hat{k}}) \in \{0, 1\}^{\hat{k} \times n}$ , samples  $w_1, \dots, w_{\hat{k}} \in \{0, 1\}^n$  and a permutation  $\pi: [k] \rightarrow [k]$  uniformly at random, and answers each  $x_i$  with  $\mathcal{P}_{\pi(2i)}(x_i + w_i) + \mathcal{P}_{\pi(2i+1)}(w_i)$ .

We show that  $\hat{\mathcal{P}}$  is (fully) symmetric and that, for every  $x, y \in \{0, 1\}^n$  and distinct  $i_1, i_2, i_3 \in [\hat{k}]$ ,  $\Pr[\hat{\mathcal{P}}_{i_1}(x) + \hat{\mathcal{P}}_{i_2}(y) = \hat{\mathcal{P}}_{i_3}(x + y)] > 1 - \hat{\varepsilon}$  for  $\hat{\varepsilon} := 4\varepsilon$ . This is analogous to the average-case-to-worst-case statement that we showed for non-signaling functions. We define a  $\hat{k}$ -non-signaling function  $\hat{\mathcal{F}}$  that matches  $\hat{\mathcal{P}}$  similarly to the above (by letting  $\Pr[\hat{\mathcal{F}}(x_1) = b_1, \dots, \hat{\mathcal{F}}(x_{\hat{k}}) = b_{\hat{k}}] := \Pr[\hat{\mathcal{P}}_1(x_1) = b_1, \dots, \hat{\mathcal{P}}_{\hat{k}}(x_{\hat{k}}) = b_{\hat{k}}]$ ), and show that it satisfies the analogous worst-case property, that is,  $\Pr[\hat{\mathcal{F}}(x) + \hat{\mathcal{F}}(y) = \hat{\mathcal{F}}(x + y)] \geq 1 - \hat{\varepsilon}$  for every  $x, y \in \{0, 1\}^n$ . Our earlier results imply that there exists a quasi-distribution  $\mathcal{L}$  over linear functions that is close to  $\hat{\mathcal{F}}$ , and thus also close to  $\hat{\mathcal{P}}$ .

See full version for details.

## 4 Discussion and open problems

The study of non-signaling strategies in Physics is motivated by the goal of understanding the power and limitations of Nature's apparent non-locality [35, 43, 40, 41, 8]. Prior work has explored many topics, including the inter-convertibility between quantum strategies and non-signaling strategies [19, 11, 10, 31, 17]; communication complexity with non-signaling strategies [51, 16]; non-local computation [36]; using non-signaling strategies to achieve key distribution, oblivious transfer, and bit commitments [9, 53, 18, 49, 48]; and many others [38, 27, 20].

More recently, researchers have established connections with Complexity Theory and Cryptography. Property Testing against non-signaling strategies, the subject of our work, is likely to lead to a deeper understanding of these.

### 4.1 Powers and limitations of non-local strategies

Understanding the computational complexity of computing or approximating the value of certain classes of games is a fundamental problem in Complexity Theory. Games are typically phrased in terms of one or more *non-communicating* players that interact with a probabilistic polynomial-time Referee (with polynomial randomness), who decides at the end of the game if the players win or not. The complexity of these games is well-understood.

- Results on *Interactive Proofs* (IPs) [37, 46] imply that approximating the value of single-player games is PSPACE-complete, when given enough rounds.
- Results on *Multi-prover Interactive Proofs* (MIPs) [7] imply that approximating the value of multi-player games is NEXP-complete, even with only two players.
- Results on *Probabilistically Checkable Proofs* (PCPs) [6, 24, 4, 3] imply that, if the player's strategy is non-adaptive (the player merely answers queries from the Referee) then approximating the game's value is NEXP-complete, even if the Referee asks only a constant number of queries and receives answers over a constant-size alphabet.

However, if the players can use any non-signaling strategy to win the game, *much less is known*.

If there are only *two* players, then approximating the game's value is PSPACE-complete [29, 28]. If the game has  $k$  players then its value can be computed in time  $\text{poly}(2^{kr}, |\Sigma|^k)$ , where  $r$  is the Referee's randomness complexity and  $\Sigma$  is each player's answer's alphabet [23], which means that this computation lies in EXP. This is *very unlike* the case of non-communicating players.

However, hardness results for this problem in the case of three or more players have been elusive. Recently, Kalai, Raz, and Rothblum [32, 34] established EXP-hardness for the case of polynomially-many provers, via a reduction from deterministic-time languages.

► **Theorem 4.1** ([32, 34]). *Let  $L$  be a language decidable in time  $T: \mathbb{N} \rightarrow \mathbb{N}$ . There exists a constant  $c > 0$  such that, for any function  $\lambda: \mathbb{N} \rightarrow \mathbb{N}$  with  $\lambda \geq \log^c T$ ,  $L$  has a  $(\lambda \log^c T)$ -prover MIP with soundness error  $2^{-\lambda}$  against non-signaling players. The verifier runs in time  $n\lambda^2 \log^c T$  and the provers in time  $\text{poly}(T, \lambda)$ ; each query and answer consists of  $\lambda \log^c T$  bits.*

The above theorem is proved by constructing a PCP verifier that is secure against non-signaling functions (Definition 6.1), which can then be compiled into an MIP verifier that is secure against non-signaling players. The proof is a technical tour-de-force showing that a modification of the “classical” PCP verifier in [7, 6] is secure against non-signaling functions.

The huge gap between the EXP-completeness for polynomially-many provers and the PSPACE-completeness for two provers motivates a natural question:

*Is there a non-signaling analogue of the PCP Theorem? I.e., does EXP have  $O(1)$ -query PCPs over a  $O(1)$ -size alphabet that are secure against non-signaling functions? (Equivalently,  $O(1)$ -prover MIPs over a  $O(1)$ -size alphabet that are secure against non-signaling players?)*

We believe that initiating a study of Property Testing against non-signaling strategies will drive progress on this question. In particular, linearity testing is one of the ingredients of the (classical) PCP Theorem, and linearity testing against non-signaling strategies may be a good place to start.

We also believe that Property Testing against non-signaling strategies may play a significant *simplifying role*, which could itself drive progress on this and other questions. Indeed, the analysis of classical PCP constructions (including [7, 6]) is carried out in two conceptually simple steps: first argue soundness assuming that the PCP is a low-degree function, and then rely on low-degree testing and self-correction to ensure that the PCP is close to a low-degree function [45, 44, 5]. The study of this latter step as a standalone problem in the area of Property Testing has enabled much progress on PCP research. In contrast, while the analysis in [32, 34] does analyze low-degree tests by proving certain average-case-to-worst-case statements, it *does not prove any local-to-global phenomena* for the property of “low-degreeness”.

We prove a first local-to-global phenomenon for Property Testing against non-signaling strategies. However, whether Property Testing is feasible beyond the case of linearity testing (our focus) and whether it plays a beneficial and simplifying role in PCP research are fascinating open problems.

## 4.2 Hardness of approximation

Feige et al. [24] showed a fundamental connection between MIPs/PCPs and the hardness of approximating values of constraint satisfaction problems. Kalai, Raz, and Regev [33] recently established a similar connection, this time between *non-signaling* MIPs/PCPs and the hardness of approximating values of *linear programs*. While the first connection considers approximation algorithms that are bounded in time, the second connection considers approximation algorithms that are bounded in *space*. We recall [33]’s result and its relation to our results.

► **Theorem 4.2** ([33]). *Let  $L$  be a language with a 1-round  $k$ -prover MIP with soundness error  $\epsilon$  against non-signaling players in which:*

- (i) *the verifier has time complexity  $T$ , space complexity  $S$ , and randomness complexity  $r$ ;*
- (ii) *the prover's answers are symbols in  $\Sigma$ .*

*Then there is a family of polyhedra  $\{H_n\}_{n \in \mathbb{N}}$  and a  $\text{poly}(2^{kr}, |\Sigma|^k, T)$ -time  $\text{poly}(k, r, S)$ -space reduction  $\mathcal{R}$  such that:*

- (i) *For every instance  $x \in \{0, 1\}^*$ ,  $\mathcal{R}(x)$  is a linear program with polyhedron  $H_{|x|}$  and with  $\text{poly}(2^{kr}, |\Sigma|^k)$  variables and constraints.*
- (ii) *If  $x \in L$ , then the value of the linear program  $\mathcal{R}(x)$  is 1.*
- (iii) *If  $x \notin L$ , then the value of the linear program  $\mathcal{R}(x)$  is at most  $\epsilon$ .*

The above result, when combined with the non-signaling MIPs for deterministic-time languages of [32, 34] (see Section 4.1), implies that a  $2^{\log^{o(1)}(n)}$ -space approximation algorithm for linear programming is unlikely, even when given unbounded computation based on the polyhedron. (Since that would imply, in particular, that every problem in P can be solved in  $2^{\log^{o(1)}(n)}$ -space.)

The above conclusion, however, appears sub-optimal because both  $2^{kr}$  and  $|\Sigma|^k$  are super-polynomial in the construction of [32, 34]. Ideally, we would like a construction where  $r = O(\log n)$  and  $k = O(1)$ , which again (as discussed in Section 4.1) leads to the question of whether there is a non-signaling analogue of the PCP Theorem. We conjecture that the study of Property Testing against non-signaling strategies is again very relevant.

### 4.3 One-round delegation of computation

Delegation of computation is a fundamental goal in Cryptography that involves designing protocols that enable a weak verifier to outsource expensive computations to a powerful but untrusted prover.

A key efficiency measure is round complexity (the number of back-and-forth messages between the verifier and prover). Aiello et al. [1] suggested a cryptographic method to transform any 1-round MIP into a 1-round delegation protocol, but did not provide a proof of security. Later on, Dwork et al. [23] showed that this method is not secure in the general case, by exhibiting a 1-round MIP for which the transformation yields a delegation protocol that can be fooled.

Nevertheless, Kalai, Raz, and Rothblum [32] proved that if the 1-round MIP used in the method is sound against non-signaling players then the resulting delegation protocol *cannot* be fooled (namely, is secure). More precisely, the 1-round MIP must be sound not only against all players that are non-signaling but also against all players that are *almost* non-signaling (see full version for details), where “almost” denotes a certain parameter that depends on the security reduction.

By invoking this method on the MIP of [32, 34] (which *is* secure against almost non-signaling players), one obtains a delegation protocol for all polynomial-time functions in which the prover runs in polynomial time and the verifier in polylogarithmic time.

Yet, the seemingly sub-optimal parameters of the MIP of [32, 34] suggest that there is room to improve efficiency by invoking the method on more efficient MIPs. For example:

*Is there an almost non-signaling analogue of the PCP Theorem?*

Namely, does EXP have  $O(1)$ -query PCPs (equivalently,  $O(1)$ -prover MIPs) over a  $O(1)$ -size alphabet that are secure against almost non-signaling strategies?

The study of Property Testing against almost non-signaling strategies is likely a first step, and our work establishes first results for *exact* non-signaling strategies.

► **Remark 4.3** (extension to almost non-signaling). *While almost non-signaling strategies are not our focus, in this paper we do show that almost non-signaling strategies are not outside the reach of tools that we use. Concretely, we show that every almost non-signaling function is “reasonably close” to a corresponding (exact) non-signaling function. The proof of this statement uses Fourier analysis, and the intuition behind it is similar to how almost-feasible solutions to Sherali–Adams relaxations are “smoothened” into feasible ones [42]. The generic lemma enables us, for example, to extend Theorem 2.4 to the case of almost non-signaling strategies. Whether a whitebox analysis of linearity testing against almost non-signaling strategies can improve upon such a blackbox extension remains an interesting open problem. See full version for details.*

## 5 Preliminaries

For a finite domain  $D$ , we denote by  $U_D$  the set of all boolean functions  $f: D \rightarrow \{0, 1\}$ ; when  $D$  is clear from context, we may omit the subscript in  $U_D$ . When  $D = \{0, 1\}^n$ , a function  $f \in U_{\{0,1\}^n}$  is *linear* if  $f(x) + f(y) = f(x + y)$  for all  $x, y \in \{0, 1\}^n$ ;  $\text{LIN}$  is the set of all such linear functions.

### 5.1 Fourier analysis of boolean functions

We use standard notation for Fourier analysis of boolean functions (see [39] for more details). For a domain  $D$  of size  $N$ , we consider functions  $f: \{0, 1\}^D \rightarrow \mathbb{R}$ . The inner product of two functions  $g_1, g_2: \{0, 1\}^D \rightarrow \mathbb{R}$  is  $\langle g_1, g_2 \rangle := \frac{1}{2^N} \sum_{x \in \{0,1\}^D} g_1(x)g_2(x)$ . For a subset  $T \subseteq D$ ,  $\chi_T: \{0, 1\}^D \rightarrow \mathbb{R}$  is the parity function  $\chi_T(x) = (-1)^{\sum_{i \in T} x_i}$ . It is not hard to verify that the set of functions  $\{\chi_T\}_{T \subseteq D}$  is an orthonormal basis of the space of all functions from  $\{0, 1\}^D$  to  $\mathbb{R}$ . In particular, every function  $f: \{0, 1\}^D \rightarrow \mathbb{R}$  can be written as

$$f(\cdot) = \sum_{T \subseteq D} \widehat{f}(T) \chi_T(\cdot) ,$$

where  $\widehat{f}(T) = \langle f, \chi_T \rangle = \frac{1}{2^N} \sum_{x \in \{0,1\}^D} f(x) \chi_T(x)$ . In particular, by Parseval’s identity for any two functions  $f, g: \{0, 1\}^D \rightarrow \mathbb{R}$  we have

$$\frac{1}{2^N} \sum_{x \in \{0,1\}^D} f(x)g(x) = \sum_{T \subseteq D} \widehat{f}(T) \widehat{g}(T) ,$$

which implies Plancherel’s identity

$$\frac{1}{2^N} \sum_{x \in \{0,1\}^D} f(x)^2 = \sum_{T \subseteq D} \widehat{f}(T)^2 .$$

For a set  $E \subseteq \{0, 1\}^s$ , its indicator function  $\mathbf{1}_E: \{0, 1\}^s \rightarrow \{0, 1\}$  is defined as

$$\mathbf{1}_E = \begin{cases} 1 & \text{if } x \in E \\ 0 & \text{otherwise} \end{cases} .$$

Note that by Plancherel’s identity we have  $\sum_{T \subseteq [s]} \widehat{\mathbf{1}_E}(T)^2 = \mathbb{E}[\mathbf{1}_E] = \frac{|E|}{2^s}$ . In particular, this implies  $\|\widehat{\mathbf{1}_E}\|_1 = \sum_{T \subseteq [s]} |\widehat{\mathbf{1}_E}(T)| \leq \sqrt{\sum_{T \subseteq [s]} \widehat{\mathbf{1}_E}(T)^2} \cdot \sqrt{\sum_{T \subseteq [s]} 1} \leq \sqrt{\frac{|E|}{2^s}} \cdot 2^{s/2} = \sqrt{|E|}$ .

## 5.2 Expressing boolean events as sums of parities

We state two lemmas that express the probability of certain events as probabilities about the *parities* of related events.

► **Lemma 5.1.** *Let  $X_1, \dots, X_s$  be boolean random variables. Then, for every event  $E \subseteq \{0, 1\}^s$  it holds that*

$$\Pr[(X_1, \dots, X_s) \in E] = \sum_{T \subseteq [s]} c_T \cdot \Pr \left[ \sum_{i \in T} X_i = 0 \right],$$

where  $c_\emptyset = 2 \cdot \widehat{\mathbf{1}}_E(\emptyset) - \mathbf{1}_E(\vec{0})$ , and  $c_T = 2 \cdot \widehat{\mathbf{1}}_E(T)$  for all  $T \neq \emptyset$ . In particular,  $c_T$ 's depend only on  $E$  and  $\sum_{T \subseteq [s]} |c_T| \leq 3 \|\widehat{\mathbf{1}}_E\|_1 \leq 3\sqrt{|E|}$ .

► **Corollary 5.2.** *Let  $X_1, \dots, X_s$  be boolean random variables. Then, for every  $\vec{b} = (b_1, \dots, b_s)$  in  $\{0, 1\}^s$  it holds that*

$$\Pr[X_1 = b_1, \dots, X_s = b_s] = -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \Pr \left[ \sum_{i \in T} X_i = \sum_{i \in T} b_i \right].$$

**Proof of Lemma 5.1.** Define  $p: \{0, 1\}^s \rightarrow \mathbb{R}$  as  $p(\vec{a}) = \Pr[X_1 = a_1, \dots, X_s = a_s]$ , and write  $p = \sum_{T \subseteq [s]} \hat{p}(T) \cdot \chi_T$ . We have

$$\begin{aligned} \hat{p}(T) &= \mathbb{E}[p(\vec{a}) \cdot \chi_T(\vec{a})] \\ &= \frac{1}{2^s} \left( \sum_{\vec{a}: \sum_{i \in T} a_i = 0} p(\vec{a}) - \sum_{\vec{a}: \sum_{i \in T} a_i = 1} p(\vec{a}) \right) \\ &= \frac{1}{2^s} \left( 2 \sum_{\vec{a}: \sum_{i \in T} a_i = 0} p(\vec{a}) - 1 \right) \\ &= \frac{1}{2^s} \left( 2 \Pr \left[ \sum_{i \in T} X_i = 0 \right] - 1 \right) \end{aligned}$$

Let  $E \subseteq \{0, 1\}^s$  be an event, and let  $\mathbf{1}_E: \{0, 1\}^s \rightarrow \{0, 1\}$  be its indicator function. Then, by Parseval's identity we have

$$\Pr[(X_1, \dots, X_s) \in E] = \sum_{\vec{a} \in \{0, 1\}^s} p(\vec{a}) \cdot \mathbf{1}_E(\vec{a}) = 2^s \cdot \sum_{T \subseteq [s]} \hat{p}(T) \cdot \widehat{\mathbf{1}}_E(T).$$

By plugging in the formula  $\hat{p}(T) = \frac{1}{2^s} (2 \Pr[\sum_{i \in T} X_i = 0] - 1)$ , and using  $\Pr[\sum_{i \in \emptyset} X_i = 0] = 1$  we get

$$\Pr[(X_1, \dots, X_s) \in E] = \sum_{T \subseteq [s]} \left( 2 \Pr \left[ \sum_{i \in T} X_i = 0 \right] - 1 \right) \cdot \widehat{\mathbf{1}}_E(T) = \sum_{T \subseteq [s]} c_T \cdot \Pr \left[ \sum_{i \in T} X_i = 0 \right],$$

where  $c_\emptyset = 2 \cdot \widehat{\mathbf{1}}_E(\emptyset) - \mathbf{1}_E(\vec{0})$ , and  $c_T = 2 \cdot \widehat{\mathbf{1}}_E(T)$  for all  $T \neq \emptyset$ . Since  $\mathbf{1}_E(\cdot) = \sum_{T \subseteq [s]} \widehat{\mathbf{1}}_E(T) \chi_T(\cdot)$ , it follows that  $\mathbf{1}_E(\vec{0}) = \sum_{T \subseteq [s]} \widehat{\mathbf{1}}_E(T)$ , as required.

Thus, by the argument in Section 5.1 we have  $\sum_{T \subseteq [s]} |c_T| \leq 3 \sum_{T \subseteq [s]} |\widehat{\mathbf{1}}_E(T)| \leq 3\sqrt{|E|}$ . ◀

**Proof of Corollary 5.2.** Let  $E = \{\vec{b}\}$  be the singleton event. It is easy to verify that  $\widehat{\mathbf{1}}_E(T) = (-1)^{\sum_{i \in T} b_i} \cdot 2^{-s}$ . Therefore, by Lemma 5.1 we have

$$\Pr[X_1 = b_1, \dots, X_s = b_s] = \sum_{T \subseteq [s]} c_T \cdot \Pr \left[ \sum_{i \in T} X_i = 0 \right],$$

where  $c_\emptyset = 2 \cdot \widehat{\mathbf{1}}_E(\emptyset) - \mathbf{1}_E(\vec{0}) = \frac{1}{2^{s-1}} - \mathbf{1}_E(\vec{0})$ , and  $c_T = (-1)^{\sum_{i \in T} b_i} \cdot 2^{-s+1}$  for all  $T \neq \emptyset$ . By substituting  $\Pr \left[ \sum_{i \in T} X_i = 0 \right]$  with  $1 - \Pr \left[ \sum_{i \in T} X_i = 1 \right]$  for all  $T \subseteq [s]$  such that  $\sum_{i \in T} b_i = 1$  we get

$$\begin{aligned} \Pr[X_1 = b_1, \dots, X_s = b_s] &= \sum_{T \subseteq [s]} c_T \cdot \Pr \left[ \sum_{i \in T} X_i = 0 \right] \\ &= \left( -\mathbf{1}_E(\vec{0}) - \sum_{T: \sum_{i \in T} b_i = 1} \frac{1}{2^{s-1}} \right) + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \Pr \left[ \sum_{i \in T} X_i = \sum_{i \in T} b_i \right] \\ &= -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \Pr \left[ \sum_{i \in T} X_i = \sum_{i \in T} b_i \right], \end{aligned}$$

as required.  $\blacktriangleleft$

### 5.3 A linear system

Below we prove that a certain linear system of equations, which we will use later, has a unique solution. This linear system is the inverse of the Hadamard–Walsh matrix.

► **Lemma 5.3.** *For every positive integer  $n$  and real numbers  $\{c_\beta\}_{\beta \in \{0,1\}^n}$ , the system of  $2^n$  linear equations over  $\mathbb{R}$  in  $2^n$  variables  $\{z_\alpha\}_{\alpha \in \{0,1\}^n}$  given by*

$$\left\{ \begin{array}{l} \forall \beta \in \{0,1\}^n \quad \sum_{\substack{\alpha \in \{0,1\}^n \\ \text{s.t. } \langle \alpha, \beta \rangle = 0}} z_\alpha = c_\beta \end{array} \right\}$$

*has a unique solution.*

**Proof.** Let  $A$  be the  $2^n \times 2^n$  boolean matrix corresponding to the system of linear equations, that is, such that  $Az = c$ . Note that the  $(\beta, \alpha)$ -th entry of  $A$  is equal to  $1 - \langle \alpha, \beta \rangle$ , and in particular, the row in  $A$  corresponding to  $\beta = 0^n$  is the all-ones row. Define  $H$  to be the matrix obtained from  $A$  by performing the following elementary row operations: for every  $\beta \neq 0^n$ , multiply row  $\beta$  by 2 and then subtract the all-ones row (corresponding to  $\beta = 0^n$ ).

Note that the  $(\beta, \alpha)$ -th entry of  $H$  is equal to  $(-1)^{\langle \alpha, \beta \rangle}$ . (The matrix  $H$  is sometimes called the Hadamard–Walsh matrix.) Indeed, this holds trivially for the row  $\beta = 0^n$  as  $H_{\beta, \alpha} = (-1)^{\langle \alpha, 0^n \rangle} = 1$ , and for  $\beta \neq 0^n$  we have  $H_{\beta, \alpha} = 2(1 - \langle \alpha, \beta \rangle) - 1 = 1 - 2\langle \alpha, \beta \rangle = (-1)^{\langle \alpha, \beta \rangle}$ . Since  $H$  was obtained from  $A$  by performing elementary row operations,  $A$  is invertible if and only if  $H$  is invertible. Observe that  $H$  is indeed invertible because the rows of  $H$  are mutually orthogonal since for every two distinct  $\beta$  and  $\gamma$  in  $\{0,1\}^n$  it holds that

$$\langle \text{row } \beta, \text{row } \gamma \rangle = \sum_{\alpha} (-1)^{\langle \alpha, \beta \rangle} (-1)^{\langle \alpha, \gamma \rangle} = \sum_{\alpha} (-1)^{\langle \alpha, \beta \rangle + \langle \alpha, \gamma \rangle} = \sum_{\alpha} (-1)^{\langle \alpha, \beta + \gamma \rangle} = 0,$$

where the last equality holds because  $\beta + \gamma \neq 0^n$ .  $\blacktriangleleft$

## 6 Non-signaling functions

We define *non-signaling functions*, introduce useful notation for them, and prove a simple lemma about them. The notions described here are used throughout the paper.

► **Definition 6.1** (non-signaling functions). A *k-non-signaling (boolean) function* over a finite domain  $D$  is a collection  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq D, |S| \leq k}$  where

- (i) each  $\mathcal{F}_S$  is a distribution over functions  $f: S \rightarrow \{0, 1\}$ , and
- (ii) for every two subsets  $S$  and  $T$  each of size at most  $k$ , the restrictions of  $\mathcal{F}_S$  and  $\mathcal{F}_T$  to  $S \cap T$  are equal as distributions.

(If  $S = \emptyset$  then  $\mathcal{F}_S$  always outputs the empty string.)

Given a set  $S \subseteq D$  of size  $|S| \leq k$  and a string  $\vec{b} \in \{0, 1\}^S$ , we define

$$\Pr[\mathcal{F}(S) = \vec{b}] := \Pr_{f \leftarrow \mathcal{F}_S} [f(S) = \vec{b}] .$$

The non-signaling property in this notation is the following: for every two subsets  $S, T \subseteq D$  of sizes  $|S|, |T| \leq k$  and every string  $\vec{b} \in \{0, 1\}^{S \cap T}$ ,  $\Pr[\mathcal{F}(S)|_{S \cap T} = \vec{b}] = \Pr[\mathcal{F}(T)|_{S \cap T} = \vec{b}]$ .

Sometimes it is more convenient to consider a *vector* of inputs (rather than a *set*), and so we define notation for this case. Given a vector  $\langle x_1, \dots, x_s \rangle$  with entries in  $D$  and a vector  $\langle b_1, \dots, b_s \rangle$  with entries in  $\{0, 1\}$  (with  $s \in \{1, \dots, k\}$ ), we define  $\Pr[\mathcal{F}(\langle x_1, \dots, x_s \rangle) = \langle b_1, \dots, b_s \rangle]$  and  $\Pr[\mathcal{F}(x_1) = b_1, \dots, \mathcal{F}(x_s) = b_s]$  to be the probability

$$\Pr_{f \leftarrow \mathcal{F}_{\{x_1, \dots, x_s\}}} [f(x_1) = b_1, \dots, f(x_s) = b_s] .$$

Note that  $\{x_1, \dots, x_s\}$  is an unordered set and its size may be less than  $s$ , because the entries of the vector  $\langle x_1, \dots, x_s \rangle$  may not be distinct. We abuse notation and still use symbols such as  $S$  and  $\vec{b}$  to denote vectors as above. We stress that we use an ordering on  $S$  merely to match each element of  $S$  to the corresponding element in  $\vec{b}$ ; the event remains unchanged if one permutes the entries of  $S$  and  $\vec{b}$  according to the same permutation.

► **Remark 6.2** (Sherali–Adams hierarchy). *We note that k-non-signaling functions are solutions to the linear program arising from the k-relaxation in the Sherali–Adams hierarchy [47]. The variables are of the form  $X_{S, \vec{b}}$  (for all  $S \subseteq D$  of size at most  $k$  and  $\vec{b} \in \{0, 1\}^S$ ) and express  $\Pr[\mathcal{F}(S) = \vec{b}]$ . Consistency across subsets  $S$  and  $T$  is expressed using the natural linear constraints.<sup>2</sup>*

We conclude with a useful lemma.

► **Lemma 6.3.** *Let  $\mathcal{F}$  be a k-non-signaling function over a domain  $D$ , let  $S_1, S_2$  be subsets of  $D$  with  $|S_1 \cup S_2| \leq k$ , and let  $g_1: \{0, 1\}^{S_1} \rightarrow \{0, 1\}^r$  and  $g_2: \{0, 1\}^{S_2} \rightarrow \{0, 1\}^r$  be functions. If  $\Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = g_2(\mathcal{F}(S_2))] \geq 1 - \varepsilon$ , then for every  $\vec{b} \in \{0, 1\}^r$  it holds that*

$$\left| \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b}] - \Pr_{\mathcal{F}}[g_2(\mathcal{F}(S_2)) = \vec{b}] \right| \leq \varepsilon .$$

*In particular, if  $\varepsilon = 0$  then  $\Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b}] = \Pr_{\mathcal{F}}[g_2(\mathcal{F}(S_2)) = \vec{b}]$  for every  $\vec{b} \in \{0, 1\}^r$ .*

<sup>2</sup> In fact it suffices to only have variables of the form  $X_{S, 1^S}$  as all other probabilities can be computed from these.



**Proof.** By direct computation:

$$\begin{aligned}
& \left| \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b}] - \Pr_{\mathcal{F}}[g_2(\mathcal{F}(S_2)) = \vec{b}] \right| \\
&= \left| \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b} \wedge g_2(\mathcal{F}(S_2)) = \vec{b}] + \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b} \wedge g_2(\mathcal{F}(S_2)) \neq \vec{b}] \right. \\
&\quad \left. - \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b} \wedge g_2(\mathcal{F}(S_2)) = \vec{b}] - \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) \neq \vec{b} \wedge g_2(\mathcal{F}(S_2)) = \vec{b}] \right| \\
&= \left| \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b} \wedge g_2(\mathcal{F}(S_2)) \neq \vec{b}] - \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) \neq \vec{b} \wedge g_2(\mathcal{F}(S_2)) = \vec{b}] \right| \\
&\leq \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) = \vec{b} \wedge g_2(\mathcal{F}(S_2)) \neq \vec{b}] + \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) \neq \vec{b} \wedge g_2(\mathcal{F}(S_2)) = \vec{b}] \\
&\leq \Pr_{\mathcal{F}}[g_1(\mathcal{F}(S_1)) \neq g_2(\mathcal{F}(S_2))] \leq \varepsilon .
\end{aligned}$$

Note that we are implicitly using the fact that  $|S_1 \cup S_2| \leq k$  whenever we have  $S_1$  and  $S_2$  in the same probability event because we are querying  $\mathcal{F}$  on all inputs in  $S_1 \cup S_2$  at once.  $\blacktriangleleft$

## 7 Quasi-distributions

A quasi-distribution extends the notion of a probability distribution by allowing probabilities to be negative, and is the main tool that we use to analyze non-signaling functions.

► **Definition 7.1** (quasi-distributions). Let  $D$  be a finite domain, and denote by  $U_D$  the set of all boolean functions of the form  $f: D \rightarrow \{0, 1\}$ . A **quasi-distribution**  $\mathcal{Q}$  over a subset  $G \subseteq U_D$  is a set of real numbers  $\{q_f\}_{f \in U_D}$  such that  $\sum_{f \in U_D} q_f = 1$  and  $q_f = 0$  for every  $f \notin G$ .

► **Definition 7.2** (quasi-probability). Given a quasi-distribution  $\mathcal{Q} = \{q_f\}_{f \in U_D}$ , a subset  $S \subseteq D$ , and a string  $\vec{b} \in \{0, 1\}^S$ , we define the **quasi-probability** of the event “ $\mathcal{Q}(S) = \vec{b}$ ” to be the following (possibly negative) real number

$$\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] := \sum_{f \in U_D \text{ s.t. } f(S) = \vec{b}} q_f .$$

As in the case of non-signaling functions, it is sometimes more convenient to consider a *vector* of inputs rather than a *set*. Given a vector  $\langle x_1, \dots, x_s \rangle$  with entries in  $D$  and a vector  $\langle b_1, \dots, b_s \rangle$  with entries in  $\{0, 1\}$ , we define  $\Pr[\mathcal{Q}(\langle x_1, \dots, x_s \rangle) = \langle b_1, \dots, b_s \rangle]$  and  $\Pr[\mathcal{Q}(x_1) = b_1, \dots, \mathcal{Q}(x_s) = b_s]$  to be the (possibly negative) real number  $\sum_{f \in U_D \text{ s.t. } \forall i f(x_i) = b_i} q_f$ . We abuse notation and still use symbols such as  $S$  and  $\vec{b}$  to denote vectors as above.

Since a quasi-distribution  $\mathcal{Q}$  is defined by its weights  $q = (q_f)_{f \in U_D}$ , we can view  $\mathcal{Q}$  as a function from  $\{0, 1\}^D$  to  $\mathbb{R}$ , where we identify a function  $f: D \rightarrow \{0, 1\}$  with the corresponding vector in  $\{0, 1\}^D$  and  $q(f)$  with  $q_f$ . In particular, we can write  $q(\cdot) = \sum_{T \subseteq D} \widehat{q}(T) \chi_T(\cdot)$ , where  $\chi_T(f) = (-1)^{\sum_{x \in T} f(x)}$ , and  $\widehat{q}(T) = \langle q, \chi_T \rangle = \frac{1}{2^{|D|}} \sum_{f: D \rightarrow \{0, 1\}} q(f) \chi_T(f)$ .

The following lemma is an analogue of Lemma 5.1 for quasi-distributions.

► **Lemma 7.3.** Let  $\mathcal{Q} = (q_f)_f$  be a quasi-distribution,  $S = \langle x_1, \dots, x_s \rangle$  a vector with entries in  $\{0, 1\}^n$ . Then, for every event  $E \in \{0, 1\}^s$  it holds that

$$\sum_{f: f(S) \in E} q_f = \sum_{T \subseteq [s]} c_T \cdot \widetilde{\Pr} \left[ \sum_{i \in T} \mathcal{Q}(x_i) = 0 \right] = \sum_{T \subseteq [s]} c_T \cdot \left( \sum_{f: \sum_{i \in T} f(x_i) = 0} q_f \right) ,$$

where  $c_\emptyset = 2 \cdot \widehat{\mathbf{1}}_E(\emptyset) - \mathbf{1}_E(\vec{0})$ , and  $c_T = 2 \cdot \widehat{\mathbf{1}}_E(T)$  for all  $T \neq \emptyset$ .

## 17:22 Testing Linearity against Non-Signaling Strategies

The proof of the lemma is immediate from the proof of Lemma 5.1, since the proof only uses the fact that probabilities add up to 1, which also holds for quasi-probabilities.

► **Definition 7.4** (locality). Let  $D$  be a finite domain of size  $N$ . For  $1 \leq \ell \leq N$  a quasi-distribution  $\mathcal{Q}$  over  $U_D$  is  $\ell$ -**local** if for every subset  $S \subseteq D$  of size  $|S| \leq \ell$  and string  $\vec{b} \in \{0, 1\}^S$ ,

$$\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] \in [0, 1] .$$

For completeness, we also say that all quasi-distributions are 0-local.

If  $\mathcal{Q}$  is  $\ell$ -local, then for every subset  $S \subseteq D$  of size  $|S| \leq \ell$ , we may view  $\mathcal{Q}(S)$  as a probability distribution over  $\{0, 1\}^S$ . If  $\mathcal{Q}$  is  $\ell$ -local then it is  $s$ -local for every  $s \in \{0, 1, \dots, \ell\}$ .

For  $\mathcal{Q}$  to be  $\ell$ -local, it suffices for all relevant  $\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}]$  to be non-negative (as opposed to be in  $[0, 1]$ ). This is because  $\sum_f q_f = 1$ , so that  $\sum_{\vec{b} \in \{0, 1\}^S} \widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] = 1$  and, if all terms in this sum are non-negative, then we can deduce that  $\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] \leq 1$  for every  $\vec{b}$ .

► **Definition 7.5** (statistical distance). Given a finite domain  $D$  and an integer  $\ell \in \{1, \dots, |D|\}$ , the  $\Delta_\ell$ -**distance** between two quasi-distributions  $\mathcal{Q}$  and  $\mathcal{Q}'$  is

$$\Delta_\ell(\mathcal{Q}, \mathcal{Q}') := \max_{S \subseteq D, |S| \leq \ell} \Delta(\mathcal{Q}_S, \mathcal{Q}'_S) ,$$

where  $\Delta(\mathcal{Q}_S, \mathcal{Q}'_S) := \max_{E \subseteq \{0, 1\}^S} \left| \widetilde{\Pr}[\mathcal{Q}(S) \in E] - \widetilde{\Pr}[\mathcal{Q}'(S) \in E] \right|$ .

We say that  $\mathcal{Q}$  and  $\mathcal{Q}'$  are  $\varepsilon$ -**close in the  $\Delta_\ell$ -distance** if  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}') \leq \varepsilon$ ; else, they are  $\varepsilon$ -far.

► **Remark 7.6** (distance for non-signaling functions). *The definition of  $\Delta_\ell$ -distance naturally extends to defining distances between  $k$ -non-signaling functions, as well as between quasi-distributions and  $k$ -non-signaling functions, provided that  $\ell \leq k$ .*

The notion above generalizes the standard notion of statistical (total variation) distance: if  $\mathcal{Q}$  and  $\mathcal{Q}'$  are *distributions* then their  $\Delta_{|D|}$ -distance equals their statistical distance. Also note that if  $\mathcal{Q}$  and  $\mathcal{Q}'$  are  $\ell$ -local quasi-distributions then their  $\Delta_\ell$ -distance equals the maximum statistical distance, across all subsets  $S \subseteq D$  with  $|S| \leq \ell$ , between the two *distributions*  $\mathcal{Q}_S$  and  $\mathcal{Q}'_S$  — in particular this means that any experiment that queries exactly one set of size at most  $\ell$  cannot distinguish between the two quasi-distributions with probability greater than  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}')$ .

We stress that  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}') = 0$  does *not* necessarily mean that  $\mathcal{Q} = \mathcal{Q}'$ ! In fact, it is possible to have  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}') = 0$  while  $\sum_{f \in U} |q_f - q'_f|$  is arbitrarily large. We also remark that the  $\Delta_\ell$ -distance is not necessarily upper bounded by 1, and is in general unbounded.

► **Definition 7.7** (approximate locality). Given a finite domain  $D$ , an integer  $\ell \in \{1, \dots, |D|\}$ , and a real number  $\varepsilon \geq 0$ , a quasi-distribution  $\mathcal{Q}$  over  $U_D$  is  $(\ell, \varepsilon)$ -**local** if, for every subset  $S \subseteq D$  of size  $|S| \leq \ell$  and every event  $E \subseteq \{0, 1\}^S$ ,

$$\widetilde{\Pr}[\mathcal{Q}(S) \in E] \in [-\varepsilon, 1 + \varepsilon] .$$

Approximate locality generalizes the notion of (exact) locality as in Definition 7.4. Indeed, note that in Definition 7.4 the condition is point-wise, i.e.,  $\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] \in [0, 1]$  for each  $\vec{b} \in \{0, 1\}^S$ . However, this is in fact equivalent to the event-wise definition,  $\widetilde{\Pr}[\mathcal{Q}(S) \in E] \in [0, 1]$  for all  $E \subseteq \{0, 1\}^S$ , and hence every  $\ell$ -local quasi-distribution  $\mathcal{Q}$  is  $(\ell, 0)$ -local.

Below we discuss the following questions. Given an approximately local quasi-distribution  $\mathcal{Q}$ , can we find a local quasi-distribution  $\mathcal{Q}'$  close to it? Moreover, can we ensure that  $\mathcal{Q}'$  “looks like”  $\mathcal{Q}$ ? We show that if  $\mathcal{Q}$  is  $(\ell, \varepsilon)$ -local and is supported over a set  $G$  of functions that is nice in some precise way, then there is an  $\ell$ -local  $\mathcal{Q}'$  over  $G$  that is close to  $\mathcal{Q}$ . The proof idea is similar to that of “smoothing” almost-feasible solutions to Sherali–Adams relaxations into feasible ones [42].

► **Lemma 7.8.** *Let  $D$  be a finite domain,  $\ell \in \{1, \dots, |D|\}$  be an integer, and  $\delta > 0$ ,  $\varepsilon \geq 0$  be reals. Let  $G \subseteq U_D$  be a set of functions  $f: D \rightarrow \{0, 1\}$  such that for all subsets  $S \subseteq D$  of size  $|S| \leq \ell$  and for all strings  $\vec{b} \in \{0, 1\}^S$  it holds that  $\Pr_{f \leftarrow G}[f(S) = \vec{b}] \in \{0\} \cup [\delta, 1]$ , where  $f$  is sampled uniformly at random from  $G$ . If  $\mathcal{Q}$  is a  $(\ell, \varepsilon)$ -local quasi-distribution over  $G$ , then there exists an  $\ell$ -local quasi-distribution  $\mathcal{Q}'$  over  $G$  such that  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}') \leq (1 + \varepsilon - \delta) \cdot \frac{\varepsilon}{\varepsilon + \delta}$ .*

We highlight two notable special cases for the domain  $D = \{0, 1\}^n$ . If  $G = U_{\{0, 1\}^n}$  (the set of all functions), then  $\Pr_{f \leftarrow G}[f(S) = \vec{b}] = 2^{-\ell}$ . Also, if  $G = \text{LIN}$  (the set of all linear functions), then for every subset  $S \subseteq \{0, 1\}^n$  of size at most  $\ell$  and every string  $\vec{b} \in \{0, 1\}^S$  it holds that  $\Pr_{f \leftarrow G}[f(S) = \vec{b}] = 0$  or  $\Pr_{f \leftarrow G}[f(S) = \vec{b}] = 2^{-\dim(\text{span}(S))} \geq 2^{-|S|} \geq 2^{-\ell}$ . These two cases yield the following corollary.

► **Corollary 7.9.** *If  $\mathcal{Q}$  is a  $(\ell, \varepsilon)$ -local quasi-distribution over  $U_{\{0, 1\}^n}$  (resp.,  $\text{LIN}$ ), then there is an  $\ell$ -local quasi-distribution  $\mathcal{Q}'$  over  $U_{\{0, 1\}^n}$  (resp.,  $\text{LIN}$ ) such that  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}') \leq \frac{1 + \varepsilon - 2^{-\ell}}{1 + 2^\ell \varepsilon} \cdot 2^\ell \varepsilon < 2^\ell \varepsilon$ .*

**Proof.** The hypothesis of Lemma 7.8 holds with  $\delta = 2^{-\ell}$ . So there exists an  $\ell$ -local quasi-distribution  $\mathcal{Q}'$  over  $U_{\{0, 1\}^n}$  (resp.,  $\text{LIN}$ ) such that  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}') \leq \varepsilon \cdot \frac{1 + \varepsilon - \delta}{\varepsilon + \delta} = \frac{1 + \varepsilon - 2^{-\ell}}{\varepsilon + 2^{-\ell}} \cdot \varepsilon = \frac{1 + \varepsilon - 2^{-\ell}}{1 + 2^\ell \varepsilon} \cdot 2^\ell \varepsilon$ . Clearly the fraction is smaller than 1, and so the entire expression is at most  $2^\ell \varepsilon$ . ◀

We now prove the lemma.

**Proof of Lemma 7.8.** Let  $\mathcal{U}_G$  be the uniform distribution over all functions in  $G$ . For  $\varepsilon' := \frac{\varepsilon}{\varepsilon + \delta}$ , define the quasi-distribution  $\mathcal{Q}' := (1 - \varepsilon')\mathcal{Q} + \varepsilon'\mathcal{U}_G$ . Namely, if the vector of quasi-probabilities of  $\mathcal{Q}$  is  $(q_f)_{f \in G}$ , then the the vector of quasi-probabilities of  $\mathcal{Q}'$  is  $(q'_f)_{f \in G}$  where  $q'_f := (1 - \varepsilon') \cdot q_f + \varepsilon'/|G|$ .

First, we show that  $\mathcal{Q}'$  is an  $\ell$ -local quasi-distribution. That is, for all subsets  $S \subseteq D$  of size at most  $\ell$  and for every  $\vec{b} \in \{0, 1\}^S$  it holds that  $\widetilde{\Pr}[\mathcal{Q}'(S) = \vec{b}] \geq 0$ . Fix such an  $S$  and  $\vec{b}$ . If  $\Pr_{f \in G}[f(S) = \vec{b}] = 0$ , then there is no  $f \in G$  such that  $f(S) = \vec{b}$ , and hence  $\widetilde{\Pr}[\mathcal{Q}'(S) = \vec{b}] = 0$ . Otherwise,  $\Pr_{f \in G}[f(S) = \vec{b}] \geq \delta$ , and hence,

$$\begin{aligned} \widetilde{\Pr}[\mathcal{Q}'(S) = \vec{b}] &= \sum_{f \in G: f(S) = \vec{b}} q'_f \\ &= \left( \sum_{f \in G: f(S) = \vec{b}} (1 - \varepsilon') q_f \right) + \varepsilon' \Pr_{f \in G}[f(S) = \vec{b}] \\ &\geq \left( \sum_{f \in G: f(S) = \vec{b}} (1 - \varepsilon') q_f \right) + \varepsilon' \cdot \delta \\ &\geq -\varepsilon(1 - \varepsilon') + \varepsilon' \cdot \delta \\ &= -\varepsilon \left( \frac{\delta}{\varepsilon + \delta} \right) + \frac{\varepsilon}{\varepsilon + \delta} \delta = 0. \end{aligned}$$

## 17:24 Testing Linearity against Non-Signaling Strategies

Second, we show that  $\mathcal{Q}$  and  $\mathcal{Q}'$  are close in the sense that  $\Delta_\ell(\mathcal{Q}, \mathcal{Q}') \leq 2\varepsilon' \cdot (1 + \varepsilon - \delta)$  (see Definition 7.5). Fix a subset  $S \subseteq D$  of size at most  $\ell$ , and let  $E \subseteq \{0, 1\}^S$ . Then

$$\begin{aligned} \left| \widetilde{\Pr}[\mathcal{Q}(S) \in E] - \widetilde{\Pr}[\mathcal{Q}'(S) \in E] \right| &= \left| \left( \sum_{f \in G: f(S) \in E} \varepsilon' q_f \right) - \varepsilon' \Pr_{f \in G}[f(S) \in E] \right| \\ &= \left| \varepsilon' \widetilde{\Pr}[\mathcal{Q}(S) \in E] - \varepsilon' \Pr_{f \in G}[f(S) \in E] \right| \\ &\leq \varepsilon' (1 + \varepsilon - \delta) , \end{aligned}$$

as required.  $\blacktriangleleft$

### 8 Equivalence of non-signaling functions and local quasi-distributions

We establish an equivalence between non-signaling functions and local quasi-distributions. First, we show that every local quasi-distribution induces a non-signaling function. Second, we show that the converse is also true, namely, that every non-signaling function can be described by a local quasi-distribution. In fact, the set of quasi-distributions describing it is a real affine subspace.

► **Theorem 8.1** (from local quasi-distributions to non-signaling functions). *Let  $D$  be a finite domain. For every  $\ell$ -local quasi-distribution  $\mathcal{Q}$  over functions  $f: D \rightarrow \{0, 1\}$  there exists an  $\ell$ -non-signaling function  $\mathcal{F}$  over  $D$  such that for every subset  $S \subseteq D$  of size  $|S| \leq \ell$  and string  $\vec{b} \in \{0, 1\}^S$ ,  $\Pr[\mathcal{F}(S) = \vec{b}] = \widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}]$ .*

**Proof.** For every subset  $S \subseteq D$  of size  $|S| \leq \ell$ , define  $\mathcal{F}_S$  to be the distribution over functions  $f: S \rightarrow \{0, 1\}$  where  $\Pr[\mathcal{F}_S \text{ outputs } f] := \widetilde{\Pr}[\mathcal{Q}(S) = f(S)]$ . Note that  $\mathcal{F}_S$  is indeed a distribution because  $\mathcal{Q}$  is  $\ell$ -local, so the relevant probabilities are in  $[0, 1]$  and sum to 1. The definition immediately implies that  $\Pr[\mathcal{F}(S) = \vec{b}] = \widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}]$  for every string  $\vec{b} \in \{0, 1\}^S$ . We are left to argue that  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq D, |S| \leq \ell}$  is  $\ell$ -non-signaling.

Consider any two distinct subsets  $S, T \subseteq D$  of size at most  $\ell$ , and any string  $\vec{b} \in \{0, 1\}^{S \cap T}$ . Let  $U_S$  denote the set of functions from  $S \rightarrow \{0, 1\}$ . We have that

$$\begin{aligned} \Pr_{f \leftarrow \mathcal{F}_S}[f(S \cap T) = \vec{b}] &= \sum_{\substack{f \in U_S \text{ s.t.} \\ f(S \cap T) = \vec{b}}} \Pr[\mathcal{F}_S \text{ outputs } f] = \sum_{\substack{f \in U_S \text{ s.t.} \\ f(S \cap T) = \vec{b}}} \widetilde{\Pr}[\mathcal{Q}(S) = f(S)] \\ &= \sum_{\substack{f \in U_S \text{ s.t.} \\ f(S \cap T) = \vec{b}}} \sum_{\substack{g \in U \text{ s.t.} \\ g(S) = f(S)}} q_g = \sum_{\substack{g \in U \text{ s.t.} \\ g(S \cap T) = \vec{b}}} q_g = \widetilde{\Pr}[\mathcal{Q}(S \cap T) = \vec{b}] \end{aligned}$$

Similarly, we have that  $\Pr_{f \leftarrow \mathcal{F}_T}[f(S \cap T) = \vec{b}] = \widetilde{\Pr}[\mathcal{Q}(S \cap T) = \vec{b}]$ , and we conclude that  $\Pr_{f \leftarrow \mathcal{F}_S}[f(S \cap T) = \vec{b}] = \Pr_{f \leftarrow \mathcal{F}_T}[f(S \cap T) = \vec{b}]$ . Since  $S, T$  were arbitrary,  $\mathcal{F}$  is  $\ell$ -non-signaling.  $\blacktriangleleft$

We now show that every  $k$ -non-signaling function  $\mathcal{F}$  arises from a  $k$ -local quasi-distribution  $\mathcal{Q}$ . Moreover, the set of such quasi-distributions is an affine subspace of co-dimension  $\binom{N}{\leq k}$  in  $\mathbb{R}^{2^N}$ , where  $N = |D|$  and  $\binom{N}{\leq k} := \sum_{i=0}^k \binom{N}{i}$ . This converse is the interesting direction of the equivalence.

► **Theorem 8.2** (from non-signaling functions to local quasi-distributions). *For every  $k$ -non-signaling function  $\mathcal{F} = \{\mathcal{F}_S\}_{S \subseteq D, |S| \leq k}$  over a finite domain  $D$  of size  $N$  there exists a  $k$ -local*

quasi-distribution  $\mathcal{Q}$  over functions  $f: D \rightarrow \{0, 1\}$  that describes  $\mathcal{F}$  (for every subset  $S \subseteq D$  of size  $|S| \leq k$  and string  $\vec{b} \in \{0, 1\}^S$  it holds that  $\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] = \Pr[\mathcal{F}(S) = \vec{b}]$ ).

Moreover, the set of such quasi-distributions (viewed as vectors in  $\mathbb{R}^{2^N}$ ) is the affine subspace of co-dimension  $\binom{N}{\leq k}$  given by  $\mathcal{Q}_0 + \text{span}\{\chi_T : T \subseteq D, |T| > k\}$ , where  $\mathcal{Q}_0$  is any solution and  $\chi_T: \{0, 1\}^D \rightarrow \mathbb{R}$  is defined as  $\chi_T(f) := (-1)^{\sum_{x \in T} f(x)}$ .

**Proof.** We break the proof into three parts. First, we find one quasi-distribution that matches  $\mathcal{F}$ . Then, we find an affine space of such quasi-distributions. Finally, we prove that this affine space contains all possible solutions.

**Finding one solution.** We construct a  $k$ -local quasi-distribution  $\mathcal{Q}$  that behaves like  $\mathcal{F}$  on all sets of size at most  $k$ . Consider  $q(\cdot) := \sum_{T: |T| \leq k} \widehat{q}(T) \chi_T(\cdot)$ , where  $\widehat{q}(T)$  is defined as follows.

$$\widehat{q}(T) := \begin{cases} \frac{1}{2^N} & \text{if } T = \emptyset \\ \frac{2}{2^N} (\Pr[\sum_{x \in T} \mathcal{F}(x) = 0] - \frac{1}{2}) & \text{if } 1 \leq |T| \leq k \\ 0 & \text{if } |T| > k \end{cases} .$$

Note that  $\mathcal{Q}$  is a quasi-distribution because  $\sum_f q_f = 2^N \langle q, \chi_\emptyset \rangle = 2^N \widehat{q}(\emptyset) = 1$ . Now, for any subset  $S = \langle x_1, \dots, x_s \rangle$  with  $|S| \leq k$ ,

$$\begin{aligned} \sum_{f: \sum_{x \in S} f(x) = 0} q_f &= \sum_f q_f (-1)^{\sum_{x \in S} f(x)} + \sum_{f: \sum_{x \in S} f(x) = 1} q_f \\ &= 2^N \langle q, \chi_S \rangle + \left( 1 - \sum_{f: \sum_{x \in S} f(x) = 0} q_f \right) \\ &= 2^N \frac{1}{2^{N-1}} \left( \Pr \left[ \sum_{x \in S} \mathcal{F}(x) = 0 \right] - \frac{1}{2} \right) + \left( 1 - \sum_{f: \sum_{x \in S} f(x) = 0} q_f \right) \\ &= 2 \Pr \left[ \sum_{x \in S} \mathcal{F}(x) = 0 \right] - \sum_{f: \sum_{x \in S} f(x) = 0} q_f , \end{aligned}$$

which implies that

$$\widetilde{\Pr} \left[ \sum_{x \in S} \mathcal{Q}(x) = 0 \right] = \sum_{f: \sum_{x \in S} f(x) = 0} q_f = \Pr \left[ \sum_{x \in S} \mathcal{F}(x) = 0 \right] .$$

Therefore,

$$\widetilde{\Pr} \left[ \sum_{x \in S} \mathcal{Q}(x) = 1 \right] = 1 - \widetilde{\Pr} \left[ \sum_{x \in S} \mathcal{Q}(x) = 0 \right] = 1 - \Pr \left[ \sum_{x \in S} \mathcal{F}(x) = 0 \right] = \Pr \left[ \sum_{x \in S} \mathcal{F}(x) = 1 \right]$$

**17:26 Testing Linearity against Non-Signaling Strategies**

Thus, by Corollary 5.2 for any choice of bits  $b_1, \dots, b_s \in \{0, 1\}$  we have

$$\begin{aligned} \Pr[\mathcal{F}(x_1) = b_1, \dots, \mathcal{F}(x_s) = b_s] &= -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \Pr \left[ \sum_{i \in T} \mathcal{F}(x_i) = \sum_{i \in T} b_i \right] \\ &= -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \widetilde{\Pr} \left[ \sum_{i \in T} \mathcal{Q}(x_i) = \sum_{i \in T} b_i \right] \\ &= \widetilde{\Pr}[\mathcal{Q}(x_1) = b_1, \dots, \mathcal{Q}(x_s) = b_s] . \end{aligned}$$

This shows that  $\mathcal{Q}$  behaves like  $\mathcal{F}$  on all sets of size at most  $k$ .

**Finding more solutions.** We argue that Fourier coefficients for subsets  $T$  of size greater than  $k$  do not affect the induced non-signaling function. Indeed, fix a subset  $T \subseteq D$  of size greater than  $k$ , and let  $\mathcal{Q}' = (q'_f)_f$  be the quasi-distribution obtained from  $\mathcal{Q} = (q_f)_f$  by defining its weights as  $q'_f := q_f + c\chi_T(f)$ . Observe that for every ordered subset  $S = \langle x_1, \dots, x_s \rangle$  with  $s \leq k$  and bits  $b_1, \dots, b_s$  it holds that

$$\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] = \sum_{f: f(S) = \vec{b}} q_f = \sum_{f: f(S) = \vec{b}} (q_f + c\chi_T(f)) = \widetilde{\Pr}[\mathcal{Q}'(S) = \vec{b}] .$$

To see that the middle equality holds, observe that there exists  $y \in T \setminus S$ , and thus

$$\begin{aligned} \sum_{f: f(S) = \vec{b}} \chi_T(f) &= \sum_{f(S) = \vec{b}} (-1)^{\sum_{x \in T} f(x)} \\ &= \sum_{\substack{f: f(S) = \vec{b} \\ f(y) = 0}} (-1)^{\sum_{x \in T \setminus \{y\}} f(x)} - \sum_{\substack{f: f(S) = \vec{b} \\ f(y) = 1}} (-1)^{\sum_{x \in T \setminus \{y\}} f(x)} = 0 . \end{aligned}$$

Therefore,  $\mathcal{Q}'$  matches  $\mathcal{Q}$  (and thus also  $\mathcal{F}$ ) on all sets of size at most  $k$ . Since this holds for every  $T$  with  $|T| > k$ , we see that *every*  $q'$  in  $q + \text{span}\{\chi_T : T \subseteq D, |T| > k\}$  also matches  $\mathcal{F}$  on all subsets of size at most  $k$ .

**We found all solutions.** Observe that if  $\mathcal{Q}$  is a quasi-distribution, then for every subset  $T \subseteq D$  with  $1 \leq |T| \leq k$  it holds that

$$\begin{aligned} \widehat{q}(T) &= \frac{1}{2^N} \sum_f q_f (-1)^{\sum_{x \in T} f(x)} \\ &= \frac{1}{2^N} \left( \sum_{f: \sum_{x \in T} f(x) = 0} q_f - \sum_{f: \sum_{x \in T} f(x) = 1} q_f \right) \\ &= \frac{1}{2^N} \left( \widetilde{\Pr} \left[ \sum_{x \in T} \mathcal{Q}(x) = 0 \right] - \widetilde{\Pr} \left[ \sum_{x \in T} \mathcal{Q}(x) = 1 \right] \right) \\ &= \frac{1}{2^{N-1}} \left( \widetilde{\Pr} \left[ \sum_{x \in T} \mathcal{Q}(x) = 0 \right] - \frac{1}{2} \right) . \end{aligned}$$

If  $\mathcal{Q}$  and  $\mathcal{F}$  match on all input sets of size at most  $k$ , then they match on all parity events of size at most  $k$ , and so  $\widehat{q}(T) = \frac{1}{2^{N-1}} \left( \widetilde{\Pr}[\sum_{x \in T} \mathcal{F}(x) = 0] - \frac{1}{2} \right)$ . Since  $\widehat{q}(\emptyset) = \frac{1}{2^N} \sum_f q_f = \frac{1}{2^N}$ ,

we see that exactly  $\binom{N}{\leq k}$  Fourier coefficients are determined. Thus, the set of all solutions is contained in  $q + \text{span}\{\chi_T : T \subseteq D, |T| > k\}$ .

On the other hand, we have already shown that the affine space  $q + \text{span}\{\chi_T : T \subseteq D, |T| > k\}$  contains only quasi-distributions that match  $\mathcal{F}$  on all sets of size at most  $k$ . Thus, the affine space of *all* quasi-distributions that match  $\mathcal{F}$  is precisely  $q + \text{span}\{\chi_T : T \subseteq D, |T| > k\}$ . ◀

## 9 Quasi-distributions over functions with small support

We show that *every*  $k$ -non-signaling function can be expressed as a quasi-distribution over functions with small support, namely, functions that evaluate to 1 for at most  $k$  inputs. For linearity testing, this implies that restricting a quasi-distribution to functions that are  $\varepsilon$ -close to linear is an empty condition, because all  $k$ -non-signaling functions can be expressed by such quasi-distributions for  $\varepsilon = \frac{k}{2^n}$ , regardless of whether they pass the linearity test with high or low probability.

For a finite domain  $D$ , we denote by  $U_D$  the set of all boolean functions  $f : D \rightarrow \{0, 1\}$  and, for  $k \leq |D|$ , denote by  $U_{\leq k}$  the subset of  $U_D$  of all functions that evaluate to 1 for at most  $k$  values in  $D$ . We show that every  $k$ -non-signaling function  $\mathcal{F}$  is described by a quasi-distribution over  $U_{\leq k}$ .

► **Theorem 9.1.** *Let  $D$  be a finite domain. For every  $k$ -non-signaling function  $\mathcal{F}$  over  $D$  there exists a  $k$ -local quasi-distribution  $\mathcal{Q}$  over  $D$  supported on  $U_{\leq k}$  such that for every subset  $S \subseteq D$  of size  $|S| \leq k$  and string  $\vec{b} \in \{0, 1\}^S$  it holds that  $\Pr[\mathcal{Q}(S) = \vec{b}] = \Pr[\mathcal{F}(S) = \vec{b}]$ .*

The proof of Theorem 9.1 relies on the following claim.

► **Claim 9.2.** *Let  $\mathcal{F}$  be a  $k$ -non-signaling function over a domain  $D$ , and let  $\mathcal{Q}$  be a quasi-distribution over functions  $f : D \rightarrow \{0, 1\}$ . If for every subset  $S \subseteq D$  with  $1 \leq |S| \leq k$  it holds that  $\Pr[\mathcal{Q}(S) = 1^{|S|}] = \Pr[\mathcal{F}(S) = 1^{|S|}]$  then for every subset  $S \subseteq D$  with  $|S| \leq k$  and string  $\vec{b} \in \{0, 1\}^S$  it holds that  $\Pr[\mathcal{Q}(S) = \vec{b}] = \Pr[\mathcal{F}(S) = \vec{b}]$ .*

We first prove Theorem 9.1 using the claim, and then prove the claim.

**Proof of Theorem 9.1.** By Claim 9.2 it suffices to prove that the following linear system of equations, in the variables  $\{q_f\}_{f \in U_{\leq k}}$ , has a solution:

$$\left\{ \begin{array}{l} \sum_{f \in U_{\leq k}} q_f = 1 \\ \sum_{\substack{f \in U_{\leq k} \text{ s.t.} \\ f(S) = 1^{|S|}}} q_f = \Pr[\mathcal{F}(S) = 1^{|S|}] \quad \forall S \subseteq D \text{ with } 1 \leq |S| \leq k \end{array} \right\} .$$

We do so by iteratively assigning values to the variables  $\{q_f\}_{f \in U_{\leq k}}$ , by considering all functions with support size  $k$ , then with support size  $k - 1$ , and so on. At a high level, we shall use the fact that this system of linear equations corresponds to an upper triangular matrix (once variables are ordered according to support sizes), and thus can be solved via back substitution.

First, consider any  $f \in U_{\leq k}$  such that  $|\text{supp}(f)| = k$ , and let  $S := \text{supp}(f)$ . Since  $f$  is the *only* function in  $U_{\leq k}$  whose support equals  $S$ , we must assign

$$q_f := \Pr[\mathcal{F}(\text{supp}(f)) = 1^k] .$$

## 17:28 Testing Linearity against Non-Signaling Strategies

Next, we use induction on  $s = k - 1, \dots, 1$  in decreasing order. Consider any  $f \in U_{\leq k}$  such that  $|\text{supp}(f)| = s$ , and set

$$q_f := \Pr[\mathcal{F}(\text{supp}(f)) = 1^s] - \left( \sum_{\substack{f' \in U_{\leq k} \text{ s.t.} \\ \text{supp}(f') \supsetneq \text{supp}(f)}} q_{f'} \right).$$

The above is well-defined since we first define  $q_f$  for all functions with larger support. Moreover, any choice of  $q_{f'}$  for functions  $f'$  whose support does not contain  $\text{supp}(f)$  does *not* affect the quasi-probability  $\widetilde{\Pr}[\mathcal{Q}(\text{supp}(f)) = 1^s]$ , and so we may think of this assignment as  $q_f$  satisfying the constraint  $\widetilde{\Pr}[\mathcal{Q}(\text{supp}(f)) = 1^s] = \Pr[\mathcal{F}(\text{supp}(f)) = 1^s]$ .

Finally, if  $f$  is the all-zero function we define

$$q_f := 1 - \sum_{f' \neq f} q_{f'},$$

so that  $\sum_{f \in U_{\leq k}} q_f = 1$ . It is clear from the construction that the assignments to the variables  $\{q_f\}_{f \in U_{\leq k}}$  above satisfy the necessary linear constraints, as desired.  $\blacktriangleleft$

**Proof of Claim 9.2.** Fix any subset  $S \subseteq D$  with  $|S| \leq k$  and string  $\vec{b} \in \{0, 1\}^S$ . We prove that  $\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] = \Pr[\mathcal{F}(S) = \vec{b}]$ , via induction on  $|Z|$  where  $Z := \{i \in S : b_i = 0\}$ .

If  $|Z| = 0$ , then  $\widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] = \Pr[\mathcal{F}(S) = \vec{b}]$  holds by the assumption of the claim.

Now suppose that  $|Z| > 0$ , and let  $i^* \in S$  be any coordinate such that  $b_{i^*} = 0$ . Let  $\vec{b}_{-i^*} \in \{0, 1\}^S$  be the vector obtained from  $\vec{b}$  by *flipping* the  $i^*$ -th coordinate to 1, and let  $\vec{b}_{-i^*} \in \{0, 1\}^{S \setminus \{i^*\}}$  be the vector obtained from  $\vec{b}$  by *removing* the  $i^*$ -th coordinate. We deduce that

$$\begin{aligned} \Pr[\mathcal{F}(S) = \vec{b}] &= \Pr[\mathcal{F}(S \setminus \{i^*\}) = \vec{b}_{-i^*}] - \Pr[\mathcal{F}(S) = \vec{b}_{-i^*}], \text{ and} \\ \widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}] &= \widetilde{\Pr}[\mathcal{Q}(S \setminus \{i^*\}) = \vec{b}_{-i^*}] - \widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}_{-i^*}]. \end{aligned}$$

The inductive hypothesis tells us that  $\Pr[\mathcal{F}(S \setminus \{i^*\}) = \vec{b}_{-i^*}] = \widetilde{\Pr}[\mathcal{Q}(S \setminus \{i^*\}) = \vec{b}_{-i^*}]$  and  $\Pr[\mathcal{F}(S) = \vec{b}_{-i^*}] = \widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}_{-i^*}]$ , from which we obtain that  $\Pr[\mathcal{F}(S) = \vec{b}] = \widetilde{\Pr}[\mathcal{Q}(S) = \vec{b}]$ , as claimed.  $\blacktriangleleft$

## 10 Exact local characterization of linear functions

We prove our results about non-signaling functions that always pass the linearity test. The theorem below states that the test passes with probability 1 if and only if the non-signaling function on sets of size at most  $k - 1$  can be described by a  $(k - 1)$ -local quasi-distribution over linear functions.

► **Theorem 10.1** (exact local characterization). *Let  $\mathcal{F}$  be a  $k$ -non-signaling function with  $k \geq 4$ . The following statements are equivalent.*

1. *The linearity test always accepts:  $\Pr_{x,y,\mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] = 1$ .*
2. *For all  $x, y \in \{0, 1\}^n$  it holds that  $\Pr_{\mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] = 1$ .*
3. *There exists a unique  $(k - 1)$ -local quasi-distribution  $\mathcal{L}$  over LIN such that for every set  $S \subseteq \{0, 1\}^n$  of size  $|S| \leq k - 1$  and vector  $\vec{b} \in \{0, 1\}^S$  it holds that  $\Pr[\mathcal{F}(S) = \vec{b}] = \widetilde{\Pr}[\mathcal{L}(S) = \vec{b}]$ .*

We comment on several aspects of the theorem.



- **The case of large  $k$ .** If  $k \geq n + 1$ , then  $\mathcal{L}$  in Item 3 is in fact a (standard) distribution over linear functions. *Explanation.* Let  $\ell_\alpha$  be the weight assigned to the linear function  $\langle \alpha, \cdot \rangle$  by  $\mathcal{L}$ . Since  $\mathcal{L}$  matches  $\mathcal{F}$  on sets of size  $n$ , we see that each  $\ell_\alpha$  is non-negative:

$$\ell_\alpha = \sum_{\alpha': \langle \alpha', e_i \rangle = \alpha_i \ 1 \leq i \leq n} \ell_{\alpha'} = \Pr[\mathcal{F}(e_1) = \alpha_1, \dots, \mathcal{F}(e_n) = \alpha_n] \geq 0 .$$

- **Agreement on  $k - 1$  layers.** The fact that  $|S| < k$  in Item 3 is necessary, because we can construct a  $k$ -non-signaling function  $\mathcal{F}$  where  $\Pr[\mathcal{F}(S) = \vec{b}] \neq \widetilde{\Pr}[\mathcal{L}(S) = \vec{b}]$  when  $|S| = k$ .

*Explanation.* Let  $S_1$  be the set of  $S$  such that  $|S| < k$  or  $S$  is linearly dependent, and  $S_2$  be the set of  $S$  such that  $|S| = k$  and  $S$  is linearly independent. The non-signaling function  $\mathcal{F}$  that answers according to a uniformly random linear function on all sets in  $S_1$  and answers with uniformly random bits that sum to 0 on all sets in  $S_2$  is  $k$ -non-signaling. Furthermore, the corresponding unique  $\mathcal{L}$  is the uniform distribution over linear functions, and so  $\Pr[\mathcal{F}(S) = \vec{b}] \neq \widetilde{\Pr}[\mathcal{L}(S) = \vec{b}]$  when  $S \in S_2$ .

- **The case of  $k = 3$ .** In the theorem it is necessary to have  $k \geq 4$ . This is because for  $k = 3$  it is not true that Item 3 always implies Item 2: it is possible for Item 3 to hold while the linearity test passes with probability 0.

*Explanation.* Let  $\mathcal{L}$  be a uniform distribution over linear functions, and let  $\mathcal{F}$  be a 3-non-signaling function that agrees with  $\mathcal{L}$  on all query sets of size 2. For every subset  $\{x, y, z\} \subseteq \{0, 1\}^n \setminus \{0^n\}$  of size 3, the distribution of  $\mathcal{F}$  is uniform over the set of tuples  $\{(1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 1)\}$ . If the input set  $S$  contains  $0^n$ ,  $\mathcal{F}_S$  assigns  $0^n$  to 0 and answers the rest according to  $\mathcal{F}_{S \setminus \{0^n\}}$ . One can verify that  $\mathcal{F}$  is indeed a 3-non-signaling function. Clearly,  $\mathcal{F}$  satisfies Item 3, but passes the linearity test with probability 0, and hence does not satisfy Item 2.

**Proof that 1  $\iff$  2.** The acceptance probability of the test can be re-written as

$$\Pr_{x, y \leftarrow \{0, 1\}^n, \mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] = \frac{1}{2^{2n}} \sum_{x, y \in \{0, 1\}^n} \Pr_{\mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] ,$$

and note that each of the probabilities in the sum lies in  $[0, 1]$ . Therefore, the acceptance probability is 1 if and only if for *all*  $x, y \in \{0, 1\}^n$  it holds that  $\Pr_{\mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] = 1$ .  $\blacktriangleleft$

**Proof that 2  $\implies$  3.** We first argue that if  $\mathcal{F}$  behaves linearly on sets of the form  $\{x, y, x + y\}$ , then it behaves linearly on all sets of size less than  $k$ . Let  $s \in \{2, \dots, k - 1\}$ ,  $x_1, \dots, x_s \in \{0, 1\}^n$ , and  $b \in \{0, 1\}$ , and define  $S_i := \{\sum_{j=1}^i x_j, x_{i+1}, \dots, x_s\}$  for every  $i \in \{1, \dots, s\}$ . Note that  $|S_i \cup S_{i+1}| = s - i + 2 \leq s + 1 \leq k$ . Letting  $\text{add}(\cdot)$  be the addition function, the fact that the linearity test always passes implies that

$$\Pr \left[ \text{add}(\mathcal{F}(S_i)) = \text{add}(\mathcal{F}(S_{i+1})) \right] = \Pr \left[ \mathcal{F} \left( \sum_{j=1}^i x_j \right) + \mathcal{F}(x_{i+1}) = \mathcal{F} \left( \sum_{j=1}^{i+1} x_j \right) \right] = 1 .$$

**17:30 Testing Linearity against Non-Signaling Strategies**

This implies that  $\Pr[\mathcal{F}(\sum_{i=1}^s x_i) = b] = \Pr[\sum_{i=1}^s \mathcal{F}(x_i) = b]$ , via the following argument:

$$\begin{aligned} & \left| \Pr \left[ \sum_{i=1}^s \mathcal{F}(x_i) = b \right] - \Pr \left[ \mathcal{F} \left( \sum_{i=1}^s x_i \right) = b \right] \right| \\ &= |\Pr[\text{add}(\mathcal{F}(S_1)) = b] - \Pr[\text{add}(\mathcal{F}(S_s)) = b]| \\ &= \left| \sum_{i=1}^{s-1} \Pr[\text{add}(\mathcal{F}(S_i)) = b] - \Pr[\text{add}(\mathcal{F}(S_{i+1})) = b] \right| \\ &\leq \sum_{i=1}^{s-1} |\Pr[\text{add}(\mathcal{F}(S_i)) = b] - \Pr[\text{add}(\mathcal{F}(S_{i+1})) = b]| = 0 \quad , \end{aligned}$$

where the last equality is by Lemma 6.3, since  $|S_i \cup S_{i+1}| \leq k$  for every  $i$ . Note that  $s$  must be strictly less than  $k$  because  $|S_1 \cup S_2| = s + 1$ .

We now construct  $\mathcal{L}$ , and argue that it has the desired properties. Define  $(\ell_\alpha)_{\alpha \in \{0,1\}^n}$  to be the solution to the system of equations in Lemma 5.3 where  $c_\beta := \Pr[\mathcal{F}(\beta) = 0]$  for each  $\beta \in \{0,1\}^n$ , and let  $\mathcal{L}$  be the quasi-distribution over LIN that assigns weight  $\ell_\alpha$  to the linear function  $\langle \alpha, \cdot \rangle$ . That is,  $(\ell_\alpha)_{\alpha \in \{0,1\}^n}$  satisfy the linear equations

$$\sum_{\alpha: \langle \alpha, x \rangle = 0} \ell_\alpha = \Pr[\mathcal{F}(x) = 0]$$

for all  $x \in \{0,1\}^n$ . Note that  $\mathcal{L}$  is indeed a quasi-distribution, because  $\sum_\alpha \ell_\alpha = \Pr[\mathcal{F}(0^n) = 0] = \Pr_{x \leftarrow \{0,1\}^n} [\mathcal{F}(0^n) + \mathcal{F}(x) = \mathcal{F}(x)] = 1$  (as  $\mathcal{F}$  always passes the linearity test). We remark that every quasi-distribution supported on LIN is uniquely determined by its induced distributions on sets of size 1: a quasi-distribution is supported on LIN if and only if its distributions on sets of size 1 determine all of its Fourier coefficients (see full version for details).

Moreover, by definition of  $(\ell_\alpha)_{\alpha \in \{0,1\}^n}$ , for every  $x \in \{0,1\}^n$  it holds that

$$\Pr[\mathcal{F}(x) = 0] = \sum_{\alpha: \langle \alpha, x \rangle = 0} \ell_\alpha = \widetilde{\Pr}[\mathcal{L}(x) = 0] \quad ,$$

which implies that for every  $x \in \{0,1\}^n$  and bit  $b \in \{0,1\}$  it holds that  $\Pr[\mathcal{F}(x) = b] = \widetilde{\Pr}[\mathcal{L}(x) = b]$ . In other words,  $\mathcal{F}$  and  $\mathcal{L}$  match on sets of size 1. This allows us to derive the same conclusion for all sets of size less than  $k$ , as follows.

For every  $s \in \{1, \dots, k-1\}$ ,  $x_1, \dots, x_s \in \{0,1\}^n$ , and  $b_1, \dots, b_s \in \{0,1\}$ ,

$$\begin{aligned} & \Pr[\mathcal{F}(x_1) = b_1, \dots, \mathcal{F}(x_s) = b_s] \\ &= -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \Pr \left[ \sum_{i \in T} \mathcal{F}(x_i) = \sum_{i \in T} b_i \right] \quad (\text{by Corollary 5.2}) \\ &= -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \Pr \left[ \mathcal{F} \left( \sum_{i \in T} x_i \right) = \sum_{i \in T} b_i \right] \quad (\text{by linearity}) \\ &= -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \widetilde{\Pr} \left[ \mathcal{L} \left( \sum_{i \in T} x_i \right) = \sum_{i \in T} b_i \right] \\ &= -1 + \frac{1}{2^{s-1}} \sum_{T \subseteq [s]} \widetilde{\Pr} \left[ \sum_{i \in T} \mathcal{L}(x_i) = \sum_{i \in T} b_i \right] \quad (\text{since } \text{supp}(\mathcal{L}) \subseteq \text{LIN}) \\ &= \widetilde{\Pr}[\mathcal{L}(x_1) = b_1, \dots, \mathcal{L}(x_s) = b_s] \quad . \quad (\text{by Lemma 7.3}) \end{aligned}$$

Finally, since  $\mathcal{L}$  agrees with  $\mathcal{F}$  on all subsets of size less than  $k$ , the quasi-probabilities must be in  $[0, 1]$ , which means that  $\mathcal{L}$  is  $(k - 1)$ -local.  $\blacktriangleleft$

**Proof that 3  $\implies$  2.** Suppose that there exists a  $(k - 1)$ -local quasi-distribution  $\mathcal{L}$  over LIN such that for every  $s \in \{1, \dots, k - 1\}$ ,  $x_1, \dots, x_s \in \{0, 1\}^n$ , and  $b_1, \dots, b_s \in \{0, 1\}$  it holds that  $\Pr[\mathcal{F}(x_1) = b_1, \dots, \mathcal{F}(x_s) = b_s] = \widetilde{\Pr}[\mathcal{L}(x_1) = b_1, \dots, \mathcal{L}(x_s) = b_s]$ . For every  $\alpha \in \{0, 1\}^n$  denote by  $\ell_\alpha$  the weight assigned by  $\mathcal{L}$  to the linear function  $\langle \alpha, \cdot \rangle$ . For every  $x, y \in \{0, 1\}^n$  it holds that

$$\begin{aligned} \Pr[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] &= \sum_{b_1, b_2} \Pr[\mathcal{F}(x) = b_1, \mathcal{F}(y) = b_2, \mathcal{F}(x + y) = b_1 + b_2] \\ &= \sum_{b_1, b_2} \widetilde{\Pr}[\mathcal{L}(x) = b_1, \mathcal{L}(y) = b_2, \mathcal{L}(x + y) = b_1 + b_2] \\ &= \sum_{b_1, b_2} \sum_{\substack{\alpha: \langle \alpha, x \rangle = b_1 \\ \langle \alpha, y \rangle = b_2 \\ \langle \alpha, x+y \rangle = b_1 + b_2}} \ell_\alpha = \sum_{b_1, b_2} \sum_{\substack{\alpha: \langle \alpha, x \rangle = b_1 \\ \langle \alpha, y \rangle = b_2}} \ell_\alpha = \sum_{\alpha} \ell_\alpha = 1, \end{aligned}$$

as desired. Note that the equality on the second line uses the assumption that  $k \geq 4$ . This is because we need  $\mathcal{L}$  to match  $\mathcal{F}$  on sets of size 3, and we only know that  $\mathcal{L}$  matches  $\mathcal{F}$  on all sets of size at most  $k - 1$ .  $\blacktriangleleft$

## 11 Robust local characterization of linear functions

We prove our results about non-signaling functions that pass the linearity test with high probability. Given a  $k$ -non-signaling function  $\mathcal{F}$ , define its self-correction  $\hat{\mathcal{F}}$  as follows. On an input  $x \in \{0, 1\}^n$  we sample from  $\hat{\mathcal{F}}_{\{x\}}$  by drawing a uniform  $w \in \{0, 1\}^n$ , sampling a function  $f$  from  $\mathcal{F}_{\{x+w, w\}}$ , and outputting  $f(x + w) + f(w)$ . We generalize this correction to larger input sets in the natural way.

► **Definition 11.1.** Given a  $k$ -non-signaling function  $\mathcal{F}$ , define the *self-correction of  $\mathcal{F}$*  as follows. Given a set  $S = \{x_1, \dots, x_s\} \subseteq \{0, 1\}^n$ , we sample from  $\hat{\mathcal{F}}_{\{x_1, \dots, x_s\}}$  by drawing uniform and independent  $w_1, \dots, w_s \in \{0, 1\}^n$ , sampling a function  $f$  from the distribution  $\mathcal{F}_{\{x_1+w_1, \dots, x_s+w_s, w_1, \dots, w_s\}}$ , and outputting the function  $\hat{f}$  that maps each  $x_i$  to  $f(x_i + w_i) + f(w_i)$ . That is, for every subset  $S = \{x_1, \dots, x_s\} \subseteq \{0, 1\}^n$  of size at most  $\hat{k}$  and  $\vec{b} \in \{0, 1\}^S$ ,

$$\Pr[\hat{\mathcal{F}}(S) = \vec{b}] := \Pr_{\substack{w_1, \dots, w_s \leftarrow \{0, 1\}^n \\ \mathcal{F}}} \begin{bmatrix} \mathcal{F}(x_1 + w_1) + \mathcal{F}(w_1) = b_1 \\ \vdots \\ \mathcal{F}(x_s + w_s) + \mathcal{F}(w_s) = b_s \end{bmatrix}.$$

$\hat{\mathcal{F}}$  is a  $\hat{k}$ -non-signaling function for  $\hat{k} \leq \lfloor k/2 \rfloor$ . This follows immediately from the fact that the  $w_i$ 's are random and independent, and the fact that  $\mathcal{F}$  is  $k$ -non-signaling.

The following theorem says that, if a  $k$ -non-signaling function  $\mathcal{F}$  passes the linearity test with high probability, then  $\hat{\mathcal{F}}$  is close to a quasi-distribution over linear functions.

► **Theorem 11.2 (robust local characterization).** *Let  $\mathcal{F}$  be a  $k$ -non-signaling function with  $k \geq 7$ , and let  $\hat{\mathcal{F}}$  be its ( $\hat{k}$ -non-signaling) self-correction. Each of the following statements implies the next one.*

1. *The linearity test accepts with probability  $1 - \varepsilon$ :  $\Pr_{x, y, \mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)] \geq 1 - \varepsilon$ .*

2. For all  $x, y \in \{0, 1\}^n$  it holds that  $\Pr_{\hat{\mathcal{F}}}[\hat{\mathcal{F}}(x) + \hat{\mathcal{F}}(y) = \hat{\mathcal{F}}(x + y)] \geq 1 - \hat{\varepsilon}$  with  $\hat{\varepsilon} := 4\varepsilon$ ; moreover, it also holds that  $\Pr_{\hat{\mathcal{F}}}[\hat{\mathcal{F}}(0^n) = 0] = 1$ .
3. There exists a quasi-distribution  $\mathcal{L}$  over LIN such that for every  $\ell \in \{1, \dots, \hat{k} - 1\}$  it holds that  $\mathcal{L}$  is  $(\ell, 2^{\ell/2}(\ell - 1)\hat{\varepsilon})$ -local and, for every subset  $S \subseteq \{0, 1\}^n$  of size at most  $\ell$  and every event  $E \subseteq \{0, 1\}^S$ ,  $|\Pr[\hat{\mathcal{F}}(S) \in E] - \Pr[\mathcal{L}(S) \in E]| \leq (|S| - 1) \cdot \|\widehat{\mathbf{1}}_E\|_1 \cdot \hat{\varepsilon} \leq (|S| - 1) \cdot \sqrt{|E|} \cdot \hat{\varepsilon}$ .
4. For every  $\ell \in \{1, \dots, \hat{k} - 1\}$ , there exists an  $\ell$ -local quasi-distribution  $\mathcal{L}'$  over LIN such that  $\Delta_\ell(\hat{\mathcal{F}}, \mathcal{L}') \leq (2^\ell + 1) \cdot 2^{\ell/2}(\ell - 1)\hat{\varepsilon}$ .

We highlight some of the differences of Theorem 11.2 ( $\varepsilon \geq 0$ ) from Theorem 10.1 ( $\varepsilon = 0$ ).

- In Item 2, we now need to use the self-correction  $\hat{\mathcal{F}}$  to ensure that  $\Pr_{\hat{\mathcal{F}}}[\hat{\mathcal{F}}(x + y) = \hat{\mathcal{F}}(x) + \hat{\mathcal{F}}(y)]$  is large for every  $x, y \in \{0, 1\}^n$ , as opposed to random  $x, y \in \{0, 1\}^n$ . This is necessary because otherwise it is possible for  $\Pr_{\mathcal{F}}[\mathcal{F}(x) + \mathcal{F}(y) = \mathcal{F}(x + y)]$  to be small for certain choices of  $x$  and  $y$ , and in this case a quasi-distribution supported only on linear functions has no hope of approximating  $\mathcal{F}$  on sets containing  $\{x, y, x + y\}$ .
- In Item 3, we choose  $\mathcal{L}$  to match  $\hat{\mathcal{F}}$  exactly on all sets of size 1, as before. However, since the linearity condition only holds approximately, this means that we only get approximate matching on larger input sets, and this approximation deteriorates as the sets get larger.
- Since  $\mathcal{L}$  only matches  $\hat{\mathcal{F}}$  approximately, it is only an approximately  $\ell$ -local distribution. Thus, we require the additional step of Item 4, where we correct  $\mathcal{L}$  to an exactly  $\ell$ -local distribution.

We now proceed to the proof of Theorem 11.2.

**Proof that 1  $\implies$  2.** Fix  $x, y \in \{0, 1\}^n$ . The definition of  $\hat{\mathcal{F}}$  implies that

$$\begin{aligned} & \Pr_{\hat{\mathcal{F}}}[\hat{\mathcal{F}}(x) + \hat{\mathcal{F}}(y) = \hat{\mathcal{F}}(x + y)] \\ &= \Pr_{\substack{w_x \\ w_y \\ w_{x+y} \\ \mathcal{F}}}[\mathcal{F}(x + w_x) + \mathcal{F}(w_x) + \mathcal{F}(y + w_y) + \mathcal{F}(w_y) = \mathcal{F}(x + y + w_{x+y}) + \mathcal{F}(w_{x+y})] . \end{aligned}$$

Define

$$\begin{aligned} S_1 &:= \{x + w_x, y + w_y, x + y + w_{x+y}, w_x, w_y, w_{x+y}\} , \\ S_2 &:= \{x + w_x + w_y, y + w_y, x + y + w_{x+y}, w_x, w_{x+y}\} , \\ S_3 &:= \{x + w_x + w_y, y + w_y + w_{x+y}, x + y + w_{x+y}, w_x\} , \\ S_4 &:= \{x + w_x + w_y, y + w_y + w_{x+y}, x + y + w_{x+y} + w_x\} . \end{aligned}$$

Observe that  $|S_i \cup S_{i+1}| \leq 7 \leq k$ . Letting  $\text{add}(\cdot)$  be the addition function, the linearity test passing with probability at least  $1 - \varepsilon$  implies that

$$\begin{aligned} & \Pr[\text{add}(\mathcal{F}(S_1)) = \text{add}(\mathcal{F}(S_2))] \\ &= \Pr_{\substack{w_x, w_y \\ \mathcal{F}}}[\mathcal{F}(x + w_x + w_y) = \mathcal{F}(x + w_x) + \mathcal{F}(w_y)] \geq 1 - \varepsilon , \\ & \Pr[\text{add}(\mathcal{F}(S_2)) = \text{add}(\mathcal{F}(S_3))] \\ &= \Pr_{\substack{w_y, w_{x+y} \\ \mathcal{F}}}[\mathcal{F}(y + w_y + w_{x+y}) = \mathcal{F}(y + w_y) + \mathcal{F}(w_{x+y})] \geq 1 - \varepsilon , \\ & \Pr[\text{add}(\mathcal{F}(S_3)) = \text{add}(\mathcal{F}(S_4))] \\ &= \Pr_{\substack{w_x, w_{x+y} \\ \mathcal{F}}}[\mathcal{F}(x + y + w_{x+y} + w_x) = \mathcal{F}(x + y + w_{x+y}) + \mathcal{F}(w_x)] \geq 1 - \varepsilon , \\ & \Pr[\text{add}(\mathcal{F}(S_4)) = 0] \\ &= \Pr_{\substack{w_x, w_y \\ w_{x+y} \\ \mathcal{F}}}[\mathcal{F}(x + w_x + w_y) + \mathcal{F}(y + w_y + w_{x+y}) = \mathcal{F}(x + y + w_{x+y} + w_x)] \geq 1 - \varepsilon . \end{aligned}$$

Therefore, by Lemma 6.3,

$$\begin{aligned} & |\Pr[\text{add}(\mathcal{F}(S_1)) = 0] - \Pr[\text{add}(\mathcal{F}(S_4)) = 0]| \\ & \leq \sum_{i=1}^3 |\Pr[\text{add}(\mathcal{F}(S_i)) = 0] - \Pr[\text{add}(\mathcal{F}(S_{i+1})) = 0]| \leq 3\varepsilon . \end{aligned}$$

Since  $\Pr[\text{add}(\mathcal{F}(S_4)) = 0] \geq 1 - \varepsilon$ , it follows that  $\Pr_{\hat{\mathcal{F}}}[\hat{\mathcal{F}}(x) + \hat{\mathcal{F}}(y) = \hat{\mathcal{F}}(x + y)] = \Pr[\text{add}(\mathcal{F}(S_1)) = 0] \geq 1 - 4\varepsilon = 1 - \hat{\varepsilon}$ , as claimed. Finally,  $\Pr[\hat{\mathcal{F}}(0^n) = 0] = \Pr_{w \in \{0,1\}^n}[\mathcal{F}(w + 0^n) + \mathcal{F}(w) = 0] = 1$ .  $\blacktriangleleft$

**Proof that 2  $\implies$  3.** This proof generalizes the proof that 2  $\implies$  3 in Theorem 10.1. We begin by arguing that  $\hat{\mathcal{F}}$  behaves almost linearly on sets of size at most  $\hat{k} - 1$ . Let  $s \in \{2, \dots, \hat{k} - 1\}$ ,  $x_1, \dots, x_s \in \{0, 1\}^n$ , and  $b \in \{0, 1\}$ , and define  $S_i := \{\sum_{j=1}^i x_j, x_{i+1}, \dots, x_s\}$  for every  $i \in \{1, \dots, s\}$ . Note that  $|S_i \cup S_{i+1}| = s - i + 2 \leq s + 1 \leq \hat{k}$ . Letting  $\text{add}(\cdot)$  be the addition function, the fact that the linearity test passes with probability at least  $1 - \hat{\varepsilon}$  implies that

$$\Pr \left[ \text{add}(\hat{\mathcal{F}}(S_i)) = \text{add}(\hat{\mathcal{F}}(S_{i+1})) \right] = \Pr \left[ \hat{\mathcal{F}} \left( \sum_{j=1}^i x_j \right) + \hat{\mathcal{F}}(x_{i+1}) = \hat{\mathcal{F}} \left( \sum_{j=1}^{i+1} x_j \right) \right] \geq 1 - \hat{\varepsilon} .$$

This implies that  $\left| \Pr[\hat{\mathcal{F}}(\sum_{i=1}^s x_i) = b] - \Pr[\sum_{i=1}^s \hat{\mathcal{F}}(x_i) = b] \right| \leq (s - 1)\hat{\varepsilon}$ , via the following argument:

$$\begin{aligned} & \left| \Pr \left[ \sum_{i=1}^s \hat{\mathcal{F}}(x_i) = b \right] - \Pr \left[ \hat{\mathcal{F}} \left( \sum_{i=1}^s x_i \right) = b \right] \right| \\ & = \left| \Pr[\text{add}(\hat{\mathcal{F}}(S_1)) = b] - \Pr[\text{add}(\hat{\mathcal{F}}(S_s)) = b] \right| \\ & = \left| \sum_{i=1}^{s-1} \Pr[\text{add}(\hat{\mathcal{F}}(S_i)) = b] - \Pr[\text{add}(\hat{\mathcal{F}}(S_{i+1})) = b] \right| \\ & \leq \sum_{i=1}^{s-1} \left| \Pr[\text{add}(\hat{\mathcal{F}}(S_i)) = b] - \Pr[\text{add}(\hat{\mathcal{F}}(S_{i+1})) = b] \right| \\ & \leq (s - 1)\hat{\varepsilon} . \end{aligned}$$

where the last inequality is by Lemma 6.3, since  $|S_i \cup S_{i+1}| \leq \hat{k}$  for every  $i$ . Note that  $s$  must be strictly less than  $\hat{k}$  because  $|S_1 \cup S_2| = s + 1$ .

We construct  $\mathcal{L}$  as before. Define  $(\ell_\alpha)_{\alpha \in \{0,1\}^n}$  to be the solution to the system of equations in Lemma 5.3 where  $c_\beta := \Pr[\hat{\mathcal{F}}(\beta) = 0]$  for each  $\beta \in \{0, 1\}^n$ , and let  $\mathcal{L}$  be the quasi-distribution over LIN that assigns weight  $\ell_\alpha$  to the linear function  $\langle \alpha, \cdot \rangle$ . Note that  $\mathcal{L}$  is indeed a quasi-distribution, because  $\sum_\alpha \ell_\alpha = \Pr[\hat{\mathcal{F}}(0^n) = 0] = 1$ .

Moreover, by definition of  $(\ell_\alpha)_{\alpha \in \{0,1\}^n}$ , for every  $x \in \{0, 1\}^n$  and  $b \in \{0, 1\}$  it holds that  $\Pr[\hat{\mathcal{F}}(x) = b] = \widetilde{\Pr}[\mathcal{L}(x) = b]$ . In other words,  $\mathcal{F}$  and  $\mathcal{L}$  match *exactly* on sets of size one. We now prove that  $\mathcal{F}$  and  $\mathcal{L}$  match *approximately* for sets of larger size (but still less than  $\hat{k}$ ) with a guarantee that degrades with the set size.

Fix  $s \in \{1, \dots, k - 1\}$ ,  $S = \{x_1, \dots, x_s\} \subseteq \{0, 1\}^n$ , and  $E \subseteq \{0, 1\}^S$ . We use Lemma 5.1

to get real numbers  $\{c_T\}_{T \subseteq [s]}$  that depend only on  $E$  such that

$$\begin{aligned}
 & \left| \Pr[\hat{\mathcal{F}}(S) \in E] - \widetilde{\Pr}[\mathcal{L}(S) \in E] \right| \\
 &= \left| \sum_{T \subseteq [s]} c_T \cdot \Pr \left[ \sum_{i \in T} \hat{\mathcal{F}}(x_i) = 0 \right] - \sum_{T \subseteq [s]} c_T \cdot \widetilde{\Pr} \left[ \sum_{i \in T} \mathcal{L}(x_i) = 0 \right] \right| \\
 &= \left| \sum_{T \subseteq [s]} c_T \left( \Pr \left[ \sum_{i \in T} \hat{\mathcal{F}}(x_i) = 0 \right] - \widetilde{\Pr} \left[ \sum_{i \in T} \mathcal{L}(x_i) = 0 \right] \right) \right| \\
 &= \left| \sum_{T \subseteq [s]} c_T \left( \Pr \left[ \sum_{i \in T} \hat{\mathcal{F}}(x_i) = 0 \right] - \widetilde{\Pr} \left[ \mathcal{L} \left( \sum_{i \in T} x_i \right) = 0 \right] \right) \right| \\
 &= \left| \sum_{T \subseteq [s]} c_T \left( \Pr \left[ \sum_{i \in T} \hat{\mathcal{F}}(x_i) = 0 \right] - \Pr \left[ \hat{\mathcal{F}} \left( \sum_{i \in T} x_i \right) = 0 \right] \right) \right| \\
 &\leq \sum_{T \subseteq [s]} |c_T| (|T| - 1) \hat{\varepsilon} \leq \hat{\varepsilon} \cdot (s - 1) \cdot \sum_{T \subseteq [s]} |c_T| \\
 &\leq \hat{\varepsilon} \cdot (s - 1) \|\widehat{\mathbf{1}}_E\|_1 \leq \hat{\varepsilon} \cdot (s - 1) \sqrt{|E|} .
 \end{aligned}$$

Since  $\hat{\mathcal{F}}$  defines probabilities in  $[0, 1]$ ,  $\mathcal{L}$  is  $(\ell, \varepsilon')$ -local with  $\varepsilon' = (\ell - 1)2^{\ell/2}\hat{\varepsilon}$  for any  $\ell < \hat{k}$ .  $\blacktriangleleft$

**Proof that 3  $\implies$  4.** Fix  $\ell \in \{1, \dots, \hat{k} - 1\}$ , and let  $\mathcal{L}$  be the  $(\ell, 2^{\ell/2}(\ell - 1)\hat{\varepsilon})$ -local quasi-distribution  $\mathcal{L}$  over LIN such that for every subset  $S \subseteq \{0, 1\}^n$  of size at most  $\ell$  and event  $E \subseteq \{0, 1\}^S$  it holds that

$$\left| \Pr[\hat{\mathcal{F}}(S) \in E] - \widetilde{\Pr}[\mathcal{L}(S) \in E] \right| \leq \sqrt{|E|}(|S| - 1)\hat{\varepsilon} \leq 2^{\ell/2}(\ell - 1)\hat{\varepsilon} .$$

Thus,  $\Delta_\ell(\hat{\mathcal{F}}, \mathcal{L}) \leq 2^{\ell/2}(\ell - 1)\hat{\varepsilon}$ . By Corollary 7.9, there is an  $\ell$ -local quasi-distribution  $\mathcal{L}'$  such that  $\Delta_\ell(\mathcal{L}, \mathcal{L}') \leq 2^\ell \cdot 2^{\ell/2}(\ell - 1)\hat{\varepsilon}$ . Therefore,

$$\Delta_\ell(\hat{\mathcal{F}}, \mathcal{L}') \leq \Delta_\ell(\hat{\mathcal{F}}, \mathcal{L}) + \Delta_\ell(\mathcal{L}, \mathcal{L}') \leq 2^{\ell/2}(\ell - 1)\hat{\varepsilon} + 2^\ell \cdot 2^{\ell/2}(\ell - 1)\hat{\varepsilon} = (2^\ell + 1) \cdot 2^{\ell/2}(\ell - 1)\hat{\varepsilon} . \blacktriangleleft$$

---

## References

- 1 William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, ICALP '00, pages 463–474, 2000.
- 2 Sabri W. Al-Safi and Anthony J. Short. Simulating all nonsignaling correlations via classical or quantum theory with negative probabilities. *Physical Review Letters*, 111:170403, 2013.
- 3 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in FOCS '92.
- 4 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in FOCS '92.
- 5 Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003. Preliminary version appeared in STOC '97.

- 6 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, 1991.
- 7 László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991. Preliminary version appeared in FOCS '90.
- 8 Jonathan Barrett. Information processing in generalized probabilistic theories. *Physical Review A*, 75:032304, 2007.
- 9 Jonathan Barrett, Lucien Hardy, and Adrian Kent. No signaling and quantum key distribution. *Physical Review Letters*, 95:010503, 2005.
- 10 Jonathan Barrett, Noah Linden, Serge Massar, Stefano Pironio, Sandu Popescu, and David Roberts. Nonlocal correlations as an information-theoretic resource. *Physical Review Letters*, 71:022101, 2005.
- 11 Jonathan Barrett and Stefano Pironio. Popescu–Rohrlich correlations as a unit of nonlocality. *Physical Review Letters*, 95:140401, 2005.
- 12 Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos A. Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, 1996.
- 13 Michael Ben-Or, Don Coppersmith, Mike Luby, and Ronitt Rubinfeld. Non-abelian homomorphism testing, and distributions close to their self-convolutions. *Random Structures and Algorithms*, 32(1):49–70, 2008.
- 14 Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of Reed-Muller codes. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 488–497, 2010.
- 15 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.
- 16 Gilles Brassard, Harry Buhrman, Noah Linden, André Allan Méthot, Alain Tapp, and Falk Unger. Limit on nonlocality in any world in which communication complexity is not trivial. *Physical Review Letters*, 96:250401, 2006.
- 17 Anne Broadbent and André Allan Méthot. On the power of non-local boxes. *Theoretical Computer Science*, 358(1):3–14, 2006.
- 18 Harry Buhrman, Matthias Christandl, Falk Unger, Stephanie Wehner, and Andreas Winter. Implications of superstrong non-locality for cryptography. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 462(2071):1919–1932, 2006.
- 19 Nicolas J. Cerf, Nicolas Gisin, Serge Massar, and Sandu Popescu. Simulating maximal quantum entanglement without communication. *Physical Review Letters*, 94:220403, 2005.
- 20 Rui Chao and Ben W. Reichardt. Test to separate quantum theory from non-signaling theories. arXiv quant-ph/1706.02008, 2017.
- 21 Roe David, Irit Dinur, Elazar Goldenberg, Guy Kindler, and Igor Shinkar. Direct sum testing. *SIAM Journal on Computing*, 46:1336–1369, 2017.
- 22 Paul A. M. Dirac. The physical interpretation of quantum mechanics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 180(980):1–40, 1942.
- 23 Cynthia Dwork, Michael Langberg, Moni Naor, Kobbi Nissim, and Omer Reingold. Succinct NP proofs and spooky interactions, December 2004. Available at [www.openu.ac.il/home/mikel/papers/spooky.ps](http://www.openu.ac.il/home/mikel/papers/spooky.ps).
- 24 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996. Preliminary version in FOCS '91.

- 25 Richard P. Feynman. Negative probability. In Basil J. Hiley and D. Peat, editors, *Quantum Implications: Essays in Honour of David Bohm*, pages 235–248. Law Book Co of Australasia, 1987.
- 26 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.
- 27 Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(1):141–172, 2009. Preliminary version appeared in STOC '07.
- 28 Tsuyoshi Ito. Polynomial-space approximation of no-signaling provers. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming, ICALP '10*, pages 140–151, 2010.
- 29 Tsuyoshi Ito, Hirofumi Kobayashi, and Keiji Matsumoto. Oracularization and two-prover one-round interactive proofs against nonlocal strategies. In *Proceedings of the 24th IEEE Annual Conference on Computational Complexity, CCC '09*, pages 217–228, 2009.
- 30 Tsuyoshi Ito and Thomas Vidick. A multi-prover interactive proof for NEXP sound against entangled provers. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science, FOCS '12*, pages 243–252, 2012.
- 31 Nick S. Jones and Lluís Masanes. Interconversion of nonlocal correlations. *Physical Review A*, 72:052312, 2005.
- 32 Yael Kalai, Ran Raz, and Ron Rothblum. Delegation for bounded space. In *Proceedings of the 45th ACM Symposium on the Theory of Computing, STOC '13*, pages 565–574, 2013.
- 33 Yael Tauman Kalai, Ran Raz, and Oded Regev. On the space complexity of linear programming with preprocessing. In *Proceedings of the 7th Innovations in Theoretical Computer Science Conference, ITCS '16*, pages 293–300, 2016.
- 34 Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *Proceedings of the 46th ACM Symposium on Theory of Computing, STOC '14*, pages 485–494, 2014. Full version available at <https://eccc.weizmann.ac.il/report/2013/183/>.
- 35 Leonid A. Khalfin and Boris S. Tsirelson. Quantum and quasi-classical analogs of Bell inequalities. *Symposium on the Foundations of Modern Physics*, pages 441–460, 1985.
- 36 Noah Linden, Sandu Popescu, Anthony J. Short, and Andreas Winter. Quantum nonlocality and beyond: Limits from nonlocal computation. *Physical Review Letters*, 99:180502, 2007.
- 37 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.
- 38 Lluís Masanes, Antonio Acín, and Nicolas Gisin. General properties of nonsignaling theories. *Physical Review A*, 73:012112, 2006.
- 39 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.
- 40 Sandu Popescu and Daniel Rohrlich. Quantum nonlocality as an axiom. *Foundations of Physics*, 24(3):379–385, 1994.
- 41 Sandu Popescu and Daniel Rohrlich. *Causality and Nonlocality as Axioms for Quantum Mechanics*, pages 383–389. Springer Netherlands, 1998.
- 42 Prasad Raghavendra and David Steurer. Integrality gaps for strong SDP relaxations of UNIQUE GAMES. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science, FOCS '09*, pages 575–585, 2009. Full version at <http://people.eecs.berkeley.edu/~prasad/Files/cspgaps.pdf>.
- 43 Peter Rastall. Locality, Bell's theorem, and quantum mechanics. *Foundations of Physics*, 15(9):963–972, 1985.
- 44 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th ACM Symposium on Theory of Computing, STOC '97*, pages 475–484, 1997.



- 45 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.
- 46 Adi Shamir.  $IP = PSPACE$ . *Journal of the ACM*, 39(4):869–877, 1992.
- 47 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(3):411–430, 1990.
- 48 Anthony J. Short, Nicolas Gisin, and Sandu Popescu. The physics of no-bit-commitment: Generalized quantum non-locality versus oblivious transfer. *Quantum Information Processing*, 5(2):131–138, 2006.
- 49 Anthony J. Short, Sandu Popescu, and Nicolas Gisin. Entanglement swapping for generalized nonlocal correlations. *Physical Review A*, 73:012101, 2006.
- 50 Amir Shpilka and Avi Wigderson. Derandomizing homomorphism testing in general groups. In *Proceedings of the 36th ACM Symposium on the Theory of Computing*, STOC '04, pages 427–435, 2004.
- 51 Wim van Dam. Implausible consequences of superstrong nonlocality. *Natural Computing*, 12:9–12, 2013.
- 52 Thomas Vidick. Linearity testing with entangled provers, 2014. [http://users.cms.caltech.edu/~vidick/linearity\\_test.pdf](http://users.cms.caltech.edu/~vidick/linearity_test.pdf).
- 53 Stefan Wolf and Jürg Wullschleger. Oblivious transfer and quantum non-locality. In *Proceedings of the 2005 International Symposium on Information Theory*, ISIT '05, pages 1745–1748, 2005.



# Earthmover Resilience and Testing in Ordered Structures

**Omri Ben-Eliezer**

School of Computer Science, Tel Aviv University, Tel Aviv, Israel  
omrib@mail.tau.ac.il

**Eldar Fischer**

Faculty of Computer Science, Israel Institute of Technology (Technion), Haifa, Israel  
eldar@cs.technion.ac.il

---

## Abstract

---

One of the main challenges in property testing is to characterize those properties that are testable with a constant number of queries. For unordered structures such as graphs and hypergraphs this task has been mostly settled. However, for ordered structures such as strings, images, and ordered graphs, the characterization problem seems very difficult in general.

In this paper, we identify a wide class of properties of ordered structures – the *earthmover resilient* (ER) properties – and show that the “good behavior” of such properties allows us to obtain general testability results that are similar to (and more general than) those of unordered graphs. A property  $\mathcal{P}$  is ER if, roughly speaking, slight changes in the order of the elements in an object satisfying  $\mathcal{P}$  cannot make this object far from  $\mathcal{P}$ . The class of ER properties includes, e.g., all unordered graph properties, many natural visual properties of images, such as convexity, and all hereditary properties of ordered graphs and images.

A special case of our results implies, building on a recent result of Alon and the authors, that the distance of a given image or ordered graph from *any* hereditary property can be estimated (with good probability) up to a constant additive error, using a constant number of queries.

**2012 ACM Subject Classification** Theory of computation → Streaming, sublinear and near linear time algorithms

**Keywords and phrases** characterizations of testability, distance estimation, earthmover resilient, ordered structures, property testing

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.18

**Related Version** <https://arxiv.org/abs/1801.09798>

**Acknowledgements** We wish to thank the anonymous reviewers for helpful comments.

## 1 Introduction

*Property testing* is mainly concerned with understanding the amount of information one needs to extract from an unknown input function  $f$  to approximately determine whether the function satisfies a property  $\mathcal{P}$  or is far from satisfying it. In this paper, the types of functions we consider are *strings*  $f: [n] \rightarrow \Sigma$ ; *images* or *matrices*  $f: [m] \times [n] \rightarrow \Sigma$ ; and *edge-colored graphs*  $f: \binom{[n]}{2} \rightarrow \Sigma$ , where the set of possible colors for each edge is  $\Sigma$ . In all cases  $\Sigma$  is a *finite* alphabet. Note that the usual notion of a graph corresponds to the special case where  $|\Sigma| = 2$ .

The systematic study of property testing was initiated by Rubinfeld and Sudan [33], and Goldreich, Goldwasser and Ron [24] were the first to study property testing of combinatorial structures. An  $\epsilon$ -*test* for a property  $\mathcal{P}$  of functions  $f: X \rightarrow \Sigma$  is an algorithm that, given



© Omri Ben-Eliezer and Eldar Fischer;  
licensed under Creative Commons License CC-BY  
33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 18; pp. 18:1–18:35

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



query access to an unknown input function  $f$ , distinguishes with good probability (say, with probability  $2/3$ ) between the case that  $f$  satisfies  $\mathcal{P}$  and the case that  $f$  is  $\epsilon$ -far from  $\mathcal{P}$ ; the latter meaning that one needs to change the values of at least an  $\epsilon$ -fraction of the entries of  $f$  to make it satisfy  $\mathcal{P}$ . In an  $n$ -vertex graph, for example, changing an  $\epsilon$ -fraction of the representation means adding or removing  $\epsilon \binom{n}{2}$  edges. (The representation model we consider here for graphs is the adjacency matrix. This is known as the *dense model*.)

In many cases, such as that of visual properties of images (where the input is often noisy to some extent), it is more natural to consider a robust variant of tests, that is *tolerant* to noise in the input. Such tests were first considered by Parnas, Ron and Rubinfeld [31]. A test is  $(\epsilon, \delta)$ -tolerant for some  $0 \leq \delta(\epsilon) < \epsilon$  if it distinguishes, with good probability, between inputs that are  $\epsilon$ -far from satisfying  $\mathcal{P}$  and those that are  $\delta(\epsilon)$ -close to (i.e., not  $\delta(\epsilon)$ -far from) satisfying  $\mathcal{P}$ .

One of the main goals in property testing is to characterize properties in terms of the number of queries required by an optimal test for them. If a property  $\mathcal{P}$  has, for any  $\epsilon > 0$ , an  $\epsilon$ -test that makes a constant number of queries, depending only on  $\epsilon$  and not on the size of the input, then  $\mathcal{P}$  is said to be *testable*.  $\mathcal{P}$  is *tolerantly testable* if for any  $\epsilon > 0$  it has a constant-query  $(\epsilon, \delta)$ -test for some  $0 < \delta(\epsilon) < \epsilon$ . Finally,  $\mathcal{P}$  is *estimable* if it has a constant query  $(\epsilon, \delta)$ -test for *any* choice of  $0 < \delta(\epsilon) < \epsilon$ . In other words,  $\mathcal{P}$  is estimable if the distance of an input to satisfying  $\mathcal{P}$  can be estimated up to a constant error, with good probability, using a constant number of queries.

The meta-question that we consider in this paper is the following.

*What makes a certain property  $\mathcal{P}$  testable, tolerantly testable, or estimable?*

## 1.1 Previous works: Characterizations of graphs and hypergraphs

For graphs, it was shown by Fischer and Newman [22] that the above three notions are equivalent, i.e., any testable graph property is estimable (and thus trivially also tolerantly testable). A combinatorial characterization of the testable graph properties was obtained by Alon, Fischer, Newman and Shapira [4] and analytic characterizations were obtained independently by Borgs, Chayes, Lovász, Sós, Szegedy and Vesztergombi [14] and Lovász and Szegedy [30] through the study of graph limits. The combinatorial characterization relates testability with *regular reducibility*, meaning, roughly speaking, that a graph property  $\mathcal{P}$  is testable (or estimable) if and only if satisfying  $\mathcal{P}$  is equivalent to approximately having one of finitely many prescribed types of Szemerédi regular partitions [35]. A formal definition of regular reducibility is given in Section 2.

Very recently, a similar characterization for hypergraphs was obtained by Joos, Kim, Kühn and Osthus [29], who proved that as in the graph case, testability, estimability and regular reducibility are equivalent for any hypergraph property.

A (partial) characterization of the graph properties  $\mathcal{P}$  that have a constant-query test whose error is one-sided (i.e., tests that always accept inputs satisfying  $\mathcal{P}$ ) was obtained by Alon and Shapira [5]. They showed that the only properties testable using an important and natural type of one-sided tests, that are *oblivious* to the input size, are essentially the *hereditary* properties.

The above characterizations for graphs rely on a conversion of tests into *canonical tests*, due to Goldreich and Trevisan [26]. A canonical test  $T$  always behaves as follows: First it picks a set  $U$  of vertices non-adaptively and uniformly at random in the input graph  $G$ , and queries all pairs of these vertices, to get the induced subgraph  $G[U]$ . Then  $T$  decides whether to accept or reject the input deterministically, based only on the identity of  $G[U]$  and the

size of  $G$ . The number of queries needed by the canonical test is only polynomial in the number of queries required by the original test, implying that any testable property is also canonically testable.

To summarize, all of the following conditions are equivalent for graphs: Testability, tolerant testability, canonical testability, estimability, and regular reducibility.

## 1.2 From unordered to ordered structures

Common to all of the above characterization results is the fact that they apply to unlabeled graphs and hypergraphs, which are *unordered* structures: Graph (and hypergraph) properties are symmetric in the sense that they are invariant under any relabeling (or equivalently, reordering) of the vertices. That is, if a labeled graph  $G$  satisfies an unordered graph property  $\mathcal{P}$ , then any graph resulting from  $G$  by changing the labels of the vertices is isomorphic to  $G$  (as an unordered graph), and so it satisfies  $\mathcal{P}$  as well.

A natural question that one may ask is whether similar characterizations hold for the more general setting of *ordered* structures over a finite alphabet, such as *images* and *vertex-ordered graphs* in the two-dimensional case, and *strings* in the one-dimensional case. While an unordered property is defined as a family of (satisfying) instances that is closed under relabeling, in the ordered setting, *any* family of instances is considered a valid property. The ordered setting is indeed much more general than the unordered one, as best exemplified by string properties: On one hand, unordered string properties are essentially properties of distributions over the alphabet  $\Sigma$ . On the other hand, *any* property of *any* finite discrete structure can be encoded as an ordered string property!

In general, the answer to the above question is *negative*. It is easy to construct simple string properties that are testable and even estimable, but are neither canonically testable nor regular reducible.<sup>1</sup> As an example, consider the binary string property  $\mathcal{P}_{111}$  of “not containing three consecutive ones”. The following is an  $\epsilon$ -test for  $\mathcal{P}_{111}$  (estimation is done similarly): Pick a random consecutive substring  $S$  of the input, of length  $O(1/\epsilon)$ , and accept if and only if  $S$  satisfies  $\mathcal{P}_{111}$ . On the other hand, global notions like canonical testability and regular reducibility cannot capture the local nature of  $\mathcal{P}$ .

Moreover, it was shown by Fischer and Fortnow [20], building on ideas from probabilistically checkable proofs of proximity (PCPP), that there exist testable string properties that are not tolerantly testable, as opposed to the situation in unordered graphs [22].

However, it may still be possible that a *positive* answer holds for the above question if we restrict our view to a class of “well behaved” properties.

*Does there exist a class of properties that is **wide** enough to capture many interesting properties, yet **well behaved** enough to allow simple characterizations for testability?*

So far, we have seen that in general, properties in which the exact location of entries is important to some extent, like  $\mathcal{P}_{111}$  and the property from [20], do not admit characterizations of testability that are similar to those of unordered graphs. But what about properties that

<sup>1</sup> To this end, canonical tests in ordered structures are similar to their unordered counterparts, but they act in an order-preserving manner. For example, a  $q$ -query test for a property  $\mathcal{P}$  of strings  $f: [n] \rightarrow \Sigma$  is *canonical* if, given an unknown string  $f: [n] \rightarrow \Sigma$ , the test picks  $q$  entries  $x_1 < \dots < x_q \in [n]$ , queries them to get the values  $y_1 = f(x_1), \dots, y_q = f(x_q)$ , and decides whether to accept or reject the input only based on the tuple  $(y_1, \dots, y_q)$ . Canonical tests in ordered graphs or images are defined similarly, but instead of querying a random substring, we query a random induced ordered subgraph or a random submatrix, respectively.

are ultimately global? Can one find, say, an ordered graph property that is canonically testable but not estimable, for example? Stated differently,

*Do the characterizations of testability in unordered graphs have analogues for canonical testability in ordered graphs and images?*

### 1.3 Our contributions

In this paper, we provide a partial positive answer to the first question, and a more complete positive answer to the second question. For the second question, we show that canonical testability in ordered graphs and images implies estimability and is equivalent to (an ordered version of) regular reducibility, similarly to the case in unordered graphs. Addressing the first question, we identify a wide class of well-behaved properties of ordered structures, called the *earthmover resilient* (ER) properties, providing characterizations of tolerant testability and estimability for these properties.

#### Earthmover resilient properties

Roughly speaking, a property  $\mathcal{P}$  of a certain type of functions is *earthmover resilient* if slight changes in the *order* of the “base elements”<sup>2</sup> of a function  $f$  satisfying  $\mathcal{P}$  cannot turn  $f$  into a function that is *far* from satisfying  $\mathcal{P}$ . The class of ER properties captures several types of interesting properties:

1. Trivially, *all* properties of *unordered* graphs and hypergraphs.
2. Global *visual* properties of images. In particular, this includes any property  $\mathcal{P}$  of black-white images satisfying the following: Any image  $I$  satisfying  $\mathcal{P}$  has a sparse black-white boundary. This includes, as special cases, properties like convexity and being a half plane, which were previously investigated in [10, 11, 15, 16, 32]. See Subsection 2.1 for the precise definitions and statement and Appendix A for the proof.
3. All *hereditary* properties of ordered graphs and images, as implied by a recent result of Alon and the authors [2]. While all hereditary unordered graph properties obviously fit under this category, it also includes interesting order-based properties, such as the widely investigated property of *monotonicity* (see [17, 18] for results on strings and images over a finite alphabet), *k-monotonicity* [15], forbidden poset type problems [21], and more generally forbidden submatrix type problems [1, 2, 3, 23].

#### The new results

ER properties behave well enough to allow us to fully characterize the *tolerantly testable* properties among them in images and ordered graphs. In strings, it turns out that earthmover resilience is *equivalent* to canonical testability.

Our first result relates between earthmover resilience, tolerant testability and canonical testability in images and edge colored ordered graphs.

► **Theorem 1** (See also Theorem 11). *The following conditions are equivalent for any property  $\mathcal{P}$  of edge colored ordered graphs or images.*

1.  $\mathcal{P}$  is earthmover resilient and tolerantly testable.
2.  $\mathcal{P}$  is canonically testable.

---

<sup>2</sup> The base elements in an ordered graph are the vertices, and in images these are the rows and the columns; in strings the base elements are the entries themselves.

Theorem 11, which is the more detailed version of Theorem 1, also states that *efficient* non-adaptive tolerant  $(\epsilon, \delta)$ -tests – in which the query complexity is polynomial in  $\delta(\epsilon)$  – can be converted, under certain conditions, into efficient canonical tests, and vice versa.

Let us note that Theorem 1 can be extended to high-dimensional ordered structures, such as tensors (e.g. 3D images) or edge colored ordered hypergraphs. As our focus in this paper is on one- and two-dimensional structures, the full proof of the extended statement is not given here, but it is a straightforward generalization of the 2D proof.

In (one-dimensional) strings, it turns out that the tolerant testability condition of Theorem 1 is not needed. That is, ER and canonical testability are equivalent for string properties.

► **Theorem 2.** *A string property  $\mathcal{P}$  is canonically testable if and only if it is earthmover resilient.*

As in Theorem 1, the transformation between canonical testability and earthmover resilience is efficient: If a string property has a tolerant  $(\epsilon, \delta(\epsilon))$ -test for any  $\epsilon > 0$ , and  $\delta$  is polynomial in  $\epsilon$ , then the number of queries of the corresponding canonical test is also polynomial in  $\epsilon$ . The converse is also true: A canonical  $\epsilon$ -test for  $\mathcal{P}$  with number of queries that is polynomial in  $\epsilon$  is in fact a tolerant  $(\epsilon, \delta)$ -test for  $\mathcal{P}$  where  $\delta(\epsilon)$  is polynomial in  $\epsilon$ .

In the unordered graph case, it was shown that testability is equivalent to estimability [22] and to regular reducibility [4]; we note that the conversions induce a tower-type blowup in the number of queries. Here, we establish analogous results for canonical tests in ordered structures. The notion of (ordered) regular reducibility that we use here is similar in spirit to the unordered variant, but is slightly more involved. The formal definition is given in Subsection 2.5.

► **Theorem 3.** *Any canonically testable property of edge colored ordered graphs and images is (canonically) estimable.*

► **Theorem 4.** *A property of edge colored ordered graphs or images is canonically testable if and only if it is regular reducible.*

#### A tower-type blowup in Theorems 3 and 4

While the conversion between tolerant tests and canonical tests (and vice versa) among earthmover resilient properties has a reasonable polynomial blowup in the number of queries under certain conditions, for the relation between canonical testability and estimability or regular reducibility this is not known to be the case. The proofs of Theorems 3 and 4 go through Szemerédi-regularity type arguments, and this yields at least a tower-type blowup in the number of queries. Currently, it is not known how to avoid this tower-type blowup in general, even for unordered graphs. However, interesting recent results of Hoppen, Kohayakawa, Lang, Lefmann and Stagni [27, 28] state that for hereditary properties of unordered graphs, the blowup between testability and estimability is at most exponential, and extending this line of research would serve as an intriguing direction for future research.

The characterization of tolerant testability in ER properties, given below, is a direct corollary of Theorems 1, 3, and 4.

► **Corollary 5.** *The following conditions are equivalent for any earthmover resilient property  $\mathcal{P}$  of edge colored ordered graphs or images.*

1.  $\mathcal{P}$  is tolerantly testable.
2.  $\mathcal{P}$  is canonically testable.

3.  $\mathcal{P}$  is estimable.
4.  $\mathcal{P}$  is regular reducible.

Alon and the authors [2] recently showed that any hereditary property of edge-colored ordered graphs and images is canonically testable, by proving an order-preserving removal lemma for all such properties. From Theorem 3 and [2] we derive the following very general result.

► **Corollary 6.** *Any hereditary property of edge-colored ordered graphs or images is (canonically) estimable.*

In particular, this re-proves the estimability of previously investigated properties such as monotonicity [17, 18] and more generally  $k$ -monotonicity [15], and proves the estimability of forbidden-submatrix and forbidden-poset type properties [1, 2, 3, 21, 23].

► **Remark.** The characterization of the one-sided error obliviously testable properties by Alon and Shapira [5], mentioned in Subsection 1.1, carries on to canonical tests in ordered graphs and images. That is, a property  $\mathcal{P}$  of such structures has a one-sided error oblivious canonical test if and only if it is (essentially) hereditary. The fact that hereditary properties are obliviously canonically testable with one-sided error is proved in [2]; the proof of the other direction is very similar to its analogue in unordered graphs [5], and is therefore omitted.

## 1.4 Related work

### Canonical versus sample-based testing in strings

The notion of a sample-based test, already defined in the seminal work of Goldreich, Goldwasser and Ron [24], refers to tests that cannot choose which queries to make. A  $q$ -query test for  $\mathcal{P}$  is *sample-based* if it receives pairs of the form  $(x_1, f(x_1)), \dots, (x_q, f(x_q))$  where  $f$  is the unknown input function and  $x_1, \dots, x_q$  are picked uniformly at random from the domain of  $X$  (compare this to the definition of canonical tests from Subsection 1.2). A recent work of Blais and Yoshida [13] characterizes the properties  $\mathcal{P}$  that have a constant query *sample-based* test.

In strings, sample-based testability might seem equivalent to *canonical* testability at first glance, but this is actually not the case, as sample-based tests have more power than canonical ones (canonical testability implies sample-based testability, but the converse is not true). Consider, e.g., the property of equality to the string 010101..., which is trivially sample-based testable, yet not canonically testable. Thus, sample-based testability does not imply canonical testability, so the results of Blais and Yoshida [13] are not directly comparable to Theorem 2 above.

### Previously investigated properties of ordered structures

On top of the hereditary properties mentioned earlier, several different types of properties of ordered structures have been investigated in the property testing literature. Without trying to be comprehensive, here is a short summary of some of these types of properties.

**Geometric & visual properties.** Image properties that exhibit natural visual conditions, such as connectivity, convexity and being a half plane, were considered e.g. in [11, 10, 16, 32]. Typically in these cases, images with two colors – black and white – are considered, where the “shape” consists of all black pixels, and the “background” consists of all white pixels. For example, convexity simply means that the black shape is convex. As we shall see, some



of these properties that are global in nature, such as convexity and being a half plane, are ER, while connectivity – a property that is sensitive to local modifications – is not ER.

**Algebraic properties.** String properties related to low-degree polynomials, PCPs and locally testable error correcting codes have been thoroughly investigated, starting with the seminal papers of Rubinfeld and Sudan [33] and Goldreich and Sudan [25]. As shown in [20], there exist properties of this type that are testable but not tolerantly testable. In this sense, algebraic properties behave very differently from unordered graph properties. This should not come as a surprise: In a PCP or a code, the exact *location* of each bit is majorly influential on its “role”. This kind of properties is therefore not ER in general.

**Local properties.** These are image properties  $\mathcal{P}$  where one can completely determine whether a given image  $\mathcal{I}$  satisfies  $\mathcal{P}$  based only on the statistics of the  $k \times k$  consecutive sub-images of  $\mathcal{I}$ , for a fixed constant  $k$ . Recently, Ben-Eliezer, Korman and Reichman [9] observed that for almost all (large enough) patterns  $Q$ , the local property of not containing a *consecutive* copy of  $Q$  in the image is tolerantly testable. Note that monotonicity can also be represented as a local property, taking  $k = 2$  (but  $\ell$ -monotonicity cannot be represented this way). Local properties are not ER in general, and obtaining characterizations of testability for them remains an intriguing open problem.

## 2 Preliminaries

This Section contains all required definitions, including those that are related to earthmover resilience (Subsection 2.1), a discussion on earthmover resilient properties (Subsection 2.2), property testing notation (Subsection 2.3), and finally, the definition of ordered regular reducibility (Subsection 2.5). Along the way, we state the full version of Theorem 1 (Subsection 2.4).

We start with some standard definitions. A *property*  $\mathcal{P}$  of functions  $f: X \rightarrow \Sigma$  is simply viewed as a collection of such functions, where  $f$  is said to *satisfy*  $\mathcal{P}$  if  $f \in \mathcal{P}$ . The *absolute* Hamming distance between two functions  $f, f': X \rightarrow Y$  is  $D_H(f, f') = |\{x \in X : f(x) \neq f'(x)\}|$ , and the *relative distance* is  $d_H(f, f') = D_H(f, f')/|X|$ ; note that  $0 \leq d_H(f, f') \leq 1$  always holds.  $f$  and  $f'$  are  $\epsilon$ -*far* if  $d_H(f, f') > \epsilon$ , and  $\epsilon$ -*close* otherwise. The distance of  $f$  to a property  $\mathcal{P}$  is  $\min_{f' \in \mathcal{P}} d_H(f, f')$ .  $f$  is  $\epsilon$ -*far* from  $\mathcal{P}$  if the distance between  $f$  and  $\mathcal{P}$  is larger than  $\epsilon$ , and  $\epsilon$ -*close* to  $\mathcal{P}$  otherwise.

### Representing images using ordered graphs

An image  $f: [n] \times [n] \rightarrow \Sigma$  can be represented by an edge colored ordered graph  $g: \binom{[2n]}{2} \rightarrow \Sigma \cup \{\perp\}$ , where  $\perp \notin \Sigma$  can be thought of as a special “no edge” symbol.  $g$  is defined as follows.  $g(x, y) = \perp$  for any pair  $x \neq y$  satisfying  $1 \leq x, y \leq n$  (“pair of rows”) or  $n + 1 \leq x, y \leq 2n$  (“pair of columns”); and  $g(x, n + y) = f(x, y)$  for any  $x, y \in [n]$ . From now onwards, we almost exclusively use this representation of images as ordered graphs, usually giving our definitions and proofs only for strings and ordered graphs. It is not hard to verify that all results established for ordered graphs can be translated to images through this representation.

### 2.1 Earthmover resilience

We now formalize our notion of being “well behaved”. As both strings and ordered graphs are essentially functions of the form  $f: \binom{[n]}{k} \rightarrow \Sigma$  (for  $k = 1$  and  $k = 2$ , respectively), we simplify the presentation by giving here the general definition for functions of this type.

► **Definition 7** (Earthmover distance). Fix  $k > 0$  and let  $f: \binom{[n]}{k} \rightarrow \Sigma$ . A *basic move* between consecutive elements  $x, x+1 \in [n]$  in  $f$  is the operation of swapping  $x$  and  $x+1$  in  $f$ . Formally, let  $\sigma_x: [n] \rightarrow [n]$  be the permutation satisfying  $\sigma_x(x) = x+1$ ,  $\sigma_x(x+1) = x$ , and  $\sigma_x(i) = i$  for any  $i \neq x, x+1$ . For any  $X \in \binom{[n]}{k}$ , define  $\sigma_x^k(X) = \{\sigma_x(i) : i \in X\}$ . The result of a basic move between  $x$  and  $x+1$  in  $f$  is the composition  $f' = f \circ \sigma_x^k$ .

The *absolute earthmover distance*  $D_e(f, f')$  between two functions  $f, f': \binom{[n]}{k} \rightarrow \Sigma$  is the minimum number of basic move operations needed to produce  $f'$  from  $f$ . The distance is defined to be  $+\infty$  if  $f'$  cannot be obtained from  $f$  using any number of basic moves. The *normalized earthmover distance* between  $f$  and  $f'$  is  $d_e(f, f') = D_e(f, f') / \binom{[n]}{k}$ , and we say that they are  $\epsilon$ -earthmover-far if  $d_e(f, f') > \epsilon$ , and  $\epsilon$ -earthmover-close otherwise.

Our definition of earthmover distance matches the standard definition [34] for  $k = 1$ , and we extend it conservatively to higher  $k$ , so that the basic earthmoving step involves switching between neighboring vertices (or neighboring rows or columns, in the case of images). For images, this definition is non-standard; In [34], for example, the basic move in images corresponds to switching between neighboring entries (compared to switching neighboring rows and columns, as in our case). Our definition is much more restrictive than that of [34] in general: There exist two images  $f$  and  $f'$  such that the absolute distance between them is  $\infty$  under our definition, and 1 under the definition of [34].

► **Definition 8** (Earthmover resilience). Fix a function  $\delta: (0, 1) \rightarrow (0, 1)$ . A property  $\mathcal{P}$  is  $\delta$ -*earthmover resilient* if for any  $\epsilon > 0$ , function  $f$  satisfying  $\mathcal{P}$ , and function  $f'$  which is  $\delta(\epsilon)$ -earthmover-close to  $f$ , it holds that  $f'$  is  $\epsilon$ -close to  $\mathcal{P}$  (in the usual Hamming distance).  $\mathcal{P}$  is *earthmover resilient* if it is  $\delta$ -earthmover resilient for some choice of  $\delta$ .

Intuitively, a property is earthmover resilient if it is insensitive to local changes in the order of the base elements.

### Hereditary properties are earthmover resilient

It was shown in [2] that any hereditary property satisfies a *removal lemma*: If an ordered graph (or image)  $G$  is  $\epsilon$ -far from an hereditary property  $\mathcal{P}$ , then  $G$  contains  $\delta n^h$  ordered copies of some  $h$ -vertex subgraph  $H$  not satisfying  $\mathcal{P}$ , for suitable choices of  $\delta = \delta_{\mathcal{P}}(\epsilon) > 0$  and  $h = h_{\mathcal{P}}(\epsilon) > 0$ . Since one basic move can destroy no more than  $n^{h-2}$  such  $H$ -copies (those that include both swapped vertices), one has to make at least  $\delta n^2$  basic moves to make  $G$  satisfy  $\mathcal{P}$ . Thus,  $\epsilon$ -farness implies  $\delta_{\mathcal{P}}(\epsilon)$ -earthmover-farness from  $\mathcal{P}$ .

## 2.2 Earthmover resilience in visual properties

Convexity and being a half plane are earthmover resilient. This is a special case of a much wider phenomenon concerning properties of black-white images in which the number of pixels lying in the boundary between the black shape and the white background is small. Here, an  $m \times n$  white/black image is represented by a 0/1-matrix  $M$  of the same dimensions, where the  $(i, j)$ -pixel of the image is black if and only if  $M(i, j) = 1$ . The definition below is given for square images, but can be easily generalized to  $m \times n$  images with  $m = \Theta(n)$ .

► **Definition 9** (Sparse boundary). The *boundary*  $\mathcal{B} = \mathcal{B}(\mathcal{I})$  of an  $n \times n$  black-white image  $\mathcal{I}$  is the set of all pixels in  $\mathcal{I}$  that are black and have a white neighbor.<sup>3</sup>  $\mathcal{B}$  is  $c$ -sparse for a

<sup>3</sup> Here, two pixels are neighbors if they share one coordinate and differ by one in the other coordinate. An alternative definition (that will yield the same results in our case) is that two pixels are neighbors if they differ by at most one in each of the coordinates, and are not equal.

constant  $c > 0$  if  $|\mathcal{B}| \leq cn$ . A property  $\mathcal{P}$  has a  $c$ -sparse boundary if the boundaries of all images satisfying  $\mathcal{P}$  are  $c$ -sparse.

For example, for any property  $\mathcal{P}$  of  $n \times n$  images such that the black area in any image satisfying  $\mathcal{P}$  is the union of at most  $t$  convex shapes (that do not have to be disjoint),  $\mathcal{P}$  has a  $4t$ -sparse boundary. This follows from the fact that the boundary of each of the black shapes is of size at most  $4n$ . For  $t = 1$ , this captures both convexity and being a half plane as special cases. The following result states that  $c$ -sparse properties are earthmover resilient.

► **Theorem 10.** *Fix  $c \geq 1$ . Then any property with a  $c$ -sparse boundary is  $\delta$ -earthmover-resilient, where  $\delta(\epsilon) \leq \alpha\epsilon^2/c^2$  for some absolute constant  $\alpha > 0$  and any  $\epsilon > 0$ .*

The result still holds if  $c$  is taken as a function of  $\epsilon$ . The (non-trivial) proof serves as a good example showing how to prove earthmover resilience of properties, and is given in Appendix A.

Naturally, not all properties of interest are earthmover resilient. For example, the local property  $\mathcal{P}$  of “not containing two consecutive horizontal black pixels” in a black/white image is not earthmover resilient: Consider the chessboard  $n \times n$  image, which satisfies  $\mathcal{P}$ , but by partitioning the board into  $n/4$  quadruples of consecutive columns and switching between the second and the third column in each quadruple, we get an image that is  $O(1/n)$ -earthmover-close to  $\mathcal{P}$  yet  $1/4$ -far from it in Hamming distance. A similar but slightly more complicated example shows that connectivity is not earthmover resilient as well.

### 2.3 Definitions: Testing and estimation

A  $q$ -query algorithm  $T$  is said to be an  $\epsilon$ -test for  $\mathcal{P}$  with *confidence*  $c > 1/2$ , if it acts as follows. Given an unknown input function  $f : X \rightarrow \Sigma$  (where  $X$  and  $\Sigma$  are known),  $T$  picks  $q$  elements  $x_1, \dots, x_q \in X$  of its choice, and queries the values  $f(x_1), \dots, f(x_q)$ .<sup>4</sup> Then  $T$  decides whether to accept or reject  $f$ , so that

- If  $f$  satisfies  $\mathcal{P}$  then  $T$  accepts  $f$  with probability at least  $c$ .
- If  $f$  is  $\epsilon$ -far from  $\mathcal{P}$ , then  $T$  rejects it with probability at least  $c$ .

Now let  $\delta : (0, 1) \rightarrow (0, 1)$  be a function that satisfies  $\delta(x) < x$  for any  $0 < x < 1$ . An  $(\epsilon, \delta)$ -tolerant test  $T$  is defined similarly to an  $\epsilon$ -test, with the first condition replaced with the following strengthening: If  $f$  is  $\delta(\epsilon)$ -close to  $\mathcal{P}$ , then  $T$  accepts it with probability at least  $1 - c$ . Unless stated otherwise, the default choice for the confidence is  $c = 2/3$ .  $\mathcal{P}$  is *testable* if it has a constant-query  $\epsilon$ -test (whose number of queries depends only on  $\epsilon$ ) for any  $\epsilon > 0$ . Similarly,  $\mathcal{P}$  is  $\delta$ -tolerantly testable, for a valid choice of  $\delta : (0, 1) \rightarrow (0, 1)$ , if it has a constant query  $(\epsilon, \delta)$ -test for any  $\epsilon > 0$ . If  $\mathcal{P}$  is  $\delta$ -tolerantly testable for *some* valid choice of  $\delta$ , we say that it is *tolerantly testable*. Finally,  $\mathcal{P}$  is *estimable* if it is  $\delta$ -tolerantly testable for *any* valid choice of  $\delta$ .

Next, we formally define what it means for a test (or a tolerant test)  $T$  to be *canonical*, starting with the definition for strings.

A  $q$ -query test (or tolerant test)  $T$  for a property  $\mathcal{P}$  of strings  $f : [n] \rightarrow \Sigma$  is *canonical* if it acts in two steps. First, it picks  $x_1 < \dots < x_q$  uniformly at random, and queries the entries  $y_1 = f(x_1), \dots, y_q = f(x_q)$ . The second step only receives the ordered tuple  $Y = (y_1, \dots, y_q)$  and decides (possibly probabilistically) whether to accept or reject only based on the values

<sup>4</sup>  $T$  as defined here is a *non-adaptive* test, that chooses which queries to make in advance. Adaptivity does not matter for our discussion, since we are only interested in constant-query tests, and since an adaptive test making a constant number  $q$  of queries can be turned into a non-adaptive one making  $2^q$  queries, which is still a constant.

of  $Y$ . Note that the second step does *not* “know” the values of  $x_1, \dots, x_q$  themselves. As before,  $\mathcal{P}$  is *canonically testable* if it has a  $q_{\mathcal{P}}(\epsilon)$ -query canonical test for any  $\epsilon > 0$ , where  $q_{\mathcal{P}}(\epsilon)$  depends only on  $\epsilon$ .

In contrast, a test for string properties is *sample based* if it has the exact same first step, but the second step receives more information: It also receives the values of  $x_1, \dots, x_q$ . A sample-based test is more powerful than a canonical test in general. For example, the property of “being equal to the string 010101...” is trivially sample-based  $\epsilon$ -testable with  $O(1/\epsilon)$  queries, but is *not* canonically testable with a constant number of queries (that depends only on  $\epsilon$ ).

For *ordered graphs*  $f: \binom{[n]}{2} \rightarrow \Sigma$ , a test (or a tolerant test)  $T$  is *canonical* if, again, it acts in two steps. In the first step,  $T$  picks  $q$  vertices  $v_1 < \dots < v_q$  uniformly at random, and queries all  $\binom{q}{2}$  values  $y_{ij} = f(v_i, v_j)$ . The second step receives the ordered tuple  $Y = (y_{11}, y_{12}, \dots, y_{1q}, \dots, y_{q-1,q})$ , and decides (possibly probabilistically) whether to accept or reject only based on the value of  $Y$ .

We take a short detour to explain why asking  $T$  to make a deterministic decision in the second step of the canonical test, rather than a probabilistic one, will not make an essential difference for our purposes. It was proved by Goldreich and Trevisan [26] that any probabilistic canonical test (for which the decision to accept or reject in the second step is not necessarily deterministic) can be converted into a deterministic one, with a blowup that is at most polynomial in the number of queries. The proof was given for unordered graph properties, but it can be translated to ordered structures like strings, ordered graphs and images in a straightforward manner. Thus, the requirement that the canonical test makes a deterministic decision is not restrictive.

## 2.4 The full statement of Theorem 1

We are finally ready to present the more precise version of Theorem 1. This version depicts an *efficient* transformation from earthmover resilience and tolerant testability to canonical testability, and vice versa.

► **Theorem 11.** *Let  $\mathcal{P}$  be a property of edge-colored ordered graphs or images, and let  $\delta: (0, 1) \rightarrow (0, 1)$  and  $\eta: (0, 1) \rightarrow (0, 1)$  such that  $\eta(\epsilon) < \epsilon$  for any  $\epsilon > 0$ .*

1. *If  $\mathcal{P}$  is  $\delta$ -earthmover resilient and  $\eta$ -tolerantly testable, where the number of queries of a corresponding  $(\epsilon, \eta)$ -tolerant non-adaptive test is denoted by  $q(\epsilon)$ , then  $\mathcal{P}$  is canonically testable. Moreover, if  $q$ ,  $\eta^{-1}$  and  $\delta^{-1}$  are polynomial in  $\epsilon^{-1}$ , then the number of queries of the canonical  $\epsilon$ -test is also polynomial in  $\epsilon^{-1}$ .*
2. *If  $\mathcal{P}$  is canonically testable, where the number of queries of the canonical (non adaptive)  $\epsilon$ -test is denoted by  $q'(\epsilon)$ , then  $\mathcal{P}$  is both  $\delta'$ -earthmover resilient and  $\delta'$ -tolerantly testable where  $\delta': (0, 1) \rightarrow (0, 1)$  depends only on  $q'$  and  $\epsilon$ . Moreover, if  $q'$  is polynomial in  $\epsilon^{-1}$ , then  $\delta'$  is polynomial in  $\epsilon$ .*

The proof is given along Sections 4, 5, and 6.

## 2.5 Regular reducibility

The last notion to be formally defined is that of ordered regular reducibility. This notion is a natural analogue of the unordered variant, and is rather complicated to describe and define. Since the intuition behind this definition is quite similar to that of the unordered case, we refer the reader to a more thorough discussion on regular reducibility (and the relation to Szemerédi’s regularity lemma) in [4]. Here, we only provide the set of definitions required for our purposes.

► **Definition 12** (Regularity, regular partition). Let  $f: \binom{[n]}{2} \rightarrow \Sigma$  be an edge-colored ordered graph. For any  $\sigma \in \Sigma$ , the  $\sigma$ -density of a disjoint pair  $A, B \subseteq [n]$  is  $d_\sigma(A, B) = |f^{-1}(\sigma) \cap (A \times B)| / |A||B|$ . A pair  $(A, B)$  is  $\gamma$ -regular if for any two subsets  $A' \subseteq A$  and  $B' \subseteq B$  satisfying  $|A'| \geq \gamma|A|$  and  $|B'| \geq \gamma|B|$ , and any  $\sigma \in \Sigma$ , it holds that  $|d_\sigma(A', B') - d_\sigma(A, B)| \leq \gamma$ . An equipartition of  $[n]$  into  $k$  parts  $V_1, \dots, V_k$  is  $\gamma$ -regular if all but at most  $\gamma \binom{k}{2}$  of the pairs  $(V_i, V_j)$  are  $\gamma$ -regular.

► **Definition 13** (Interval partitions). The  $k$ -interval equipartition of  $[n]$  is the unique partition of  $[n]$  into sets  $X_1, \dots, X_k$ , such that  $x < x'$  for any  $x \in X_i, x' \in X_{i'}, i < i'$  and  $|X_{i'}| \leq |X_i| \leq |X_{i'}| + 1$  for any  $i < i'$ . An interval partition of an ordered graph or a string is defined similarly.

► **Definition 14** (Ordered regularity instance). An ordered regularity instance  $R$  for  $\Sigma$ -colored ordered graphs is given by an error parameter  $\gamma$ , integers  $r, k$ , a set of  $K = \binom{r}{2} k^2 |\Sigma|$  densities  $0 \leq \eta_{ij}^{i'j'}(\sigma) \leq 1$  indexed by  $i < i' \in [r], j, j' \in [k]$  and  $\sigma \in \Sigma$ , and a set  $\bar{R}$  of tuples  $(i, j, i', j')$  of size at most  $\gamma K$ . An ordered graph  $f: \binom{[n]}{2} \rightarrow \Sigma$  satisfies the regularity instance if there is an equitable refinement  $\{V_{ij} : i \in [r], j \in [k]\}$  of the  $r$ -interval equipartition  $V_1, \dots, V_r$  where  $V_{ij} \subseteq V_i$  for any  $i$  and  $j$ , such that for all  $(i, j, i', j') \notin \bar{R}$  the pair  $V_{ij}, V_{i'j'}$  is  $\gamma$ -regular and satisfies  $d_\sigma(V_{ij}, V_{i'j'}) = \eta_{ij}^{i'j'}(\sigma)$  for any  $\sigma \in \Sigma$ . The complexity of the regularity instance is  $\max\{1/\gamma, K\}$ .

With some abuse of notation, when writing  $d_\sigma(V_{ij}, V_{i'j'}) = \eta_{ij}^{i'j'}(\sigma)$  we mean that the number of  $\sigma$ -colored edges between  $V_{ij}$  and  $V_{i'j'}$  is  $\lfloor \eta_{ij}^{i'j'}(\sigma) |V_{ij}| |V_{i'j'}| \rfloor$  or  $\lceil \eta_{ij}^{i'j'}(\sigma) |V_{ij}| |V_{i'j'}| \rceil$ . This way we avoid divisibility issues, without affecting any of our arguments.

The definition of an ordered regularity instance differs slightly from the analogous definition for unordered graphs in [4]: Here we insist that the regular partition will be a refinement of an interval equipartition, disregarding pairs of parts inside the same interval. We also allow a color set of size bigger than two. The definition of regular reducibility is analogous to the unordered case, though obviously the regularity instances used in the definition are of the ordered type.

► **Definition 15** (Regular reducible). An edge-colored ordered graph property  $\mathcal{P}$  is *regular-reducible* if for any  $\delta > 0$  there exists  $t = t_{\mathcal{P}}(\delta)$  such that for any  $n$  there is a family  $\mathcal{R}$  of at most  $t$  regularity instances, each of complexity at most  $t$ , such that the following holds for every  $\epsilon > 0$  and ordered graph  $f: \binom{[n]}{2} \rightarrow \Sigma$ :

- If  $f$  satisfies  $\mathcal{P}$  then for some  $R \in \mathcal{R}$ ,  $f$  is  $\delta$ -close to satisfying  $R$ .
- If  $f$  is  $\epsilon$ -far from satisfying  $\mathcal{P}$ , then for any  $R \in \mathcal{R}$ ,  $f$  is  $(\epsilon - \delta)$ -far from satisfying  $R$ .

### 3 Proof outline

In this section, we shortly describe the main ingredients of our proofs.

#### Earthmover distance and mixingness

Suppose that  $G, G' : \binom{[n]}{2} \rightarrow \Sigma$  are two ordered graphs with a finite earthmover distance between them (all results mentioned here also apply for strings). In this case,  $G$  and  $G'$  are isomorphic as unordered graphs, meaning that the collection of vertex permutations  $\pi : [n] \rightarrow [n]$  that “turn”  $G$  into  $G'$  is not empty. We define the (absolute) *mixingness* between  $G$  and  $G'$  as the minimal number of pairs  $x < y \in [n]$  such that  $\pi(x) > \pi(y)$ , over

all possible choice of  $\pi$  from the collection. We show, via a simple inductive proof, that the mixingness between  $G$  and  $G'$  is *exactly equal* to the earthmover distance between them.

With the tool of mixingness in hand, it is not hard to prove that canonical testability implies earthmover resilience and tolerant testability. The basic idea is that, if two graphs  $G$  and  $G'$  are sufficiently close in terms of mixingness, then the distributions of their  $q$ -vertex subgraphs are very similar, and so a  $q$ -query canonical test cannot distinguish between them with good probability. See Section 4 for more details.

### Earthmover resilience to piecewise-canonical testability

A test  $T$  is *piecewise-canonical* if it acts in the following manner on the  $t$ -interval partition of the unknown input graph (or string). First,  $T$  chooses how many vertices (entries, respectively) to take from each interval, where the number of vertices may differ between different intervals. Then  $T$  picks the vertices (entries) from the intervals in a uniformly random manner. Finally,  $T$  queries precisely all pairs of picked vertices (or all entries, in the string case), and decides whether to accept or reject based on the ordered tuple of the values returned by the queries.

For strings of length  $n$  over  $\Sigma$ , if  $\mathcal{P}$  is earthmover resilient then it is also piecewise-canonically testable. The main idea of the proof is the following. If one takes a string  $S$  and partitions it into sufficiently many equitable interval parts  $S_1, \dots, S_t$ , then “shuffling” entries inside each of the interval parts  $S_i$  will not change the distance of  $S$  to  $\mathcal{P}$  significantly. With this idea in hand, it is not hard to observe that knowing the histograms  $H_i$  of all parts  $S_i$  (with respect to letters in  $\Sigma$ ) is enough to estimate the distance of  $S$  to  $\mathcal{P}$  up to a small additive constant error. These histograms cannot be computed exactly with a constant number of queries, but it is well known that each  $H_i$  can be estimated up to a small constant error with a constant number of queries, which is enough for our purposes.

For properties  $\mathcal{P}$  of ordered graphs (or images), earthmover resilience by itself is not enough to imply piecewise-canonical testability, but earthmover resilience and tolerant testability are already enough. The idea is somewhat similar to the one we used for strings. We may assume that  $\mathcal{P}$  has a tolerant test  $T$  whose set of queried pairs is always an induced subgraph of  $G$ . Like before, we partition our input graph  $G$  into sufficiently many interval parts  $V_1, \dots, V_t$ . Now the piecewise canonical test  $T^*$  simulates a run of the original tolerant test  $T$  (without making the actual queries that  $T$  decided on). Denote the vertices that  $T$  decides to pick in  $V_i$  by  $v_1^i, \dots, v_{q_i}^i$ .  $T^*$  picks exactly  $q_i$  vertices uniformly at random in each part  $V_i$ , and queries all edges between all chosen vertices. Now  $T^*$  randomly “assigns” the labels  $v_1^i, \dots, v_{q_i}^i$  to the vertices that it queried from  $V_i$ , and returns the same answer that  $T$  would have returned for this set of queries. It can be shown that  $T^*$  is a test whose probability to return the same answer as  $T$  is high, as desired. For the full details, see Section 5.

### Piecewise-canonical testability to canonical testability

We describe the transformation for ordered graph properties; for strings this is very similar. Let  $T$  be piecewise-canonical test for  $\mathcal{P}$  that partitions the input into  $t$  intervals  $U_1, \dots, U_t$ . Consider the following canonical test  $T'$ :  $T'$  picks  $qt$  vertices  $v_1 < \dots < v_{qt}$  uniformly at random, for large enough  $q$ . Then  $T'$  partitions the vertices into  $t$  intervals  $A_1 = \{v_1, \dots, v_q\}, \dots, A_t = \{v_{(t-1)q+1}, v_{tq}\}$ . Now  $T'$  simulates a run of  $T$ . If  $T$  chose to take  $q_i$  vertices from  $U_i$ , then  $T'$  picks exactly  $q_i$  vertices from  $A_i$ . Finally,  $T'$  queries all edges between all vertices it picked, and returns the same answer as  $T$  (where the simulation of  $T$  assumes here that the vertices that were actually picked from  $A_i$  come from  $U_i$ ).

A rather straightforward but somewhat technical proof (that we do not describe at this point, see Section 6) shows that the probability that  $T'$  returns the answer that  $T$  would have returned on the same input is high, establishing the validity of  $T'$ . For the full details, see Section 6.

### Canonical testability, estimability and regular reducibility

The proofs of Theorems 3 and 4 are technically involved. Fortunately, the proofs follow the same spirit as those of the unordered case, considered in [4, 22], and in this paper we only describe how to adapt the unordered proofs to our case.

Sections 7 and 8 contain the proofs of Theorems 3 and 4, respectively. It is shown in these sections that for our ordered case, in some sense it is enough to make the proofs work for  $k$ -partite graphs, for a fixed  $k$ . The intuition is that for our purposes, it is enough to view an ordered graph  $G$  as a  $k$ -partite graph (for a large enough constant  $k$ ), where the parts are the intervals of a  $k$ -interval partition of  $G$ .

At this point, it is too difficult to explain the proof idea in high level without delving deeply into the technical details. Therefore, all details are deferred to Sections 7 and 8.

## 4 Earthmover-resilience and mixing

► **Definition 16.** Let  $\mu$  and  $\eta$  be two distributions over a finite family  $\mathcal{H}$  of combinatorial structures. The *variation distance* between  $\mu$  and  $\eta$  is  $|\mu - \eta| = \frac{1}{2} \sum_{H \in \mathcal{H}} |\Pr_{\mu}(H) - \Pr_{\eta}(H)|$ .

The following folklore fact regarding the variation distance will be useful later.

► **Lemma 17.** Let  $\mu$  and  $\eta$  be two distributions over a finite family  $\mathcal{H}$ . Then  $|\mu - \eta| = \max_{\mathcal{F} \subseteq \mathcal{H}} |\Pr_{\mu}(\mathcal{F}) - \Pr_{\eta}(\mathcal{F})| = \sum_{H \in \mathcal{H}: \Pr_{\mu}(H) > \Pr_{\eta}(H)} (\Pr_{\mu}(H) - \Pr_{\eta}(H))$ .

► **Definition 18.** An *unordered isomorphism* between two ordered graphs  $G, H : \binom{[n]}{2} \rightarrow \Sigma$  is a permutation  $\sigma : [n] \rightarrow [n]$  such that  $G(ij) = H(\sigma(i)\sigma(j))$  for any  $i < j \in [n]$ .

Given a permutation  $\sigma$  of  $[n]$ , the *mixing set* of  $\sigma$  is  $MS(\sigma) = \{i < j : \sigma(i) > \sigma(j)\} \subseteq \binom{[n]}{2}$ , its *mixingness* is  $D_m(\sigma) = |MS(\sigma)|$  and its *normalized mixingness* is  $d_m(\sigma) = |MS(\sigma)| / \binom{[n]}{2}$ . Given graphs  $G$  and  $H$ , their normalized mixingness  $d_m(G, H)$  is defined as the minimal normalized mixingness of an unordered isomorphism from  $G$  to  $H$  (and  $+\infty$  if  $G$  and  $H$  are not isomorphic as unordered graphs).

Our next goal is to show that the earthmover distance between two ordered graphs is equal to the mixingness between them. Given a permutation  $\sigma : [n] \rightarrow [n]$ , a *basic move* for  $\sigma$  transforms it to a permutation  $\sigma'$  of the same length, such that for some  $i$ ,  $\sigma(i) = \sigma'(i+1)$  and  $\sigma'(i) = \sigma(i+1)$ , and  $\sigma(j) = \sigma'(j)$  for any  $j \neq i, i+1$ . Let  $b(\sigma)$  denote the minimal number of basic moves required to turn  $\sigma$  into the identity permutation  $id$  satisfying  $id(i) = i$  for any  $i$ .

► **Lemma 19.**  $D_m(\sigma) = b(\sigma)$  for any permutation  $\sigma : [n] \rightarrow [n]$ .

**Proof.** The inequality  $D_m(\sigma) \leq b(\sigma)$  is trivial: Any basic move changes the relative order between a (single) pair of entries in the permutation, and thus cannot decrease the size of the mixing set by more than one. Next we show by induction that  $b(\sigma) \leq D_m(\sigma)$ .  $D_m(\sigma) = 0$  implies that  $\sigma = id$  and  $b(\sigma) = 0$  in this case. Now assume that  $D_m(\sigma) > 0$  and pick some  $i < j$  such that  $\sigma(i) > \sigma(j)$ . Take  $i' < j$  to be the largest for which  $\sigma(i') > \sigma(j)$  – such an  $i'$  exists since  $\sigma(i) > \sigma(j)$ . Note that  $\sigma(i'+1) \leq \sigma(j) < \sigma(i')$  due to the maximality of  $i'$ . Take



$\sigma'$  to be the result of the basic move between  $i'$  and  $i' + 1$  in  $\sigma$ .  $D_m(\sigma') = D_m(\sigma) - 1$ , and by the induction assumption we know that  $b(\sigma') = D_m(\sigma') = D_m(\sigma) - 1$ . But since  $\sigma'$  is the result of a basic move on  $\sigma$ , we conclude that  $b(\sigma) \leq b(\sigma') + 1 = D_m(\sigma)$ , as desired.  $\blacktriangleleft$

The equivalence between the earthmover distance and the mixingness is now immediate.

► **Lemma 20.** *For any two graphs  $G, H: \binom{[n]}{2} \rightarrow \Sigma$ ,  $d_e(G, H) = d_m(G, H)$ .*

**Proof.**  $D_m(G, H)$  is the minimum value of  $D_m(\sigma)$  among all unordered isomorphisms  $\sigma$  from  $G$  to  $H$ , and  $D_e(G, H)$  is the minimum value of  $b(\sigma)$  among all such isomorphisms. By Lemma 19, these two values are equal, and thus the corresponding relative measures are also equal.  $\blacktriangleleft$

► **Lemma 21.** *Let  $\delta : (0, 1) \rightarrow (0, 1)$  and let  $\mathcal{P}$  be a  $\delta$ -earthmover-resilient property. If two graphs  $G, H: \binom{[n]}{2} \rightarrow \Sigma$  satisfy  $d_e(G, H) \leq \delta(\epsilon)$  for some  $\epsilon > 0$ , then  $d_H(G, \mathcal{P}) \leq d_H(H, \mathcal{P}) + \epsilon$ .*

**Proof.** Suppose that  $G$  and  $H$  satisfy  $d_e(G, H) \leq \delta(\epsilon)$ . By definition, there exists an unordered isomorphism  $\sigma: G \rightarrow H$  such that  $d_m(G, H) = d_m(\sigma)$ . Let  $G': \binom{[n]}{2} \rightarrow \Sigma$  be the graph in  $\mathcal{P}$  that is closest to  $G$  (in Hamming distance). Consider the graph  $H'$  satisfying  $H'(\sigma(u)\sigma(v)) = G'(uv)$  for any  $u \neq v \in V$ , then  $d_H(H, H') = d_H(G, G')$ . Note that  $\sigma$  is an unordered isomorphism between  $G'$  and  $H'$ . It follows, building on Lemma 20, that  $d_m(G', H') \leq d_m(\sigma) = d_m(G, H) = d_e(G, H) \leq \delta(\epsilon)$ . This implies (by the earthmover resilience) that  $H'$  is  $\epsilon$ -close to  $\mathcal{P}$ . The triangle inequality concludes the proof.  $\blacktriangleleft$

### Canonical testability implies earthmover resilience

► **Definition 22.** Let  $H$  and  $G$  be  $\Sigma$ -edge-colored ordered graphs on  $q$  and  $n$  vertices respectively. The number of (ordered) copies of  $H$  in  $T$ , i.e., the number of induced subgraphs of  $G$  of size  $q$  isomorphic to  $H$ , is denoted by  $h(H, G)$ . The *density* of  $H$  in  $G$  is  $t(H, G) = h(H, G) / \binom{[n]}{q}$  (where  $t(H, G) = 0$  if  $q > n$ ). The  $q$ -*statistic* of  $G$  is the vector  $(t(H, G))_{H \in \mathcal{H}_q}$ , where  $\mathcal{H}_q$  is the family of all  $\Sigma$ -edge-colored ordered graphs with  $q$  vertices.

Every property of ordered graphs already testable by a canonical test is  $\delta$ -earthmover-resilient for some  $\delta$  (depending on the number of its query vertices as a function of  $\epsilon$ ), as implied by the following lemma.

► **Lemma 23.** *Let  $\epsilon, \delta > 0$ . For any canonical  $\epsilon$ -test querying up to  $q$  vertices and any two graphs  $G$  and  $G'$  of either Hamming distance or earthmover distance at most  $\delta$ , the difference between the acceptance probabilities of  $G$  and of  $G'$  is at most  $\delta \binom{q}{2}$ .*

**Proof.** We may assume that the test queries exactly  $q$  vertices. For Hamming distance, the statement is well known, and follows easily by taking a union bound over all  $\binom{q}{2}$  queried edges. Assume then that  $d_e(G, G') \leq \delta$ . Let  $\mu, \mu'$  be the  $q$ -statistics of  $G, G'$  respectively, where  $G, G': \binom{[n]}{2} \rightarrow \Sigma$  are two graphs with earthmover distance at most  $\delta$  between them. By Lemma 17 it will be enough to show that  $|\mu - \mu'| \leq \delta \binom{q}{2}$ . Lemma 20 implies that there is an unordered isomorphism  $\sigma: G \rightarrow G'$  with  $d_m(\sigma) \leq \delta$ .

For any set  $Q$  of  $q$  vertices, let  $\sigma(Q) = \{\sigma(v) : v \in Q\}$ , and note that  $Q \mapsto \sigma(Q)$  is a bijective mapping from  $\mathcal{H}_q$  to itself. Observe that the induced subgraph  $G[Q]$  can be non-isomorphic to  $G'[\sigma(Q)]$  (as an ordered graph on  $q$  vertices) only if there exist two vertices  $u, v \in Q$  satisfying  $uv \in MS(\sigma)$ . By a union bound, the probability of a uniformly random  $Q \in \mathcal{H}_q$  to have such a pair is at most  $d_m(\sigma) \binom{q}{2} \leq \delta \binom{q}{2}$ , implying that  $|\mu - \mu'| \leq \delta \binom{q}{2}$ .  $\blacktriangleleft$



The next lemma proves the second (and easier) direction of Theorem 11. It uses Lemma 23 to conclude that a canonically testable property is earthmover-resilient and tolerantly testable.

► **Lemma 24.** *Let  $\mathcal{P}$  be an ordered graph property. Suppose that  $\mathcal{P}$  has a canonical  $\epsilon$ -test  $T$  making  $q(\epsilon)$  vertex queries for any  $\epsilon > 0$ . Then  $\mathcal{P}$  is  $\delta$ -earthmover-resilient and  $\delta$ -tolerantly  $\epsilon$ -testable with  $9q(\epsilon)$  vertex queries, where  $\delta(\epsilon) = 1/20 \binom{q(\epsilon)}{2}$  for any  $\epsilon > 0$ .*

**Proof.** Let  $\epsilon > 0$ , and suppose that  $G$  and  $G'$  are of earthmover distance at most  $\delta(\epsilon)$  between them, where  $G$  satisfies  $\mathcal{P}$ ; to prove the earthmover resilience, we need to show that  $G'$  is  $\epsilon$ -close to satisfying  $\mathcal{P}$ . Since  $G \in \mathcal{P}$ , it is accepted by  $T$  with probability at least  $2/3$ . By Lemma 23, the acceptance probability of  $G'$  by  $T$  is at least  $\frac{2}{3} - \delta(\epsilon) \binom{q(\epsilon)}{2} > 1/3$ . Since  $T$  rejects any graph  $\epsilon$ -far from  $\mathcal{P}$  with probability at least  $2/3$ , we conclude that  $G'$  must be  $\epsilon$ -close to  $\mathcal{P}$ .

For the second part, regarding tolerant testability, Lemma 23 implies that for any graph that is  $\delta(\epsilon)$ -close to satisfying  $\mathcal{P}$ , the acceptance probability of  $T$  is at least  $2/3 - \delta(\epsilon) \binom{q(\epsilon)}{2} > 0.61$ . By applying  $T$  independently 9 times and accepting if and only if the majority of the runs accepted, we get a test that accepts  $\delta(\epsilon)$ -close graphs with probability at least  $2/3$  and rejects  $\epsilon$ -far graphs with probability at least  $2/3$  as well. This test can be made canonical with no need for additional queries. ◀

Let us finish with two comments. First, in the last two lemmas it was implicitly assumed that the canonical test is a deterministic one, but they also hold for randomized ones: The fact that  $|\mu - \mu'| \leq \delta \binom{q}{2}$  in Lemma 23 is actually enough to imply the statement of Lemma 23 for any (deterministic or randomized) canonical test, and Lemma 24 follows accordingly.

Second, the results in this section, along with Sections 5 and 6, are not exclusive to two-dimensional structures, and naturally generalize to  $k$ -dimensional structures for any  $k$ . Thus, in ordered hypergraphs and tensors in three dimensions or more, it is still true that the combination of earthmover resilience and tolerant testability is equivalent to canonical testability.

## 5 Piecewise-canonical testability

In this section, we show that ER string properties and ER tolerantly testable ordered graph properties have a constant-query *piecewise canonical test*. This is a test that consider a  $k$ -interval partition of the input, picking a predetermined number of vertices (or entries, in the string case) uniformly at random from each interval (this number may differ between different intervals), and finally, queries all edges between the picked vertices from all intervals. We always assume that our tolerant tests are non-adaptive and based on  $q$  query vertices (we assume they query the entire induced subgraph even if they do not use all of it). Note that unlike the case of unordered graphs, the move from an adaptive test to a non-adaptive one can cause an exponential blowup in the query complexity (we may need to “unroll” the entire decision tree).

► **Definition 25.** A (probabilistic) *piecewise-canonical* test with  $k$  parts and  $q$  query vertices for a property  $\mathcal{P}$  of functions  $f: \binom{[n]}{\ell} \rightarrow \Sigma$  works as follows. First, the test non-adaptively selects (possibly non-deterministically) numbers  $q_1, \dots, q_k$  that sum up to  $q$ , and then it considers a  $k$ -interval partition  $I_1, I_2, \dots, I_k$  of the input function  $f$ , selecting a uniformly random set of  $q_j$  vertices from  $I_j$  for every  $1 \leq j \leq k$ . The test finally accepts or rejects  $f$  based only on the selected numbers  $q_1, \dots, q_k$  and the unique function  $f': \binom{[q]}{\ell} \rightarrow \Sigma$  that is isomorphic (in the ordered sense) to the restriction of  $f$  on the selected vertices.

A property  $\mathcal{P}$  is *piecewise-testable* if for every  $\epsilon$  there exist  $k(\epsilon)$  and  $q(\epsilon)$  for which  $\mathcal{P}$  has a piecewise canonical  $\epsilon$ -test with  $k(\epsilon)$  parts and  $q(\epsilon)$  query vertices.

► **Remark.** In Section 2 it was noted that a probabilistic canonical test for a property can be transformed into a deterministic one, with the same confidence, as was shown in [26]. This is true for any choice of confidence  $c$  (not only the “default” confidence  $c = 2/3$ ). Since one can always amplify a (probabilistic or deterministic) test to get a test of the same type with confidence arbitrarily close to 1, we conclude that if a property  $\mathcal{P}$  has a probabilistic canonical test with a certain confidence  $c > 1/2$ , then for any  $\zeta > 0$ ,  $\mathcal{P}$  has a deterministic canonical test with confidence at least  $1 - \zeta$ .

All of the above is also true for piecewise-canonical tests; the proof for canonical tests carries over naturally to this case, so we omit it. Here, the simulating deterministic test has the same number of parts as the original test.

### 5.1 Strings: Earthmover resilience to piecewise-canonical testability

In this subsection, we prove that ER properties of strings are piecewise canonically testable. In Section 6, we show that the latter condition implies canonical testability.

For a string  $S : [n] \rightarrow \Sigma$  let  $d_\sigma(S) = |\Sigma^{-1}(\sigma)|/n$  denote the *density* of  $\sigma$  in  $S$ . Let  $T(S) = (d_\sigma(S))_{\sigma \in \Sigma}$  denote the distribution vector of letters in  $S$ . The following well known fact is important for the proof.

► **Fact 26.** *The distribution vector of a string over  $\Sigma$  can be approximated up to variation distance  $\zeta$ , with probability at least  $1 - \tau$ , using  $O(|\Sigma|^2 \log(\tau^{-1})\zeta^{-2})$  queries.*

Fix a function  $\delta: (0, 1) \rightarrow (0, 1)$ , a  $\delta$ -earthmover resilient property  $\mathcal{P}$  of strings over  $\Sigma$ , and  $\epsilon > 0$ . Take  $t = \lceil 1/2\delta(\epsilon/2) \rceil$ . For any string  $S$  over  $\Sigma$ , let  $S_1, \dots, S_t$  be the  $t$ -interval partition of  $S$  and let the  $t$ -interval distribution  $\Gamma_t(S) = (T(S_1), \dots, T(S_t))$  denote the  $t$ -tuple of the distribution vectors of  $S_1, \dots, S_t$ . For  $S$  as above and another string  $S'$  over  $\Sigma$  with  $t$ -interval partition  $S'_1, \dots, S'_t$ , the  $t$ -aggregated distance between  $S$  and  $S'$  is  $d_A(S, S') = \sum_{i=1}^t |T(S_i) - T(S'_i)| \cdot |S_i|/|S|$ ; recall that  $|T(S_i) - T(S'_i)|$  is the variation distance between  $T(S_i)$  and  $T(S'_i)$ . As usual, we define  $d_A(S, \mathcal{P}) = \min_{S' \in \mathcal{P}} d_A(S, S')$ . The next easy lemma relates between the Hamming distance and the  $t$ -aggregated distance of  $S$  to  $\mathcal{P}$ .

► **Lemma 27.** *For any string  $S$  over  $\Sigma$  we have  $0 \leq d_H(S, \mathcal{P}) - d_A(S, \mathcal{P}) \leq \epsilon/2$ .*

**Proof.** Let  $S'$  be the string that is closest to  $\mathcal{P}$  among those that can be generated from  $S$  only using basic moves inside the intervals  $S_1, \dots, S_t$ . In particular, it is trivial that  $d_H(S', \mathcal{P}) \leq d_H(S, \mathcal{P})$  and we know by Lemma 20 that  $d_e(S, S') \leq 2/t \leq \delta(\epsilon/2)$ . By Lemma 21, we get that  $d_H(S, \mathcal{P}) - d_H(S', \mathcal{P}) \leq \epsilon/2$ . On the other hand,  $d_H(S', \mathcal{P}) = d_A(S', \mathcal{P}) = d_A(S, \mathcal{P})$  follows by the definitions of the distance functions and the minimality of  $S'$ . ◀

Finally we present the piecewise canonical test for  $\mathcal{P}$ . More accurately, we describe a piecewise-canonical algorithm  $\mathcal{A}$  that, given an unknown string  $S$  over  $\Sigma$  of an unknown length  $n$ , approximates the  $t$ -aggregated distance of  $S$  to  $\mathcal{P}$  up to an additive error of  $\epsilon/6$ , with probability at least  $2/3$ . The test simply runs  $\mathcal{A}$  and accepts if and only if its output value is at most  $\epsilon/4$ . The algorithm  $\mathcal{A}$  acts as follows. First, it runs the algorithm of Fact 26 in each interval of the  $t$ -interval partition of  $S$ , with parameters  $\zeta = \epsilon/6$  and  $\tau = 1/3t$ . For any  $1 \leq i \leq t$ , let  $T_i^*$  denote the distribution returned by this algorithm for interval  $i$ . Then, Algorithm  $\mathcal{A}$  returns  $r = \min_{S' \in \mathcal{P}} \sum_{i=1}^t |T_i^* - T_{S'_i}| \cdot |S_i|/|S|$ .

With probability  $2/3$ , we get that  $|T(S_i) - T_i^*| \leq \epsilon/6$  for any  $i$ . Suppose from now on that the latter happens. It follows from the triangle inequality for the variation distance that

$d_A(S, \mathcal{P}) \leq d_A(S, S') \leq r + \epsilon/6$ , where  $r$  is the minimum defined above and  $S' \in \mathcal{P}$  is the string achieving this minimum. Conversely, there exists  $S'' \in \mathcal{P}$  such that  $d_A(S, S'') = d_A(S, \mathcal{P})$ . But the minimality of  $S'$  implies that  $\sum_{i=1}^t |T_i^* - T_{S''}^i| \cdot |S_i|/|S| \geq r$ , and again, from the triangle inequality we get that  $d_A(S, S'') \geq r - \epsilon/6$ . To summarize,

$$r - \epsilon/6 \leq d_A(S, S'') = d_A(S, \mathcal{P}) \leq d_A(S, S') \leq r + \epsilon/6$$

which means that  $r$  is, with probability at least  $2/3$ , an  $(\epsilon/6)$ -additive approximation of  $d_A(S, \mathcal{P})$ . Thus, if  $S$  satisfies  $\mathcal{P}$  (meaning that  $d_A(S, \mathcal{P}) = 0$ ) then with probability  $2/3$  the algorithm  $\mathcal{A}$  returns  $r \leq \epsilon/6$  and the test accepts. On the other hand, if  $S$  is  $\epsilon$ -far from  $\mathcal{P}$  then  $d_A(S, \mathcal{P}) \geq \epsilon/2$  by the above lemma, and  $\mathcal{A}$  returns  $r \geq \epsilon/2 - \epsilon/6 = \epsilon/3$  (making the test reject) with probability at least  $2/3$ , as desired.

## 5.2 Ordered graphs: ER and tolerant tests to piecewise-canonical tests

The next lemma shows that a tolerant test for an ER property  $\mathcal{P}$  of ordered graphs can be translated, in an efficient manner, into a piecewise-canonical test for  $\mathcal{P}$ .

► **Lemma 28.** *Let  $q : (0, 1) \rightarrow \mathbb{N}$ ,  $\eta : (0, 1) \rightarrow (0, 1)$ , and  $\delta : (0, 1) \rightarrow (0, 1)$ , and suppose that  $\mathcal{P}$  is a  $\delta$ -earthmover-resilient  $\eta$ -tolerantly testable property of ordered graphs, where for any  $\epsilon > 0$ , the corresponding  $(\epsilon, \eta(\epsilon))$ -tolerant test queries  $q(\epsilon)$  vertices. Then for any  $\epsilon > 0$  there exist  $q'$  and  $k$  such that  $\mathcal{P}$  has a piecewise-canonical  $\epsilon$ -test with  $k$  parts and  $q'$  query vertices. Moreover, if  $q, \eta, \delta$  are polynomial in  $\epsilon$ , then so are  $q'$  and  $k$ .*

**Proof.** Let  $T$  be a (non-adaptive)  $(\epsilon/2, \eta)$ -tolerant test for  $\mathcal{P}$  querying the induced subgraph on  $q' = q(\epsilon/2)$  vertices. Let  $G : \binom{[n]}{2} \rightarrow \Sigma$  denote the unknown input graph. Since  $T$  is non-adaptive, we may view it as a two-step algorithm acting as follows. In the first step,  $T$  chooses a  $q'$ -tuple  $x_1 < \dots < x_{q'} \in [n]$  (which will eventually be the vertices  $T$  will query) according to some distribution  $p_T$ . The second step receives the tuples  $(x_1, \dots, x_{q'})$  and  $(G(x_i x_j))_{i < j \in [q']}$  and decides (probabilistically) whether to accept or reject based only on these tuples.

Take  $k = \lceil 2/\delta(\eta(\epsilon/2)) \rceil$  and consider the  $k$ -interval partition  $I_1, \dots, I_k$  of the input graph  $G$ . Our piecewise-canonical test  $T'$ , also making  $q'$  vertex queries, is designed as follows. First it picks a tuple  $X$  of  $q'$  elements  $x_1 < \dots < x_{q'} \in [n]$  according to the distribution  $p_T$ . For each  $i = 1, \dots, k$ , let  $q_i = |X \cap I_i|$  and let  $S_i = \{1 + \sum_{j=1}^{i-1} q_j, \dots, \sum_{j=1}^i q_j\}$ .  $T'$  queries exactly  $q_i$  vertices from  $I_i$  uniformly at random. Now,  $T'$  picks a permutation  $\pi : [q'] \rightarrow [q']$  in the following manner: For each  $1 \leq i \leq k$ ,  $\pi$  restricted to  $S_i$  is a uniformly random permutation on  $[S_i]$ . Finally,  $T'$  runs the second step of the original test  $T$ , with tuples  $(x_1, \dots, x_{q'})$  and  $(G(x_{\pi(i)} x_{\pi(j)}))_{i < j \in [q']}$ .

Clearly,  $T'$  makes in total  $q'$  queries in  $k$  intervals, where the vertex queries within each interval are chosen uniformly at random. It only remains to show that  $T'$  is a valid  $\epsilon$ -test. Observe that applying  $T'$  on the input graph  $G$  is equivalent to the following process, in the sense that their output distribution (given any fixed  $G$ ) is identical.

1. “Shuffle” the vertices inside each interval  $I_i$  of  $G$  in a uniformly random manner, to get a new ordered graph  $G'$ .
2. Run the original test  $T$  on  $G'$ , and return its answer.

The relative mixingness between  $G$  and any such  $G'$  is at most  $k \binom{\lceil n/k \rceil}{2} / \binom{n}{2} < 2/k \leq \delta(\eta(\epsilon/2))$  where the first inequality holds for large enough  $n$ . By Lemmas 20 and 21 and the  $\delta$ -earthmover resilience of  $\mathcal{P}$ , we get that  $|d_H(G', \mathcal{P}) - d_H(G, \mathcal{P})| \leq \eta(\epsilon/2) < \epsilon/2$ . Thus, if  $G$  satisfies  $\mathcal{P}$ , then any  $G'$  possibly generated in the first step of the above process is  $\eta(\epsilon/2)$ -close to

$\mathcal{P}$ . Since  $T$  is  $(\epsilon/2, \eta)$ -tolerant, the second step of the process accepts with probability at least  $2/3$  for any fixed choice of  $G'$ . Thus, the process (or equivalently,  $T'$ ) accepts  $G$  with probability at least  $2/3$  in this case. Conversely, if  $G$  is  $\epsilon$ -far from  $\mathcal{P}$  then  $G'$  generated in the first step is  $\epsilon/2$ -far from  $\mathcal{P}$ , and, similarly, the process (or equivalently,  $T'$ ) rejects with probability at least  $2/3$ .  $\blacktriangleleft$

## 6 Piecewise-canonical testability to canonical testability

This section is dedicated to the proof that piecewise-canonically testable properties are canonically testable. While the proofs are presented here for ordered graphs, they can easily be translated to the case of strings. Therefore, the results in this section, combined with the previous two sections, complete the proof of Theorems 11 and 2.

► **Definition 29.** Given  $\{q_1, \dots, q_k\}$  that sum up to  $q$  and  $t \geq \max_{1 \leq j \leq k} q_j$ , the  $t$ -simulated piecewise distribution over subsets of  $[n]$  of size  $q$  is the result of the following process.

**Uniform sampling** Select a set of  $tk$  indices from  $[n]$ , uniformly at random. Let  $\{i_1, \dots, i_{tk}\}$  denote the set with its members sorted in ascending order.

**Simulation inside each block** For every  $1 \leq j \leq k$ , select a subset of  $\{i_{(j-1)t+1}, \dots, i_{jt}\}$  of size  $q_j$ , uniformly at random.

► **Lemma 30.** For every  $\delta, k$  and  $q$ , there exist  $t(\delta, k, q)$  and  $N(\delta, k, q)$  polynomial in  $\delta, k, q$ , so that if  $n > N(\delta, k, q)$  then the  $t$ -simulated piecewise distribution with respect to  $q_1, \dots, q_k$  is  $\delta$ -close (in the variation distance) to an actual piecewise distribution with respect to  $q_1, \dots, q_k$ , i.e., a process of the following type. Consider a  $k$ -interval partition  $I_1, \dots, I_k$  of the input graph, and for every  $1 \leq j \leq k$ , pick a uniformly random subset of  $I_j$  of size  $q_j$ .

In the proof of Lemma 30 we do not try to optimize the dependence of  $t$  and  $N$  on  $\delta, k, q$ , but just show that it is a reasonable polynomial dependence.

**Proof.** Fix  $q_1, \dots, q_k$  and write  $Q_i = \sum_{j=1}^i q_j$  for any  $1 \leq i \leq k$ . Also take  $q = Q_k$ . For any  $1 \leq l_1 < \dots < l_q \leq n$  denote by  $\Pr_{\text{piece}}(E_{l_1, \dots, l_q})$  the probability that the indices selected by a piecewise canonical distribution with parameters  $q_1, \dots, q_k$  are  $l_1, \dots, l_q$ . Similarly, for  $q_1, \dots, q_k$  as above and a fixed  $t \geq \max_{1 \leq j \leq k} q_j$ , we denote by  $\Pr_{\text{sim}}(E_{l_1, \dots, l_q})$  the probability that the indices selected by a simulated piecewise canonical distribution with parameters  $q_1, \dots, q_k$  and  $t$  are  $l_1, \dots, l_q$ . It is enough to show, for a suitable choice of  $t$  and for  $n$  large enough, that  $\sum_{l_1 < \dots < l_q} |\Pr_{\text{piece}}(E_{l_1, \dots, l_q}) - \Pr_{\text{sim}}(E_{l_1, \dots, l_q})| < \delta$ . To prove this, we show that there exist suitable events  $A$  and  $B$  satisfying the following conditions.

- $\Pr_{\text{piece}}(A) \leq \delta$  and  $\Pr_{\text{sim}}(B) \leq \delta$ .
- $\Pr_{\text{piece}}(E_{l_1, \dots, l_q} | \neg A) = \Pr_{\text{sim}}(E_{l_1, \dots, l_q} | \neg B)$  for any possible choice of  $l_1 < \dots < l_q$ , where  $\neg A$  and  $\neg B$  are the complementary events of  $A$  and  $B$ , respectively.

In the rest of the proof we define and analyze the events  $A$  and  $B$ .

**Order statistics.** Take  $t = 600k^4q^2\delta^{-3}$  and  $N = tk$ . Let  $1 \leq i_1 < \dots < i_N \leq n$  be the elements of an  $N$ -tuple from  $\binom{[n]}{N}$ , picked uniformly at random. It is well known (see, e.g., Chapter 3 in [6]) that the expected value of  $i_r$  – the  $r$ -th order statistic of the tuple – is  $\mu_r = r(n+1)/(N+1)$  and satisfies  $|\mu_r - rn/N| < n/N$ , and the variance of  $i_r$  is  $\sigma_r^2 \leq n^2/N$ .

By Chebyshev's inequality, for any  $1 \leq r \leq N$  it holds that  $\Pr(|i_r - \mu_r| > \alpha n) < 1/N\alpha^2$ . Pick  $\alpha = 3\sqrt{k/\delta N} < \delta/8k^2q$ . For any  $1 \leq j \leq k-1$ , we take  $r_j^-$  as the largest integer  $r$  for which  $\mu_r < (Q_j/k - \alpha - 1/N)n$  and  $r_j^+$  as the smallest integer  $r'$  for which  $\mu_{r'} >$

$(Q_j/k + \alpha + 1/N)n$ ; note that  $tQ_j - r_j^- < 2\alpha N$  and  $\mu_{r_j^-} > (Q_j/k - 2\alpha)n$ , and on the other hand,  $r_j^+ - tQ_j < 2\alpha N$  and  $\mu_{r_j^+} < (Q_j/k + 2\alpha)n$ . Intuitively speaking,  $r_j^-, r_j^+$  were chosen here with the following requirements in mind. With good probability,  $r_j^-$  needs to be contained in  $I_j$ ,  $r_j^+$  needs to be contained in  $I_{j+1}$ , and both  $r_j^-$  and  $r_j^+$  should be close to  $jn/k$  (which is roughly equal to the last element of  $I_j$  and the first element of  $I_{j+1}$ ).

Indeed, let  $C$  denote the event that

$$\left(\frac{Q_j}{k} - 3\alpha\right)n < i_{r_j^-} < \frac{Q_j}{k}n < i_{r_j^+} < \left(\frac{Q_j}{k} + 3\alpha\right)n \quad (1)$$

holds for any  $1 \leq j \leq k-1$ , and observe that  $\left(\frac{Q_j}{k} + 3\alpha\right)n < \left(\frac{Q_{j+1}}{k} - 3\alpha\right)n$  for any  $j$ .  $\neg C$  is contained in the event that, for some  $j$ ,  $|i_{r_j^-} - \mu_{r_j^-}| > \alpha n$  or  $|i_{r_j^+} - \mu_{r_j^+}| > \alpha n$ . The probability of the latter event is bounded by  $2k/N\alpha^2 = 2\delta/9$  by a union bound. Therefore  $C$  holds with probability at least  $1 - 2\delta/9$ .

**The “bad” events  $A$  and  $B$ .** Suppose that, after picking  $i_1 < \dots < i_N$  uniformly at random as above, we pick two (not necessarily disjoint)  $q$ -tuples  $w, w'$  of vertices from  $[n]$  simultaneously:  $w$  is picked according to the piecewise canonical distribution among all elements of  $G$ , whereas  $w'$  is picked according to the  $t$ -simulated piecewise distribution, considering  $\{i_1, \dots, i_N\}$  as the output of the first step – the *uniform sampling* step – of the simulated process. The events  $A$  and  $B$  are defined as follows.  $A$  holds if and only if either  $C$  doesn't hold or some entry of  $w$  is picked from  $I = \bigcup_{i=1}^{k-1} I_j$ , where  $I_j = \{i_{r_j^-}, i_{r_j^-} + 1, \dots, i_{r_j^+}\}$  for any  $j$ .  $B$  holds if and only if either  $C$  doesn't hold or some entry of  $w'$  is taken from  $I' = \bigcup_{j=1}^{k-1} I'_j$ , where  $I'_j = \{i_{r_j^-}, i_{r_j^-} + 1, \dots, i_{r_j^+}\}$  for any  $j$ .

**$A$  and  $B$  satisfy the requirements.** The major observation here is that the distribution of the piecewise canonical distribution under the assumption that  $A$  does not hold is identical to the distribution of the simulated process under the assumption that  $B$  does not hold. That is,  $\Pr_{\text{piece}}(E_{l_1, \dots, l_q} | \neg A) = \Pr_{\text{sim}}(E_{l_1, \dots, l_q} | \neg B)$  for any possible choice of  $l_1 < \dots < l_q$ , as required above. To see this, observe that under these assumptions, both distributions pick exactly  $q_j$  entries, uniformly at random, from the set  $\{i_{r_j^+} + 1, \dots, i_{r_{j+1}^-} - 1\}$  for any  $0 \leq j \leq k-1$  (where we define  $r_0^+ = 0$  and  $r_k^- = n + 1$ ). It remains to show that  $\Pr_{\text{piece}}(A) \leq \delta$  and  $\Pr_{\text{sim}}(B) \leq \delta$ .

In the piecewise-canonical distribution, every entry has probability at most  $q/n$  to be picked. Assuming that  $C$  holds, we get that  $|I_j| < 6\alpha n$  for any  $j$ , and so  $|I| \leq 6\alpha kn$ . Therefore,  $\Pr_{\text{piece}}(A|C) \leq |I|q/n < 6\alpha kq < 3\delta/4$ . Thus,  $\Pr_{\text{piece}}(A) \leq \Pr_{\text{piece}}(A|C) + \Pr(\neg C) < \delta$ , as needed.

In the simulated distribution, the probability that any given element from  $I'$  is taken to  $w'$  is at most  $q/t$ . Since  $|I'_j| < 4\alpha N$ , we get that  $|I'| < 4\alpha kN$  and so  $\Pr_{\text{sim}}(B) \leq |I'|q/t + \Pr(C) < 4\alpha k^2q + \delta/4 < \delta$ , as desired. ◀

► **Lemma 31.** *A piecewise-testable property has a canonical test. Moreover, if the number of parts of the piecewise-canonical  $\epsilon$ -test, denoted by  $k(\epsilon)$ , and its number of vertex queries, denoted by  $q(\epsilon)$ , are polynomial in  $\epsilon$ , then so is the number of queries of the canonical test.*

**Proof.** Let  $\mathcal{P}$  be a piecewise-testable property. Following Remark 5, for any  $\epsilon > 0$  there exists a deterministic piecewise-canonical  $\epsilon$ -test  $T$ , with confidence  $3/4$ , making exactly  $q$  queries on  $k$  parts. To simulate  $T$  using a canonical test  $T'$ , we pick  $\delta = 1/12$  and take

$t = t(\delta, k, q)$  as provided by Lemma 30 (here we also implicitly assume that  $n > N(\delta, k, q)$ ).  $T'$  is taken as the  $t$ -simulated piecewise test, that queries the induced subgraph  $H$  on  $kt$  vertices picked uniformly at random, and then imitates  $T$ : If, for any  $1 \leq i \leq k$ ,  $T$  chooses  $q_i$  vertices in part number  $i$ , then  $T'$  chooses  $q$  vertices of  $H$  using a  $t$ -simulated piecewise distribution, where  $q_i$  vertices are taken from the  $i$ -th simulated block. Then,  $T'$  makes the same decision that  $T$  would have made on the queried subgraph induced on the chosen  $q$  vertices.

By Lemma 30, the distributions  $\eta$  and  $\eta'$  over  $q$ -tuples of vertices generated by  $T$  and  $T'$ , respectively, are  $\delta$ -close. Let  $\mathcal{H}$  be a family of ordered graphs on  $q$  vertices such that  $T$  accepts its queried induced subgraph  $H$  if and only if  $H \in \mathcal{H}$ . Then,  $\Pr_\eta(H \in \mathcal{H}) \geq 3/4$  if the input graph  $G$  satisfies  $\mathcal{P}$ , whereas  $\Pr_\eta(H \in \mathcal{H}) < 1/4$  if  $G$  is  $\epsilon$ -far from  $\mathcal{P}$ . By Lemma 17, if the input graph  $G$  for  $T'$  satisfies  $\mathcal{P}$  then the queried induced subgraph  $H$  satisfies  $\Pr_{\eta'}(H \in \mathcal{H}) \geq 3/4 - \delta = 2/3$ , and if  $G$  is  $\epsilon$ -far from  $\mathcal{P}$ , then  $\Pr_{\eta'}(H \in \mathcal{H}) < 1/4 + \delta = 1/3$ . Thus,  $T'$  is a valid test for  $\mathcal{P}$ .  $\blacktriangleleft$

Lemmas 28 and 31 together prove the first (and more difficult) direction of Theorem 11.

## 7 Canonical testability to estimability

This section describes the proof of Theorem 3. The proof takes roughly the same steps as in the proof of Fischer and Newman [22] for the unordered case. For the proof of [22] to work in our case, we only need to make a few slight modifications. Therefore, instead of rewriting the whole proof, we only describe what modifications are made and how they change the proof.

The proof in [22] builds on a test for partition parameters, established in the seminal paper of Goldreich, Goldwasser and Ron [24]. The test of [24] also needs to be slightly modified for our needs. Therefore, the partition test receives the same treatment as the proof in [22]: We describe the modified statement and how to change the proof accordingly, but do not get into unnecessary technicalities.

### 7.1 The unordered proof

First we sketch the proof that canonical testability in unordered graphs implies estimability [22].

#### 7.1.1 Signatures of regular partitions and approximating the $q$ -statistic

A  $(\gamma, \epsilon)$ -signature for an equipartition  $\mathcal{A} = \{V_1, \dots, V_t\}$  is a sequence of densities  $\eta_{i,j}$ , such that the density between  $V_i$  and  $V_j$  differs from  $\eta_{i,j}$  by at most  $\gamma$ , for all but at most  $\epsilon \binom{t}{2}$  of the pairs  $i, j$ . The (labeled)  $q$ -statistic of a graph is the distribution of the labeled graphs on  $q$  vertices in it. Given a signature as above, it is natural to define the perceived  $q$ -statistic of the signature as the distribution on labeled  $q$ -vertex graphs generated as follows: First we choose  $q$  indices  $i_1, \dots, i_q$  from  $[t]$ . Then for every  $j < j'$  we add an edge between  $v_j$  and  $v_{j'}$  with probability  $\eta_{i_j, i_{j'}}$ , independently. The main observation in this part is that the perceived  $q$ -statistic of a signature with good (small) enough parameters of a regular enough partition of a graph  $G$  is close to the actual  $q$ -statistic of  $G$ . Thus, to estimate the  $q$ -statistic of a graph we just need to obtain a good signature of a regular partition of this graph. For more details, see Section 4 in [22].



### 7.1.2 Computing signature of a final partition

Implicit in the proof of the celebrated Szemerédi regularity lemma [35] is the concept of an *index* of an equipartition, which is a convex function of partitions that never decreases under taking refinements of a partition. A partition  $P$  is *robust* if, for any refinement  $Q$  of  $P$  that is not too large (in terms of the number of parts) with respect to  $P$ , the index of  $Q$  is similar to that of  $P$ . The main argument in [35] is that robustness implies regularity. An even stronger condition, that implies robustness, is finality. A partition  $P$  is *final* if for *any* partition  $Q^5$  whose number of parts is not much larger than that of  $P$ , the index of  $Q$  is also not much larger than that of  $P$ . It is easy to prove that robust and final partitions with arbitrarily good parameters exist. The definitions appear in Section 4 of [22], while the rest of the discussion here appears in Section 5 there.

Knowing the parameters of a good signature of a robust enough partition is useful for estimation, as we shall see soon. Before doing so, we explain how to find such a signature using the partition parameters test of [24]. This test is described in a more formal and detailed fashion in Subsection 7.3, but for our purposes, it acts as a test for the property of “having a given signature”. We consider a quantized set of signatures, which contains only a constant number of possible signatures, so that every graph is close to a graph satisfying one of the signatures (i.e., an  $\eta$ -net for a suitable parameter  $\eta$ ).

By applying the test of [24] to each of the signatures sufficiently many times and accepting or rejecting each of the signatures according to majority vote, we determine with good probability which signatures our input graph  $G$  is close to having. More precisely, all signatures that are very close to some actual signature  $S$  of  $G$  are accepted, and all of those that are very far from any actual signature  $S$  are rejected. Thus, this process only accepts signatures that are at the very least “quite close” to some actual one.

Finally, an index measure can also be defined for signatures, and the index of a good signature is close to that of the corresponding partition. Under the assumption that all signatures that we captured are quite close to an actual one, in particular we will find a good approximation of a final partition, and will recognize that it is final by not finding signatures of partitions that are only somewhat bigger and have a much bigger index (meaning that such partitions do not exist).

### 7.1.3 Knowing signature of a robust partition implies estimation

Note that for  $\delta > 0$  and a family  $\mathcal{H}$  of  $q$ -vertex graphs, having only a good signature  $S$  of a robust enough partition allows us to distinguish for any  $\epsilon > 0$ , deterministically, between the case that  $G$  is  $(\epsilon - \delta)$ -close to a graph  $G'$  that contains a large number of copies of labeled graphs from  $\mathcal{H}$ , and the case that all graphs that are  $\epsilon$ -close to  $G$  contain only a small number of  $\mathcal{H}$ -copies. Combining this statement with the one from the previous subsection, stating that computing the signature of some robust (and in particular, final) partition is possible with good probability in constant time, it is straightforward to conclude that any testable graph property is estimable. As the proof of this statement is rather technical and the main arguments do not change when moving to the ordered case, we do not go into the details of the proof here. Section 6 in [22] is dedicated to this proof.

---

<sup>5</sup> Here  $Q$  is not necessarily a refinement of  $P$

## 7.2 Adapting to the ordered setting

Suppose that a property  $\mathcal{P}$  has a canonical test making  $q$  queries. Using the proof for the undirected case as is will not work here. The reason is that, theoretically, a pair of vertex sets can be regular as an unordered pair, but interleaved in a way that makes it useless when we are interested in understanding the ordered  $q$ -statistic of a graph. Another issue that needs to be considered is the fact that we work here with edge-colored graphs, instead of standard ones. However, the latter is not a real issue: As observed in previous works [2, 7, 8], regularity-based arguments tend to generalize in a straightforward manner to the multicolored setting.

To accommodate for the first issue, we need a “regularity scheme” that is slightly different from the unordered instance. At the base of the scheme lies a  $k$ -interval equipartition  $I$  for a suitable  $k$ , which is known in advance. The regular, robust or final partitions that we need along the proof (analogously to the unordered case) are always refinements of the interval equipartition  $I$ , where we do not care about the relation between two parts that lie inside the same interval. Here, for partitions  $P$  and  $Q$ , we say that  $Q$  is a *refinement* of  $P$  if any part of  $Q$  is completely contained in a part of  $P$ . A formal presentation of the scheme is given in the next few definitions and lemmas. The first definition presents the  $(q, k)$ -statistic of a graph, which in some sense is the  $k$ -partite version of the  $q$ -statistic, as defined in Section 4.

► **Definition 32.** Let  $G, H$  be  $\Sigma$ -edge-colored ordered graphs on  $n \geq q$  vertices respectively, and let  $I = I_k(G) = (I_1, \dots, I_k)$  be the  $k$ -interval equipartition of  $G$  for  $k \geq q$ . A  $q$ -vertex induced subgraph of  $G$  is  $k$ -separated if, for every  $1 \leq i \leq k$ , no two vertices of the subgraph lie in  $I_i$ . The total number of  $k$ -separated subgraphs on  $q$  vertices in a graph on  $n$  vertices is denoted by  $N(k, q, n)$ . The number of  $k$ -separated  $H$ -copies in  $G$  is denoted by  $h_k(H, G)$ . The  $k$ -density of  $H$  in  $G$  is  $t_k(H, G) = h_k(H, G)/N(k, q, n)$ . Finally, the  $(q, k)$ -statistic of  $G$  is the vector  $(t_k(H, G))_{H \in \mathcal{H}_q}$ , where  $\mathcal{H}_q$  is the family of all  $\Sigma$ -edge-colored ordered graphs with  $q$  vertices.

► **Observation 33.** *The variation distance between the  $q$ -statistic and the  $(q, k)$ -statistic of a graph is at most  $q^2/2k$ .*

**Proof.** For a uniformly chosen pair  $(u, v)$  of disjoint vertices in a graph  $G$ , the probability that  $v$  lies in the same interval as  $u$  is at most  $\frac{n/k}{n-1}$ . By a union bound, the probability that a uniformly random  $q$ -tuple  $Q$  of disjoint vertices contains two vertices in the same interval is at most

$$\frac{n}{k(n-1)} \binom{q}{2} \leq \frac{q}{k(q-1)} \binom{q}{2} = \frac{q^2}{2k}.$$

Conditioning on the above not happening, the induced subgraph  $G[Q]$  is distributed according to the  $(q, k)$ -statistic. The statement of the lemma thus follows from Lemma 17. ◀

The next definition presents the  $k$ -partite notion analogous to canonical testability.

► **Definition 34.** A property  $\mathcal{P}$  of  $\Sigma$ -edge-colored ordered graphs is  $(\epsilon, q, k)$ -canonical if there exists a set  $\mathcal{A}$  of  $q$ -vertex  $\Sigma$ -edge-colored ordered graphs satisfying the following two conditions.

- If an ordered graph  $G$  satisfies  $\mathcal{P}$ , then  $\sum_{H \in \mathcal{A}} t_k(H, G) \geq 2/3$ . In this case we say that  $G$  is  $\mathcal{A}$ -positive.
- If  $G$  is  $\epsilon$ -far from satisfying  $\mathcal{P}$ , then  $\sum_{H \in \mathcal{A}} t_k(H, G) \leq 1/3$ . Here  $G$  is  $\mathcal{A}$ -negative.



Note that there may be graphs that are neither positive nor negative with respect to  $\mathcal{A}$  in the above definition. As it turns out, canonical  $\epsilon$ -testability implies  $(\epsilon, q, k)$ -canonicity for a suitable  $q$  and any  $k = \Omega(q^2)$ . In fact, the converse is also true, but is not needed for our proof.

► **Lemma 35.** *If a property  $\mathcal{P}$  of edge-colored ordered graphs is canonically testable, then there exists a function  $q : (0, 1) \rightarrow \mathbb{N}$  so that  $\mathcal{P}$  is  $(\epsilon, q(\epsilon), k)$ -canonical for any  $\epsilon > 0$  and  $k \geq 4q(\epsilon)^2$ .*

**Proof.** By Remark 5, if  $\mathcal{P}$  is canonically testable then for any  $\epsilon > 0$  it has a canonical  $\epsilon$ -test with confidence  $11/12$ , making  $q = q(\epsilon)$  queries. This means that there is a family  $\mathcal{A}$  of  $q$ -vertex graphs, such that  $\sum_{H \in \mathcal{A}} t(H, G) \geq 11/12$  for graphs  $G$  satisfying  $\mathcal{P}$  and  $\sum_{H \in \mathcal{A}} t(H, G) \leq 1/12$  for graphs that are  $\epsilon$ -far from  $\mathcal{P}$ . By Observation 33,  $\sum_{H \in \mathcal{A}} |t(H, G) - t_k(H, G)| \leq 2 \frac{q^2}{2k} \leq 1/4$ , and the statement follows. ◀

The definition of a regular partition needed for our case is given below. Here, the partition must refine the base interval equipartition, and we do not care how parts inside the same interval interact between themselves. For a single pair of parts lying in different intervals, the notion of regularity that we use is the standard multicolored notion, defined in Subsection 2.5.

► **Definition 36** ( *$k$ -refinement,  $(\gamma, k)$ -regular partition,  $(\gamma, \epsilon, k)$ -signature*). Let  $G$  be an  $\Sigma$ -edge-colored ordered graph, and let  $I = (I_1, \dots, I_k)$  be the  $k$ -interval equipartition of  $G$ . An equipartition  $P = (V_{11}, \dots, V_{1r}, \dots, V_{k1}, \dots, V_{kr})$  is a  $k$ -refinement if  $V_{ij} \subseteq I_i$  for any  $i, j$ .  $P$  is  $(\gamma, k)$ -regular if it is a  $k$ -refinement and all but a  $\gamma$ -fraction of the pairs  $(V_{ij}, V_{i'j'})$  with  $i < i'$  are  $\gamma$ -regular.

A  $(\gamma, \epsilon, k)$ -signature of  $P$  is a sequence  $S = (\eta_{ij}^{i'j'}(\sigma))$  for  $i < i' \in [k]$ ,  $j, j' \in [r]$ ,  $\sigma \in \Sigma$ , such that for all but at an  $\epsilon$ -fraction of the pairs  $(V_{ij}, V_{i'j'})$  with  $i < i'$ , we have  $|d_\sigma(V_{ij}, V_{i'j'}) - \eta_{ij}^{i'j'}(\sigma)| \leq \gamma$  for any  $\sigma \in \Sigma$ . A  $(\gamma, \epsilon, k)$ -signature is also referred to as a  $(\gamma, k)$ -signature.

In the above definition,  $d_\sigma(U, V)$  is the density of the color  $\sigma$  among edges between  $U$  and  $V$ . The *perceived  $(q, k)$ -statistic* is the natural translation of the notion of the perceived  $q$ -statistic from Definition 7 in [22] to our  $k$ -partite setting: It captures the “expected” fractions of each of the graphs on  $q$  vertices among the  $k$ -separated  $q$ -vertex subgraphs of  $G$ .  $(f, \gamma, k)$ -Robust and  $(f, \gamma, k)$ -final partitions (see Section 4 in [22] for the original unordered definitions) are also defined with respect to the  $k$ -partite structure, where we do not care about the relation between pairs of parts from the same interval. To accommodate the fact that we consider multicolored graphs, the *index* of a pair  $U, V$  is  $\sum_{\sigma \in \Sigma} d_\sigma(U, V)^2$  (compared to  $d(U, V)^2$  in the case of standard graphs). The index of an equipartition refining an interval partition is the sum of indices of all pairs not coming from the same interval, divided by the total number of such pairs.

After providing the definitions required for our ordered setting, the main statements of the proof, analogous to Lemmas 3.8, 4.4 and 4.5 in [22], are the following.

► **Lemma 37** (Ordered analogue of Lemma 3.8 in [22]). *For every  $q$  and  $\epsilon$  there exist  $\gamma$  and  $k$ , so that for every  $(\gamma, k)$ -regular partition  $P$  of  $G$  into  $t \geq k$  sets, where  $G$  has  $n \geq N(q, \epsilon, t)$  vertices, and for every  $(\gamma, k)$ -signature  $S$  of  $P$ , the variation distance between the actual  $(q, k)$ -statistic and the perceived  $(q, k)$ -statistic with respect to  $S$  is at most  $\epsilon$ .*

► **Lemma 38** (Ordered analogue of Lemma 4.4 in [22]). *For every  $k, \gamma$ , and  $f : \mathbb{N} \rightarrow \mathbb{N}$  there exist  $q, T$ , and an algorithm that makes up to  $q$  (piecewise-canonical) queries to any large*

enough graph  $G$ , computing with probability at least  $2/3$  a  $(\gamma, k)$ -signature of an  $(f, \gamma, k)$ -final partition of  $G$  into at most  $T$  sets.

Note that the second lemma requires piecewise-canonical vertex queries, making our algorithm a piecewise-canonical one. But Lemma 31 implies that this algorithm can be converted into a canonical one, since an algorithm that distinguishes between  $\delta$ -closeness to a property  $\mathcal{P}$  and  $\epsilon$ -farness from  $\mathcal{P}$ , for any  $\epsilon > \delta$ , is actually an  $(\epsilon - \delta)$ -test for being  $\delta$ -close to  $\mathcal{P}$ .

► **Lemma 39** (Ordered analogue of Lemma 4.5 in [22]). *For every  $q$  and  $\delta$  there exist  $\gamma, k$ , and  $f: \mathbb{N} \rightarrow \mathbb{N}$  with the following property. For every family  $\mathcal{H}$  of edge-colored ordered graphs with  $q$  vertices there exists a deterministic algorithm that receives as an input only a  $(\gamma, k)$ -signature  $S$  of an  $(f, \gamma, k)$ -robust partition with  $t \geq k$  sets of a graph  $G$  with  $n \geq N(q, \delta, t)$  vertices, and distinguishes given any  $\epsilon$  between the case that  $G$  is  $(\epsilon - \delta)$  close to some  $\mathcal{H}$ -positive graph, and the case that  $G$  is  $\epsilon$ -far from every graph that is not  $\mathcal{H}$ -negative.*

Once all definitions for our setting have been given, Lemma 35 brings us to a “starting point” from which the flow of the proof is essentially the same as in the unordered case, other than two issues mentioned and handled below. To avoid repeating the same ideas as in the unordered case, we will not provide the full technical details of the proofs of the three main lemmas. Deriving the proof of Theorem 3 from Lemmas 38 and 39 is similar to the unordered case.

One place where the move to a multicolored version requires more care is in proving the multicolored analogue of Lemma 6.2 in [22]. In the original proof, edges are being added/removed with a suitable probability, where the decision whether to modify an edge is independent of the other edges. In the multicolored version, the analogue of adding/removing edges is recoloring them. One way to do this is the following: for every color  $c$  where edges need to be added, we consider every relevant edge that has a “too dense” color  $c'$  and, with a suitable probability (that depends on the densities of the colors  $c, c'$  and the relevant signature), we recolor this edge from  $c'$  to  $c$ . By doing this iteratively for all colors that are in deficit, the multicolored analogue of Lemma 6.2 in [22] follows.

Another issue is that for our ordered setting, we need a “partition parameters” test that is slightly different than the one proved in [24] and used in [22]. We describe the modified partition parameters problem in Subsection 7.3.

### 7.3 The partition parameters test

Let  $\Phi = \{\rho_j^{LB}, \rho_j^{UB}\}_{j=1}^k \cup \{\varrho_{j,j'}^{LB}, \varrho_{j,j'}^{UB}\}_{j < j' \in \binom{[k]}{2}}$  be a set of nonnegative parameters so that  $\rho_j^{LB} \leq \rho_j^{UB}$  and  $\varrho_{j,j'}^{LB} \leq \varrho_{j,j'}^{UB}$ . An  $n$ -vertex graph  $G = (V, E)$  satisfies an (unordered)  $\Phi$ -instance if there is a partition  $V = V_1 \cup \dots \cup V_k \cup V'$  such that

- $0 \leq |V'| < k$  and  $|V| - |V'|$  is divisible by  $k$ .
- For any  $1 \leq j \leq k$ ,  $\rho_j^{LB} \lfloor n/k \rfloor \leq |V_j| \leq \rho_j^{UB} \lfloor n/k \rfloor$ .
- For any  $j < j' \in \binom{[k]}{2}$ ,  $\varrho_{j,j'}^{LB} \lfloor n/k \rfloor^2 \leq |E[V_j, V_{j'}]| \leq \varrho_{j,j'}^{UB} \lfloor n/k \rfloor^2$ .

In [24], it was shown that the property of having an unordered  $\Phi$ -instance is testable.

For our purposes, the base graph that we need to consider is an edge-colored  $r$ -partite graph, where the parts are of equal size (instead of a complete base graph, as in the unordered case). Formally, the partition parameters problem that we need to test is the following.

► **Definition 40** (Ordered  $\Phi$ -instance). An *ordered  $\Phi$ -instance* whose parameters are the positive integers  $r$  and  $k$  and the finite color set  $\Sigma$  consists of the following ingredients:

- For every  $i < i' \in [r]$  and  $j, j' \in [k]$  and every  $\sigma \in \Sigma$ , there are parameters  $\ell_{j,j'}^{i,i'}(\sigma) \leq h_{j,j'}^{i,i'}(\sigma)$ .
- For fixed  $i, i', j, j'$ , it holds that  $\sum_{\sigma \in \Sigma} \ell_{j,j'}^{i,i'}(\sigma) \leq 1 \leq \sum_{\sigma \in \Sigma} h_{j,j'}^{i,i'}(\sigma)$ .

Let  $G$  be an  $n$ -vertex  $\Sigma$ -edge-colored ordered graph, and denote its  $r$ -interval equipartition by  $I = (I_1, \dots, I_r)$ .  $G$  is said to *satisfy*  $\Phi$  if there exist disjoint sets of vertices  $V_{11}, \dots, V_{1k}, \dots, V_{r1}, \dots, V_{rk}$  such that for any  $i$  and  $j$ ,  $V_{ij} \subseteq I_i$  and  $|V_{ij}| = \lfloor n/rk \rfloor$ , and  $\ell_{j,j'}^{i,i'}(\sigma) \leq d_\sigma(V_{ij}) \leq h_{j,j'}^{i,i'}(\sigma)$  for any  $i < i' \in [r]$ ,  $j, j' \in [k]$  and  $\sigma \in \Sigma$ .

Recall that  $d_\sigma(A, B)$  is the *density* function of the color  $\sigma$  between the sets  $A$  and  $B$ . Note that while in the original unordered  $\Phi$ -instance, one could also specify lower and upper bounds on the number of vertices in each part, in our case it is not needed; for us it suffices to consider the special case where the size of each part is a  $1/rk$ -fraction of the total number of vertices.

► **Lemma 41.** *The edge-colored ordered graph property of satisfying an ordered  $\Phi$ -instance is testable.*

The proof is very similar to that of the unordered case in [24]. We first explain the main ideas of the proof in [24], and then describe what modifications are needed for our case.

### A sketch of the proof of Goldreich, Goldwasser and Ron [24]

- The following observation is a key to the proof: Given a partition  $P = (P_1, \dots, P_k)$  of the set of vertices  $V$  and a set  $X$  which is small relatively to  $V$ , define the *neighborhood profile* of a vertex  $v \in X$  with respect to  $P, X$  as the (ordered) set of  $k$  densities of the edges from  $v$  to each of the parts  $P_j \setminus X$ . The observation is that if all vertices of  $X$  have approximately the same neighborhood profile, and if we redistribute the vertices of  $X$  among the sets  $P_1, \dots, P_k$  so that each set receives roughly the same amount of vertices it lost to  $X$ , then the amount of edges between every pair of sets  $P_i, P_j$  is roughly maintained.
- Generally we will deal with sets  $X$  containing vertices with different neighborhood profiles, and will need a way to cluster them according to their profiles, and then be able to use the above observation. For this, one needs an *oracle* that, given a vertex  $v$ , will determine efficiently and with good probability a good approximation of the neighborhood profile of  $v$ . Another related oracle that we need is one that efficiently approximates, for  $P_1, \dots, P_k$  and  $X$ , the “ $P_j$ -fraction” with respect to  $X$ , which determines what fraction of the vertices in  $X$  with a given neighborhood profile belong to each  $P_j \cap X$ .
- Using the oracles, it is shown that if a given graph satisfies a  $\Phi$ -instance, then the following process generates, with good probability, an explicit partition  $P_1^s, \dots, P_k^s$  that approximately satisfies  $\Phi$ . Assume for now that we start with a partition  $P_1^0, \dots, P_k^0$  that satisfies the  $\Phi$ -instance exactly. We partition all vertices of the graph into a large enough constant number of sets  $X_1, \dots, X_s$  of equal size. Now we do the following for  $i = 1, \dots, s$ : We take the elements of  $X_i$ , apply the oracles on them, accordingly approximate how many elements from  $X_i$  with a certain neighborhood profile came from each  $P_j^{i-1}$ , and then “shuffle”: Return the same amount of elements from  $X_i$  with this profile to  $P_j^{i-1}$ , to create the part  $P_j^i$  (the returned elements are chosen arbitrarily among those with the relevant profile, and in particular, are not necessarily the ones that were taken from  $P_j^{i-1}$ ).
- There are two problems with the above statement. First, we do not know in advance the partition that satisfies the desired  $\Phi$ -instance, and thus along the way the partition

$P_1, \dots, P_k$  is not known to us. Second, we still do not know how to simulate the oracles. The solution to both of these problems is a brute force one: For each  $X_i$  we pick a large enough constant size set  $U_i \subseteq V \setminus X_i$ , and then enumerate on all possible partitions of  $U_i$  into  $U_i \cap P_1^{i-1}, \dots, U_i \cap P_k^{i-1}$  and all (rounded) possible values of the  $P_j^{i-1}$ -fraction for each  $j = 1, \dots, k$  and all  $i$ . As it turns out, if there is a partition of  $G$  satisfying the  $\Phi$ -instance, then our brute force search will find a good approximation of  $t$  with good probability.

- To turn the partitioning algorithm into a test, the observation is that one does not need to apply the first oracle on every vertex in each  $X_i$  to determine its neighborhood profile. Instead, we only apply it for a constant-size  $S_i \subseteq X_i$  chosen at random. As it turns out, this process is almost as accurate as the partitioning process, and in particular, it is shown that if  $G$  has a  $\Phi$ -instance then the process will accept, with good probability, a set of parameters of a  $\Phi'$ -instance which is close to the  $\Phi$ -instance. On the other hand, if  $G$  is far from having such a  $\Phi$ -instance, then the process will reject, with good probability, all sets of parameters that are close to the  $\Phi$ -instance. This concludes the proof of [24].

### Adapting the proof to our case

- The first and minor issue that we have to deal with is the fact that our graphs are edge-colored, and not standard graphs as in [24]. To handle this, instead of considering the neighborhood profile of a vertex, we are interested in the *colored neighborhood profile* of a vertex  $v$ , which keeps, for any relevant part  $P_j^i$  and any color  $\sigma$ , the fraction of vertices  $u \in P_j^i$  for which  $vu$  is colored  $\sigma$ . The rest of the proof translates naturally, implying that with this modification, the proof of [24] also applies to edge-colored graphs.
- The second issue is that our desired partition that satisfies the  $\Phi$ -instance has to be a refinement of the interval partition  $I_1, \dots, I_r$  of the input graph, as opposed to the situation in [24]. This issue is also not hard to handle. A “shuffle” operation in the unordered case was the process of removing elements from  $P_j^{i-1}$  into  $X_i$ , and then returning other elements from  $X_i$  to create  $P_j^i$ . In our case we will have to make shuffles of elements separately within each  $I_i$ , since it is not allowed to move elements between different  $I_i$ 's. The rest of the analysis is essentially the same as in the proof of [24].
- For the analysis of the last bullet to hold, we need the ability to pick a vertex uniformly at random from a given predetermined part  $V_i$ . This means that our algorithm is a piecewise canonical one, but not necessarily canonical. However, the transformation from a piecewise canonical test to a canonical one, that was proved in Section 6, implies the canonical testability of our version of the partition problem.

## 8 Canonical testability versus regular reducibility

As in the previous section, first we describe how the equivalence between testability and regular reducibility is proved in the unordered case [4], and then detail the small changes required to prove the edge-colored ordered case, namely Theorem 4.

### 8.1 The unordered case

#### 8.1.1 Enhancing regularity efficiently

In Section 3 of [4], it is shown that if a pair of vertex sets  $A, B$  has density close to  $\eta$  and its regularity measure is very close to  $\gamma$ , then by making a small number of edge modifications (insertions/deletions), one can turn the pair  $A, B$  into a “perfect” one, that has density exactly  $\eta$  and is  $\gamma$ -regular. The proof has two main steps: In the first step, we take a “convex

combination” of  $G[A, B]$  with a random bipartite graph with density  $\eta$ . This process does not change significantly the density between  $A$  and  $B$ , but since a random graph is highly regular, the combination is slightly more regular than the original  $G[A, B]$ , and this is all we need. In the second step, we fix the density between  $A$  and  $B$  to be exactly  $\eta$ . This might very slightly hurt the regularity, but if in the first step we make  $G[A, B]$  a bit more regular, i.e.,  $\gamma'$ -regular for a suitable  $\gamma' < \gamma$ , then it will remain  $\gamma$ -regular even after the loss of regularity in the second step.

### 8.1.2 Canonical testability implies regular reducibility

The easier direction of the proof is to show that any canonically testable property is also regular reducible, as is shown in Section 4 of [4]. Recall that, as discussed in Subsection 7.1.1, a regular enough partition of a graph  $G$  provides a good approximation of the  $q$ -statistic of  $G$ . We consider a canonical test  $T$  with a small enough proximity parameter, making  $q$  vertex queries. Basically, our set of “accepting” regular instances (see Definition 2.6 in [4]) will be created as follows: Initially, we take an  $\epsilon$ -net of possible parameters of regular partitions: This is a constant size quantized collection of the possible parameters of regular partitions, that “represents” all possible choices of parameters (in the sense that any possible choice of parameters has a representative in the constant size collection that is very close to it). Among the representatives from the  $\epsilon$ -net, we choose as accepting only those choices of parameters that predict acceptance of the above canonical test with probability at least  $1/2$ . Now, if a graph  $G$  satisfies our property  $\mathcal{P}$ , then it is accepted with probability  $2/3$  by the canonical test, and thus a regular enough partition of  $G$  will be similar to some accepting regularity instance, making  $G$  very close to satisfying this instance. Conversely, if  $G$  is  $\epsilon$ -far from  $\mathcal{P}$ , then it must also be far from any graph  $G'$  satisfying the accepting instance – since, by our choice of the accepting regular instances as those that indicate acceptance of the canonical test, any such  $G'$  is accepted by the canonical test with probability that is larger than  $1/3$ , meaning that  $G'$  cannot be far from satisfying  $\mathcal{P}$  (and thus  $G$  cannot be too close to  $G'$ , otherwise it would be  $\epsilon$ -close to  $\mathcal{P}$ , a contradiction)

### 8.1.3 Sampling preserves regular partitions

In Section 5 of [4] it is shown that if we sample a constant size set  $S$  of vertices in a graph  $G$ , then with good probability the induced subgraph  $G[S]$  will have  $\gamma$ -regular partitions with the same structure and approximately the same parameters (up to small differences) as those of  $G$ . The proof builds on a weaker argument of the same type, proved in [19], which states that for a regular enough partition  $P$  of  $G$ , and a large enough sample  $S$ , with good probability  $S$  has a partition with roughly the same densities as these of  $P$ , and with regularity that is slightly worse than that of  $P$ .

### 8.1.4 Regular reducibility implies testability

Due to the fact that canonical testability implies estimability, as we have seen in Section 7, it is enough to show that satisfying a specific regularity instance is testable. To do so, we take a large enough sample  $S$  of vertices and determine all possible parameters of regular partitions of  $S$ . By Subsection 8.1.3, these are essentially also all possible parameters of regular partitions of  $G$ , up to a small error. By Subsection 8.1.1, this small error is not a problem, implying that we are able to determine (with good probability) whether  $G$  satisfies the regularity instance by checking if it is close to one of the regular partitions suggested by  $S$ .

## 8.2 Adapting the proof to the ordered case

First, we need to translate the results from Section 3 in [4] to the multicolored setting. The main lemma that we need here is the following.

► **Lemma 42** (Ordered analogue of Lemma 3.1 in [4]). *There exists a function  $f: \mathbb{N} \times (0, 1) \rightarrow \mathbb{N}$  such that for any  $0 < \delta \leq \gamma \leq 1$  and finite alphabet  $\Sigma$  the following holds: Suppose that  $(A, B)$  is a  $(\gamma + \delta)$ -regular pair of sets of vertices with density between  $\eta - \delta$  and  $\eta + \delta$  in a  $\Sigma$ -edge-colored graph, where  $|A| = |B| = m \geq m_0(\eta, \delta, |\Sigma|)$ . Then, it is possible to make at most  $\delta f(|\Sigma|, \gamma) m^2$  edge color modifications in  $G$ , turning  $(A, B)$  into a  $\gamma$ -regular pair with density precisely  $\eta$ .*

The proof of Lemma 42 is largely similar to that of Lemma 3.1 in [4]. The only places that require special attention in the translation of the proof are those with “coin flip” arguments, such as the one in the proof of Lemma 3.3 in [4]. Adapting this type of arguments to the multicolored case is done as described in Subsection 7.2. In the proof of Lemma 3.3, for example, the second coin flip needs to have  $|\Sigma|$  possible outcomes instead of two (where the probability to get a  $\sigma$  should correspond to the desired density  $\eta_\sigma$ ).

The corollary of Lemma 42 that is used in our proof is the following. Note that the notation in the following statement is largely borrowed from Definition 14.

► **Lemma 43** (Ordered analogue of Corollary 3.8 from [4]). *There exists a function  $\tau: \mathbb{N} \times (0, 1) \rightarrow (0, 1)$  for which the following holds. Let  $R$  be an ordered regularity instance as in Definition 14, with the parameter  $k$  in  $R$  being large enough (as a function of the other parameters). Suppose that for some  $\epsilon > 0$ , a  $\Sigma$ -edge-colored ordered graph  $G$  has an equipartition  $(V_{11}, \dots, V_{1k}, \dots, V_{r1}, \dots, V_{rk})$  which is an  $r$ -refinement, and satisfies  $|d_\sigma(V_{ij}, V_{i'j'}, \sigma) - \eta_{ij}^{i'j'}(\sigma)| \leq \epsilon \tau(|\Sigma|, \gamma)$  for all  $i < i' \in [r]$ ,  $j, j' \in [k]$ , and  $\sigma \in \Sigma$ , and whenever  $(i, j, i', j') \notin \bar{R}$ , the pair  $V_{ij}, V_{i'j'}$  is  $(\gamma + \epsilon \tau(|\Sigma|, \gamma))$ -regular. Then  $G$  is  $\epsilon$ -close to satisfying  $R$ .*

### Canonical testability to ordered regular reducibility

The next step is to show that any canonically testable ordered graph property is (ordered) regular reducible. Recall that, by Section 7, canonical testability implies  $(\epsilon, q(\epsilon), k)$ -canonicity for  $k$  large enough (with respect to  $q(\epsilon)$ ), so it is enough to show the following.

► **Lemma 44** (Ordered analogue of Lemma 4.1 in [4]). *If a property  $\mathcal{P}$  is  $(\epsilon, q(\epsilon), k)$ -canonical for any  $\epsilon$  and any  $k$  large enough with respect to  $q(\epsilon)$ , then it is ordered regular reducible.*

For the results of Section 4 in [4], we define the ordered multicolored analogues of Definitions 4.3 and 4.7 in [4] as follows. Note the following “notational glitch”:  $\sigma$  in our definition refers to an edge color, whereas in Definition 4.3 of [4] it plays a totally different role, as a permutation.

► **Definition 45.** Let  $H = (U, E_H)$  be a  $\Sigma$ -edge-colored ordered graph on  $h$  vertices  $u_1 < \dots < u_h$ , and let  $W = (U, E_w)$  be an (edge) weighted  $\Sigma$ -edge-colored ordered graph on  $h$ -vertices, where the weight of edge  $(u_i, u_j)$  is  $\eta_{ij}$ . Define  $IC(H, W) = \prod_{\sigma \in \Sigma} \prod_{u_i u_j \in E_H^{-1}(\sigma)} \eta_{ij}$ .

Let  $R$  be an ordered regularity instance (recall Definition 14). Define  $IC(H, R) = \sum_{W \in \mathcal{W}} IC(H, W)$ , where  $\mathcal{W}$  ranges over all  $q$ -vertex weighted  $\Sigma$ -edge-colored weighted graph of the following type. Pick  $q$  pairs  $(i_1, j_1), \dots, (i_q, j_q)$  with  $i_1 < \dots < i_q \in [r]$  and  $j_1, \dots, j_q \in [k]$ , and take  $W$  to be the graph in which the weight of color  $\sigma$  between vertices  $u_a < u_b$  is  $\eta_{i_a, j_a}^{i_b, j_b}(\sigma)$ .



With these definitions, it is straightforward to translate the results of Section 4 in [4] to our setting. Note that an analogue for Definition 4.5 in that section is not needed in our case, since there are no non-trivial automorphisms in an ordered graph. In the proof of Lemma 4.1 in [4], let  $\mathcal{A}$  be the family of edge-colored ordered graphs on  $q = q(\epsilon)$  vertices, promised to us through Definition 34 by the fact that our given property  $\mathcal{P}$  is  $(\epsilon, q, k)$ -canonical, for  $k$  that is sufficiently large. As in the unordered case, we take a (constant size) set  $\mathcal{I}$  of ordered regular instances, such that any possible regular instance has parameters that are very close to one of the instance in  $\mathcal{I}$ . Our chosen  $\mathcal{R}$  in Definition 15 will be as in the unordered case:  $\mathcal{R} = \{R \in \mathcal{I} : \sum_{H \in \mathcal{A}} IC(R, H) \geq 1/2\}$ . The rest of the proof goes as in the unordered case.

### Ordered regular reducibility to (piecewise) canonical testability

It follows from the definition of regular reducibility, similarly to the unordered case, that it is enough to show that the property  $\mathcal{P}$  of satisfying a given regularity instance is canonically testable (the easy proof of the analogous unordered statement appears in Section 6 of [4], and translates directly to our case). In fact, by Lemma 31, it is enough to show that  $\mathcal{P}$  is piecewise canonically testable. Indeed, the core of the proof of this statement in the unordered case is in the fact that for  $\gamma$ , a large enough (as a function of  $\gamma$ ) sample of a graph has, with good probability, essentially the same  $\gamma$ -regular equipartitions as the containing graph, up to a small error.

The definition of *similar* regular partitions in the ordered case (analogous to Definition 5.1 from [4]) is the same as in the unordered case, but it refers to  $(\gamma, k)$ -regular partitions, instead of the unordered  $\gamma$ -regular ones. The analogue of Lemma 5.2 in the ordered case is exactly the same, except that we require the sample  $Q$  to have exactly  $q$  vertices in each interval of the  $k$ -interval equipartition (note that this is doable using piecewise-canonical algorithms). The proofs from this section (including the proof of the weaker result from [19]), as well as the proof of Theorem 1 from Section 6, translate readily to the ordered case.

## 9 Discussion and open problems

The earthmover resilient properties showcase, among other phenomena, an interesting connection between visual properties of images and the regularity-based machinery that was previously used to investigate unordered graphs. We believe that further research on the characterization problem for ordered structures would be interesting. It might also be interesting to investigate such problems using distance functions that are not Hamming distance, as was done, e.g., in [12]. Finally we present two open questions.

### 9.1 Characterization of testable earthmover-resilient properties

In this work we provide a characterization of earthmover resilient tolerantly testable properties. Although using such tests might make more sense than using intolerant tests in the presence of noise in the input (a situation that is common in areas like image processing, that are related to image property testing), it would also be very interesting to provide a characterization of the *testable* earthmover resilient properties. In particular, does there exist an earthmover resilient property that is testable but not tolerantly testable? The only known example of a (non earthmover resilient) property that is testable but not tolerantly testable is the PCPP-based property of [20], and it will certainly be interesting to find more examples of properties that have this type of behavior.

## 9.2 Alternative classes of properties

The class of earthmover resilient properties captures properties that are global in nature, and it will be interesting to identify and analyze some other wide classes of properties. A natural candidate is the class of all local properties [9]. We also believe that it might be possible to find other interesting classes of visual properties.

---

### References

- 1 Noga Alon and Omri Ben-Eliezer. Efficient removal lemmas for matrices. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh Srinivas Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPICs*, pages 25:1–25:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.APPROX-RANDOM.2017.25.
- 2 Noga Alon, Omri Ben-Eliezer, and Eldar Fischer. Testing hereditary properties of ordered graphs and matrices. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 848–858. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.83.
- 3 Noga Alon, Eldar Fischer, and Ilan Newman. Efficient testing of bipartite graphs for forbidden induced subgraphs. *SIAM J. Comput.*, 37:959–976, 2007.
- 4 Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: It’s all about regularity. *SIAM J. Comput.*, 39(1):143–167, 2009. doi:10.1137/060667177.
- 5 Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM J. Comput.*, 37(6):1703–1727, 2008. doi:10.1137/06064888X.
- 6 Barry C. Arnold, N. Balakrishnan, and H. N. Nagaraja. *A First Course in Order Statistics (Classics in Applied Mathematics)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.
- 7 Tim Austin and Terence Tao. Testability and repair of hereditary hypergraph properties. *Random Struct. Algorithms*, 36:373–463, 2010.
- 8 Maria Axenovich and Ryan R. Martin. A version of Szemerédi’s regularity lemma for multicolored graphs and directed graphs that is suitable for induced graphs. *arXiv*, 1106:2871, 2011.
- 9 Omri Ben-Eliezer, Simon Korman, and Daniel Reichman. Deleting and testing forbidden patterns in multi-dimensional arrays. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPICs*, pages 9:1–9:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.ICALP.2017.9.
- 10 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. The power and limitations of uniform samples in testing properties of figures. In Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen, editors, *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016, December 13-15, 2016, Chennai, India*, volume 65 of *LIPICs*, pages 45:1–45:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.FSTTCS.2016.45.
- 11 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Tolerant testers of image properties. 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 90:1–90:14.



- Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.90.
- 12 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L<sub>p</sub>-testing. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 164–173. ACM, 2014. doi:10.1145/2591796.2591887.
  - 13 Eric Blais and Yuichi Yoshida. A characterization of constant-sample testable properties. *arXiv*, 1612:06016, 2016.
  - 14 Christian Borgs, Jennifer Chayes, László Lovász, Vera T. Sós, Balázs Szegedy, and Katalin Vesztegombi. Graph limits and parameter testing. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing, STOC '06*, pages 261–270, New York, NY, USA, 2006. ACM.
  - 15 Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k-Monotonicity. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:21, 2017.
  - 16 Xi Chen, Adam Freilich, Rocco A. Servedio, and Timothy Sun. Sample-based high-dimensional convexity testing. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh Srinivas Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 37:1–37:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.37.
  - 17 Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In Dorit S. Hochbaum, Klaus Jansen, José D. P. Rolim, and Alistair Sinclair, editors, *Randomization, Approximation, and Combinatorial Algorithms and Techniques, Third International Workshop on Randomization and Approximation Techniques in Computer Science, and Second International Workshop on Approximation Algorithms for Combinatorial Optimization Problems RANDOM-APPROX'99, Berkeley, CA, USA, August 8-11, 1999, Proceedings*, volume 1671 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 1999. doi:10.1007/978-3-540-48413-4\_10.
  - 18 Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000. doi:10.1006/jcss.1999.1692.
  - 19 Eldar Fischer. Testing graphs for colorability properties. *Random Struct. Algorithms*, 26(3):289–309, 2005. doi:10.1002/rsa.20037.
  - 20 Eldar Fischer and Lance Fortnow. Tolerant versus intolerant testing for boolean properties. *Theory of Computing*, 2(9):173–183, 2006. doi:10.4086/toc.2006.v002a009.
  - 21 Eldar Fischer and Ilan Newman. Testing of matrix-poset properties. *Combinatorica*, 27(3):293–327, 2007. doi:10.1007/s00493-007-2154-3.
  - 22 Eldar Fischer and Ilan Newman. Testing versus estimation of graph properties. *SIAM J. Comput.*, 37(2):482–501, 2007. doi:10.1137/060652324.
  - 23 Eldar Fischer and Eyal Rozenberg. Lower bounds for testing forbidden induced substructures in bipartite-graph-like combinatorial objects. In Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings*, volume 4627 of *Lecture Notes in Computer Science*, pages 464–478. Springer, 2007. doi:10.1007/978-3-540-74208-1\_34.
  - 24 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

- 25 Oded Goldreich and Madhu Sudan. Locally testable codes and pcps of almost-linear length. *J. ACM*, 53(4):558–655, 2006. doi:10.1145/1162349.1162351.
- 26 Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. *Random Struct. Algorithms*, 23(1):23–57, 2003. doi:10.1002/rsa.10078.
- 27 Carlos Hoppen, Yoshiharu Kohayakawa, Richard Lang, Hanno Lefmann, and Henrique Stagni. Estimating parameters associated with monotone properties. In Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, volume 60 of *LIPICs*, pages 35:1–35:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.APPROX-RANDOM.2016.35.
- 28 Carlos Hoppen, Yoshiharu Kohayakawa, Richard Lang, Hanno Lefmann, and Henrique Stagni. Estimating the distance to a hereditary graph property. *Electronic Notes in Discrete Mathematics*, 61:607–613, 2017. doi:10.1016/j.endm.2017.07.014.
- 29 Felix Joos, Jaehoon Kim, Daniela Kühn, and Deryk Osthus. A characterization of testable hypergraph properties. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 859–867. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.84.
- 30 László Lovász and Balázs Szegedy. Testing properties of graphs and functions. *Israel Journal of Mathematics*, 178:113–156, 2010.
- 31 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.
- 32 Sofya Raskhodnikova. Approximate testing of visual properties. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NJ, USA, August 24-26, 2003, Proceedings*, volume 2764 of *Lecture Notes in Computer Science*, pages 370–381. Springer, 2003. doi:10.1007/978-3-540-45198-3\_31.
- 33 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.
- 34 Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. doi:10.1023/A:1026543900054.
- 35 Endre Szemerédi. Regular partitions of graphs. In *Problèmes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, volume 260 of *Colloq. Internat. CNRS*, pages 399–401. CNRS, Paris, 1978.

## **A** Sparse boundary implies earthmover resilience

Here we present the proof of Theorem 10. Let  $\mathcal{I}$  be an  $n \times n$  black-white image with a  $c$ -sparse boundary (recall Definition 9). Without loss of generality, we may assume that all pixels in locations  $\{1, n\} \times [n] \cup [n] \times \{1, n\}$  are white; otherwise, we may replace  $c$  with  $c + 4$  and turn  $\mathcal{I}$  into an  $(n + 2) \times (n + 2)$  image by adding an “artificial” white boundary to the image. To prove Theorem 10, it is enough to show that for any image  $\mathcal{I}'$  that is the result of making at most  $\delta n^2$  basic moves on  $\mathcal{I}$ , the absolute Hamming distance between  $\mathcal{I}$  and  $\mathcal{I}'$  is  $O(c\sqrt{\delta n^2})$ .

The following lemma suggests that it is enough to prove a similar statement for an image  $\mathcal{J}$  of our choice that is close enough (in Hamming distance) to  $\mathcal{I}$ .

► **Lemma 46.** *Fix  $\alpha, \beta > 0$  and let  $\mathcal{I}, \mathcal{J}: [n] \times [n] \rightarrow \Sigma$ . Suppose that  $d_H(\mathcal{J}, \mathcal{J}') \leq \alpha$  for any  $\mathcal{J}': [n] \times [n] \rightarrow \Sigma$  that satisfies  $d_e(\mathcal{J}, \mathcal{J}') \leq \beta$ . Then  $d_H(\mathcal{I}, \mathcal{I}') \leq \alpha + 2d_H(\mathcal{I}, \mathcal{J})$  for any  $\mathcal{I}': [n] \times [n] \rightarrow \Sigma$  satisfying  $d_e(\mathcal{I}, \mathcal{I}') \leq \beta$ .*

**Proof.** Write  $\gamma = d_H(\mathcal{I}, \mathcal{J})$ . Consider any  $\mathcal{I}'$  satisfying  $d_e(\mathcal{I}, \mathcal{I}') \leq \beta$  and let  $\sigma$  be a minimal unordered isomorphism of  $n \times n$  images<sup>6</sup> that maps  $\mathcal{I}$  to  $\mathcal{I}'$ . By the minimality of  $\sigma$ , the image  $\mathcal{J}' = \sigma(\mathcal{J})$  satisfies  $d_e(\mathcal{J}, \mathcal{J}') \leq \beta$  and so  $d_H(\mathcal{J}, \mathcal{J}') \leq \alpha$ . On the other hand, we know that  $d_H(\mathcal{I}', \mathcal{J}') = d_H(\sigma(\mathcal{I}), \sigma(\mathcal{J})) = d_H(\mathcal{I}, \mathcal{J}) = \gamma$  where the least equality follows from the fact that Hamming distance between two images is preserved when applying the same unordered isomorphism on both of them. The triangle inequality for the Hamming distance implies that

$$d_H(\mathcal{I}, \mathcal{I}') \leq d_H(\mathcal{I}, \mathcal{J}) + d_H(\mathcal{J}, \mathcal{J}') + d_H(\mathcal{J}', \mathcal{I}') \leq \beta + 2\gamma$$

as desired. ◀

Indeed, Lemma 46 implies that in order to prove Theorem 10, it is enough to show that there exists some  $n \times n$  black-white image  $\mathcal{J}$  with  $d_H(\mathcal{I}, \mathcal{J}) = O(c\sqrt{\delta}n^2)$ , such that for any image  $\mathcal{J}'$  that is the result of making at most  $\delta n^2$  basic moves on  $\mathcal{J}$ , we have  $d_H(\mathcal{J}, \mathcal{J}') = O(c\sqrt{\delta}n^2)$ . In order to explain which  $\mathcal{J}$  to take (as a function of  $\mathcal{I}$ ), and proceed with the rest of the proof, we need several topological definitions. A *pixel*  $P = (i, j)$  in  $\mathcal{I}$  is represented by its location  $(i, j)$ , and its color (black/white) is denoted  $\mathcal{I}[P]$ . The *distance* between two pixels  $(i, j), (i', j') \in [n] \times [n]$  is defined as  $|(i, j) - (i', j')| = |i - i'| + |j - j'|$ ; these pixels are *neighbors* if the distance between them is 1. A *shape*  $\mathcal{S}$  in  $\mathcal{I}$  is a connected component (with respect to the neighborhood relation) of pixels with the same color. We call  $P^0 = (1, 1)$  the *outer pixel* of an image, and the shape  $S^0$  that contains it is called the *outer shape*. Note that, by our assumption, the outer shape of  $\mathcal{S}$  contains all pixels in  $(\{1, n\} \times [n]) \cup ([n] \times \{1, n\})$ .

A *path* between pixels  $P$  and  $P'$  is a tuple of (not necessarily disjoint) pixels  $P_1 = P, P_2, \dots, P_t = P'$  in  $\mathcal{I}$ , such that  $P_s$  and  $P_{s+1}$  are neighbors for any  $1 \leq s \leq t - 1$ . The *outer boundary*  $B(S)$  of a shape  $S \neq S^0$  is the set of all pixels  $P$  in  $S$  satisfying the following: there exists a path from  $P^0 = (1, 1)$  to  $P$  that does not intersect  $S \setminus \{P\}$ . Finally, a pixel  $P$  is *encircled* by a shape  $S$  if any path from  $(1, 1)$  to  $P$  intersects  $S$  (this includes all pixels  $P \in S$ ). If all pixels  $P$  encircled by  $S$  satisfy  $P \in S$ , we say that  $S$  is *full*.

Our first lemma states that if two neighboring pixels have different colors, than one of them lies in the outer boundary of its shape.

► **Lemma 47.** *Let  $P_1, P_2$  be two neighboring pixels, where  $P_1$  is black and lies in shape  $S_1$  and  $P_2$  is white and lies in  $S_2$ . Then either  $P_1 \in B(S_1)$  or  $P_2 \in B(S_2)$  (or both).*

**Proof.** If there exists a path from  $(1, 1)$  to a pixel  $P'_1$  in  $S_1$ , that does not intersect  $S_2$ , then  $P_2 \in B(S_2)$ . To see this, recall that  $S_1$  is connected (by definition of a shape) and thus there exists a path from  $P'_1$  to  $P_1$  that remains inside  $S_1$ . Concatenating the above two paths and adding  $P_2$  at the end implies that  $P_2 \in B(S_2)$ .

Otherwise, all paths from  $(1, 1)$  to any pixel in  $S_1$  intersect  $S_2$ . In particular, this implies that there exists a path from  $(1, 1)$  to some  $P'_2 \in S_2$  that does not intersect  $S_1$ . Symmetrically to the previous paragraph, we get that  $P_1 \in B(S_1)$ . ◀

<sup>6</sup> The formal definition is given for ordered graphs in Definition 18, but can be translated naturally to images using our standard representation of an image as an ordered graph.

define  $B(\mathcal{I})$  as the union of all outer boundaries  $B(S)$  where  $S$  ranges over all shapes in  $\mathcal{I}$  other than  $S^0$ . The next lemma follows immediately from Lemma 47 and the fact that  $\mathcal{I}$  is  $c$ -sparse.

► **Lemma 48.**  $|B(\mathcal{I})| \leq 4cn$ , where  $S$  ranges over all shapes in  $\mathcal{I}$  other than  $S^0$ .

The next lemma implies that shapes with a small boundary cannot encircle a large number of pixels. This will play a crucial role in the design of  $\mathcal{J}$ .

► **Lemma 49.** *The total number of pixels encircled by a shape  $S \neq S^0$  is at most  $|B(S)|^2$ .*

**Proof.** We may assume that  $S$  is full. Let  $r(S)$  denote the number of pairs of neighboring pixels  $(P, P')$  where  $P \in S$  and  $P' \notin S$ . Then  $r(S) \leq 4|B(S)|$ . Among all possible full shapes  $S$  with a given value of  $r(S)$ , an (axis-aligned) rectangle contains the biggest number of pixels. This follows by iterating the following simple type of arguments as long as possible: If  $(i, j)$  and  $(i + 1, j + 1)$  are pixels of  $S$  while  $(i, j + 1) \notin S$ , then adding  $(i, j + 1)$  to  $S$  yields a shape  $S'$  with more pixels than in  $S$ , that satisfies  $r(S') \leq r(S)$ .

Now note that the number of pixels in a rectangle  $S$  is bounded by  $r(S)^2/16 \leq |B(S)|^2$ . The bound is achieved if  $S$  is a square with side length  $r(S)/4$ . ◀

We pick  $\mathcal{J}$  using the following iterative process. Start with  $\mathcal{J} = \mathcal{I}$ , and as long as possible do the following: Take a shape  $S \neq S^0$  in  $\mathcal{J}$  with  $|B(S)| \leq \sqrt{\delta}n$ , and recolor all pixels encircled by  $S$  by the opposite color to that of  $S$ ; repeat. Each such iteration deletes all pixels of  $B(S)$  from  $B(\mathcal{J})$  (and does not add any new pixels to  $B(\mathcal{J})$ ), modifying at most  $|B(S)|^2$  pixels in  $\mathcal{J}$ , so by Lemmas 48 and 49, in the end of the process we have  $d_H(\mathcal{I}, \mathcal{J}) = (4cn/\sqrt{\delta}n) \cdot O(\delta n^2) = O(c\sqrt{\delta}n^2)$  as desired.

Consider any composition  $\sigma$  of at most  $\delta n^2$  basic moves on  $\mathcal{J}$ . The new location of any pixel  $P$  after the basic moves is denoted by  $\sigma(P)$ . To conclude the proof, we need to show that the number of pixels  $P$  for which  $\mathcal{J}[P] \neq \mathcal{J}[\sigma(P)]$  is  $O(c\sqrt{\delta}n^2)$ .

Define the *boundary distance* of a pixel  $P$  in  $\mathcal{J}$  as the minimal distance of  $P$  to a pixel from  $B(\mathcal{J})$ . Our next lemma states that  $\sigma$  can only change the color of a small number of pixels with large boundary distance.

► **Lemma 50.** *No more than  $O(\sqrt{\delta}n^2)$  pixels  $P$  in  $\mathcal{J}$  have boundary distance at least  $\sqrt{\delta}n$  and satisfy  $\mathcal{J}[P] \neq \mathcal{J}[\sigma(P)]$ .*

**Proof.** By Lemma 47, a pixel  $P$  with boundary distance  $d$  that satisfies  $\mathcal{J}[P] \neq \mathcal{J}[\sigma(P)]$  must either be contained in a row that was moved at least  $d/2$  times or a column that was moved at least  $d/2$  times by the basic moves of  $\sigma$ ; here we pick  $d = \sqrt{\delta}n$ . With  $\delta n^2$  basic moves, at most  $O(\sqrt{\delta}n)$  rows and columns can be moved  $\sqrt{\delta}n/2$  or more steps away from their original location. The total number of pixels in these rows and columns is  $O(\sqrt{\delta}n^2)$ , as desired. ◀

It remains to show that no more than  $O(c\sqrt{\delta}n^2)$  pixels in  $\mathcal{J}$  have boundary distance less than  $\sqrt{\delta}n$ . The following lemma serves as a first step towards this goal.

► **Lemma 51.** *Let  $S \neq S^0$  be a shape in  $\mathcal{J}$ . Then there exists a path  $\Gamma(S)$  (possibly with repetitions of pixels) of length  $O(|B(S)|)$ , that covers all pixels of  $B(S)$ .*

**Proof.** Consider an  $n \times n$  grid in  $\mathbb{R}^2$  where the pixel  $(i, j)$  is represented by the unit square whose four endpoints are  $\{i - 1, i\} \times \{j - 1, j\}$ . Since any shape  $S$  is connected (by definition) under the neighborhood relation, in this representation  $S$  is the interior of a closed curve consisting of at most  $4|B(S)|$  axis-parallel length-1 segments. Following the segments of this

curve in a clockwise fashion and recording all pixels in  $S$  that we see on our right (including pixels that we only visit their corner) constructs a path (possibly with repetitions) that contains only the pixels of  $B(S)$  and some of their neighbors; recall that each pixel in  $\mathcal{J}$  has at most four neighbors. Moreover, each pixel appears at most  $O(1)$  times in this path, and so the total length of the path is  $O(|B(S)|)$ . ◀

Finally, the next lemma allows us to conclude the proof.

► **Lemma 52.** *Let  $S \neq S^0$ . The number of pixels in  $\mathcal{J}$  of distance at most  $d$  to  $B(S)$  is  $O(d|B(S)| + d^2)$ .*

**Proof.** Take the path  $\Gamma(S)$  obtained in Lemma 51. For each pixel  $P \in \Gamma(S)$  let  $B_d(P) = \{P' \in [n] \times [n] : |P' - P| \leq d\}$  denote the  $d$ -ball around  $P$  in  $\mathcal{I}$ . Note that the set of all pixels of distance at most  $d$  to  $B(S)$  is contained in  $\cup_{P \in \Gamma(S)} B_d(P)$ . Trivially,  $B_d(P)$  contains at most  $d^2$  pixels for any  $P$ . Moreover, if  $P_1$  and  $P_2$  are neighbors, then  $|B_d(P_1) \setminus B_d(P_2)| \leq d$ . The statement now follows since  $\Gamma(S)$ , a path, is connected under the neighborhood relation, and is of length  $O(|B(S)|)$ . ◀

Recall that  $|B(\mathcal{J})| \leq |B(\mathcal{I})| \leq 4cn$  by Lemma 48. Since all shapes  $S \neq S^0$  in  $\mathcal{J}$  satisfy  $|B(S)| > \sqrt{\delta}n$ , the number of such shapes must be at most  $4c/\sqrt{\delta}$ . Lemma 52 implies that the total number of pixels of boundary distance at most  $d = \sqrt{\delta}n$  in  $\mathcal{J}$  is at most  $O(dcn + d^2c/\sqrt{\delta}) = O(c\sqrt{\delta}n^2)$ . Along with Lemma 50, this completes the proof of Theorem 10.



# New Hardness Results for the Permanent Using Linear Optics

Daniel Grier<sup>1</sup>

MIT, Cambridge, USA

[grierd@mit.edu](mailto:grierd@mit.edu)

Luke Schaeffer

MIT, Cambridge, USA

[lrs@mit.edu](mailto:lrs@mit.edu)

---

## Abstract

---

In 2011, Aaronson gave a striking proof, based on quantum linear optics, that the problem of computing the permanent of a matrix is  $\#P$ -hard. Aaronson's proof led naturally to hardness of approximation results for the permanent, and it was arguably simpler than Valiant's seminal proof of the same fact in 1979. Nevertheless, it did not show  $\#P$ -hardness of the permanent for any class of matrices which was not previously known. In this paper, we present a collection of *new* results about matrix permanents that are derived primarily via these linear optical techniques.

First, we show that the problem of computing the permanent of a real orthogonal matrix is  $\#P$ -hard. Much like Aaronson's original proof, this implies that even a multiplicative approximation remains  $\#P$ -hard to compute. The hardness result even translates to permanents of orthogonal matrices over the finite field  $\mathbb{F}_p$ , for  $p \neq 2, 3$ . Interestingly, this characterization is tight: in fields of characteristic 2, the permanent coincides with the determinant; in fields of characteristic 3, one can efficiently compute the permanent of an orthogonal matrix by a nontrivial result of Kogan.

Finally, we use more elementary arguments to prove  $\#P$ -hardness for the permanent of a positive semidefinite matrix. This result shows that certain probabilities of boson sampling experiments with thermal states are hard to compute exactly, despite the fact that they can be efficiently sampled by a classical computer.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Problems, reductions and completeness

**Keywords and phrases** Permanent, Linear optics,  $\#P$ -hardness, Orthogonal matrices

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.19

**Funding** Both authors were supported by the Vannevar Bush Faculty Fellowship from the US Department of Defense. Part of this research was completed while visiting UT Austin.

**Acknowledgements** We would like to thank Scott Aaronson for posing the question which led to this paper and for his comments on this paper. We would also like to thank Rio LaVigne, Michael Cohen, and Alex Lombardi for some key mathematical insights.

## 1 Introduction

The permanent of a matrix has been a central fixture in computer science ever since Valiant showed how it could efficiently encode the number of satisfying solutions to classic NP-complete constraint satisfaction problems [31]. His theory led to the formalization of many

---

<sup>1</sup> Supported by an NSF Graduate Research Fellowship under Grant No. 1122374.



## 19:2 Permanent Hardness from Linear Optics

counting classes in complexity theory, including #P. Indeed, the power of these counting classes was later demonstrated by Toda's celebrated theorem, which proved that every language in the polynomial hierarchy could be computed in polynomial-time with only a single call to a #P oracle [26].

Let us recall the definition of the matrix permanent. Suppose  $A = (a_{i,j})$  is an  $n \times n$  matrix over some field. The *permanent* of  $A$  is

$$\text{per}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^n a_{i,\sigma(i)}$$

where  $S_n$  is the group of permutations of  $\{1, 2, \dots, n\}$ . Compare this to the *determinant* of  $A$ :

$$\det(A) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n a_{i,\sigma(i)}.$$

Since we can compute the determinant in polynomial time (in fact, in  $\text{NC}^2$ ; see Berkowitz [7]), the apparent difference in complexity between the determinant and permanent comes down to the cancellation of terms in the determinant [23, 33].

In his original proof, Valiant [31] casts the permanent in a combinatorial light, in terms of a directed graph rather than as a polynomial. Imagine the matrix  $A$  encodes the adjacency matrix of a weighted graph with vertices labeled  $\{1, \dots, n\}$ . Each permutation  $\sigma$  on the vertices has a cycle decomposition, which partitions the vertices into a collection of cycles known as a *cycle cover*. The *weight* of a cycle cover is the product of the edge weights of the cycles (i.e.,  $\prod_{i=1}^n a_{i,\sigma(i)}$ ). Therefore, the permanent is the sum of the weights of all cycle covers of the graph. Equipped with this combinatorial interpretation of the permanent, Valiant constructs a graph by linking together different kinds of gadgets in such a way that some cycle covers correspond to solutions to a CNF formula, and the rest of the cycle covers cancel out.

Valiant's groundbreaking proof, while impressive, is fairly opaque and full of complicated gadgets. A subsequent proof by Ben-Dor and Halevi [6] simplified the construction, while still relying on the cycle cover interpretation of the permanent. In 2009, Rudolph [22] noticed an important connection between quantum circuits and matrix permanents—a version of a correspondence we will use often in this paper. Rudolph cast the cycle cover arguments of Valiant into more physics-friendly language, which culminated in a direct proof that the amplitudes of a certain class of universal quantum circuits were proportional to the permanent. Had he pointed out that one could embed #P-hard problems into the amplitudes of a quantum circuit, then this would have constituted a semi-quantum proof that the permanent is #P-hard. Finally, in 2011, Aaronson [2] (independently from Rudolph) gave a completely self-contained and quantum linear optical proof that the permanent is #P-hard.

One must then ask, what is gained from converting Valiant's combinatorial proof to Aaronson's linear optical one? One advantage is pragmatic—much of the difficulty of arguments based on cycle cover gadgets is offloaded onto central, well-known theorems in linear optics and quantum computation. In this paper, we show that the linear optical approach has an even more important role in analyzing permanents of matrices with a global group structure. Such properties can be very difficult to handle in the "cycle cover model." For instance, the matrices which arise from Valiant's construction may indeed be invertible, but this seems to be more accidental than intentional, and a proof of their invertibility appears nontrivial. Adapting such techniques to give hardness results for orthogonal matrices would be extraordinarily tedious. In contrast, using the linear optical framework, we give proofs of hardness for many such matrices.



This gives a clean example for which a quantum mechanical approach sheds light on a problem in classical theoretical computer science. To take another example of such a quantum-classical connection, Kuperberg [19] shows that computing certain values of the Jones polynomial to high accuracy is  $\#P$ -hard using  $\text{PostBQP} = \text{PP}$ , a well-known result of Aaronson [1]. For a more thorough treatment of this topic, see the survey on quantum proofs for classical theorems of Drucker and de Wolf [11].

## 1.1 Results

We refine Aaronson’s linear optical proof technique and show that it *can* provide new  $\#P$ -hardness results. First, let us formally define what we mean by  $\#P$ -hardness throughout this paper. We say that the *permanent is  $\#P$ -hard* for a class of matrices if all functions in  $\#P$  can be efficiently computed with single-call access to an oracle which computes permanents of matrices in that class. That is, the permanent is hard for a function class  $A$  if, given an oracle  $\mathcal{O}$  for the permanent,  $A \subseteq \text{FP}^{\mathcal{O}[1]}$ .

Our main result is a linear optical proof that the permanent of a real orthogonal matrix is  $\#P$ -hard. Consequently, the permanent of matrices in any of the classical Lie groups (e.g., invertible matrices, unitary matrices, symplectic matrices) is also  $\#P$ -hard.

Our approach also reveals a surprising connection between the hardness of the permanent of orthogonal matrices over finite fields and the characteristic of the field. First notice that in fields of characteristic 2, the permanent is equal to the determinant and is therefore efficiently computable. Over fields of characteristic 3, there exists an elaborate yet polynomial time algorithm of Kogan [18] that computes the (orthogonal) matrix permanent. We give the first explanation for why no equivalent algorithm was found for the remaining prime characteristics, establishing a sharp dichotomy theorem: for fields of characteristic 2 or 3 there is an efficient procedure to compute orthogonal matrix permanents, and for *all* other primes  $p$  there exists a finite field<sup>2</sup> of characteristic  $p$  for which the permanent of an orthogonal matrix (over that field) is as hard as counting the number of solutions to a CNF formula mod  $p$ .<sup>3</sup> Furthermore, there exist infinitely many primes for which computing the permanent of an orthogonal matrix over  $\mathbb{F}_p$  (i.e., modulo  $p$ ) is hard.

Finally, we give a polynomial interpolation argument showing that the permanent of a positive semidefinite matrix is  $\#P$ -hard. This has an interesting consequence due to a recent connection between matrix permanents and boson sampling experiments with thermal input states [10, 21]. In particular, the probability of a particular experimental outcome is proportional to a positive semidefinite matrix which depends on the temperatures of the thermal states. Our result implies that it is hard to compute such output probabilities exactly despite the fact that an efficient classical sampling algorithm exists [21].

## 1.2 Proof Outline

The main result concerning the  $\#P$ -hardness of real orthogonal permanents follows from three major steps:

1. Construct a quantum circuit (over qubits) with the following property: *If* you could compute the probability of measuring the all-zeros state after the circuit has been applied to the all-zeros state, then you could calculate some  $\#P$ -hard quantity. We must modify

<sup>2</sup> We prove that this field is  $\mathbb{F}_{p^4}$ , although in some cases  $\mathbb{F}_{p^2}$  or even  $\mathbb{F}_p$  will suffice. See Section 4 for more details.

<sup>3</sup> Formally, this language is complete for the class  $\text{Mod}_pP$ . By Toda’s theorem, we have that  $\text{PH} \subseteq \text{BPP}^{\text{Mod}_pP}$ . See Appendix B for a more precise exposition of such counting classes.

the original construction of Aaronson [2], so that all the gates used in this construction are real.

2. Use a modified version of the Knill, Laflamme, Milburn protocol [17] to construct a linear optical circuit which simulates the quantum circuit in the previous step. In particular, we modify the protocol to ensure that the linear optical circuit starts and ends with one photon in every mode. Notice that this is distinct from Aaronson’s approach [2] because we can no longer immediately use the dual-rail encoding of KLM. We build new postselected encoding and decoding gadgets to circumvent this problem.
3. Use a known connection (first pointed out by Caianiello [9]) between the transition amplitude of a linear optical circuit and the permanent of its underlying matrix. Because we paid special attention to the distribution of photons across the modes of our linear optical network in the previous step, the success probability of the linear optical circuit is exactly the permanent of the underlying transition matrix. It is then simple to work backwards from this permanent to calculate our original #P-hard quantity.

The paper is organized as follows. Section 2 gives a brief introduction to the linear optical framework and the relevant tools we use in this paper. In Section 3, we use this framework to show that the permanent of a real orthogonal matrix is #P-hard. A careful analysis in Section 4 (and Appendix D) extends these gadgets to finite fields.<sup>4</sup> Finally, in Section 5, we explore other matrix classes, culminating in a proof that the permanent of a real special orthogonal symplectic involution is #P-hard.

## 2 Linear Optics Primer

In this section we will introduce the so-called “boson sampling” model of quantum computation, which will make clear the connection between the dynamics of *noninteracting* bosons and the computation of matrix permanents [9, 29]. The most promising practical implementations of this model are based on linear optics and use photons controlled by optical elements such as beamsplitters. We will use the term “linear optics” throughout, although any type of indistinguishable bosons would have the same dynamics.

Let us first consider the dynamics of a single boson. At any point in time, it is in one of finitely many *modes*. As the system evolves, the particle moves from one of  $m$  initial modes to a superposition of  $m$  final modes according to a *transition matrix* of *amplitudes*. That is, there is an  $m \times m$  unitary transition matrix  $U \in \mathbb{C}^{m \times m}$ , where  $U_{ji}$  is the amplitude of a particle going from mode  $i$  to mode  $j$ .

The model becomes more complex when we consider a system of multiple particles evolving on the same modes according to the same transition matrix. Let us define states in our space of  $k$  bosons in what is called the Fock basis. A *Fock* state for a  $k$ -photon,  $m$ -mode system is of the form  $|s_1, s_2, \dots, s_m\rangle$  where  $s_i \geq 0$  is the number of bosons in the  $i$ th mode and  $\sum_{i=1}^m s_i = k$ . Therefore, the Hilbert space which is spanned by the Fock basis states  $\Phi_{m,k}$  has dimension  $\binom{k+m-1}{k}$ . Alternatively, one can think of  $\Phi_{m,k}$  as the symmetrized subspace of  $(\mathbb{C}^m)^{\otimes k}$ . For a full exposition of the Fock space in these terms see Appendix A.

Let  $\varphi$  be the transformation which lifts the unitary  $U$  to act on a multi-particle system. On a  $k$ -particle system,  $\varphi(U)$  is a linear transformation from  $\Phi_{m,k}$  to  $\Phi_{m,k}$ . Let  $|S\rangle = |s_1, s_2, \dots, s_m\rangle$  be the Fock state describing the starting state of the system, and let  $|T\rangle =$

<sup>4</sup> As is the case with Aaronson’s proof, our real orthogonal construction also leads naturally to hardness of approximation results, which we discuss in Appendix E.

$|t_1, t_2, \dots, t_m\rangle$  be the ending state. We have:

$$\langle T|\varphi(U)|S\rangle = \frac{\text{per}(U_{S,T})}{\sqrt{s_1! \dots s_m! t_1! \dots t_m!}}$$

where  $U_{S,T}$  is the matrix obtained by taking  $s_i$  copies of the  $i$ th row and  $t_i$  copies of the column  $i$  in  $U$  for all  $i \in \{1, 2, \dots, m\}$ . We will refer to this formula as the  $\varphi$ -transition formula.<sup>5</sup>

Notice that  $s_1 + \dots + s_m$  must equal  $t_1 + \dots + t_m$  in order for  $U_{S,T}$  to be square. This expresses the physical principle that photons are not created or destroyed in the experiment.

For example, suppose  $U$  is the Hadamard gate and that we wish to apply  $U$  to two modes each with a single photon. That is,  $U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  and  $|S\rangle = |1, 1\rangle$ . Since the number of photons must be conserved, the resulting state of the system is in some linear combination of  $|2, 0\rangle, |1, 1\rangle$ , and  $|0, 2\rangle$ . We calculate these amplitudes explicitly below:

$ T\rangle$	$ 2, 0\rangle$	$ 1, 1\rangle$	$ 0, 2\rangle$
$U_{S,T}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$
$\text{per}(U_{S,T})$	1	0	-1
$\langle T \varphi(U) S\rangle$	$1/\sqrt{2}$	0	$-1/\sqrt{2}$

Therefore, when we apply Hadamard in a linear optical circuit to the state  $|1, 1\rangle$  we get the state  $\frac{|2, 0\rangle - |0, 2\rangle}{\sqrt{2}}$ . Indeed, we have derived the famous Hong-Ou-Mandel effect—the photons are noninteracting, yet the final state is clearly highly entangled [15].

Finally, we note that  $\varphi$  expresses the fact that linear optical systems are reversible and can be composed together. This behavior is captured by the following theorem:

► **Theorem 1** (see Facts 19 and 20 in Appendix A). *The map  $\varphi$  is a group homomorphism. Furthermore, if  $U \in \mathbb{C}^{n \times n}$  is unitary, then  $\varphi(U)$  is unitary.*

We now state a landmark result in linear optics, which connects the dynamics of a linear optical system with those of a traditional quantum circuit over qubits. Define  $|I\rangle = |0, 1, \dots, 0, 1\rangle$ , the Fock state with a photon in every other mode.

► **Theorem 2** (Knill, Laflamme, and Milburn [17]). *Postselected linear optical circuits are universal for quantum computation. Formally, given a quantum circuit  $Q$  consisting of CSIGN<sup>6</sup> and single-qubit gates, there exists a linear optical network  $U$  constructible in polynomial time such that*

$$\langle I|\varphi(U)|I\rangle = \frac{1}{4^\Gamma} \langle 0 \dots 0|Q|0 \dots 0\rangle,$$

where  $\Gamma$  is the number of CSIGN gates in  $Q$ .

We will refer to the construction of the linear optical network  $U$  from  $Q$  in Theorem 2 as the *KLM protocol*. It will be helpful to give some idea of its proof here. First, each qubit of  $Q$  is encoded in two modes of  $U$  in the classic dual-rail encoding. That is, the qubit state  $|0\rangle$  is encoded by the Fock state  $|0, 1\rangle$  and the state  $|1\rangle$  is encoded by the Fock state  $|1, 0\rangle$ .

<sup>5</sup> Once again, we refer readers, especially non-physicists, to Appendix A for a description of the  $\varphi$ -transition formula in terms of linear operators on the space  $(\mathbb{C}^m)^{\otimes k}$ .

<sup>6</sup> The CSIGN gate, also often referred to as a controlled- $Z$  gate, is the two-qubit operation which applies a minus phase when both of its inputs are 1. That is,  $\text{CSIGN}|x_1 x_2\rangle = (-1)^{x_1 x_2} |x_1 x_2\rangle$  for  $x_1, x_2 \in \{0, 1\}$ . It is well-known that CSIGN and single-qubit gates are universal for quantum computation [20].

Now suppose  $G$  is a single-qubit gate in  $Q$ . Using the  $\varphi$ -transition formula, it is not hard to see that applying  $G$  to the corresponding pair of dual-rail modes in the linear optical circuit implements the correct single-qubit unitary. Applying a CSIGN gate is trickier. The KLM protocol builds the CSIGN gate from a simpler  $\text{NS}_1$  gate, which flips the sign of a single mode if it has 2 photons and does nothing when the mode has 0 or 1 photon. Using two  $\text{NS}_1$  gates one can construct a CSIGN gate (see Figure 5 in Appendix F).

Unfortunately, the  $\text{NS}_1$  gate cannot be implemented with a straightforward linear optical circuit. Therefore, some additional resource is required. The original KLM protocol uses *adaptive measurements*, that is, the ability to measure in the Fock basis in the middle of a linear optical computation and adjust the remaining sequence of gates if necessary. Intuitively, using adaptive measurements one can apply some transformation and then measure a subset of the modes to “check” if the  $\text{NS}_1$  gate was applied. For simplicity, however, we will assume we have a stronger resource—namely, *postselection*—so we can assume the measurements always yield the most convenient outcome. Putting the above parts together completes the proof Theorem 2.

### 3 Permanents of Real Orthogonal Matrices

The first class of matrices we consider are the real orthogonal matrices, that is, square matrices  $A \in \mathbb{R}^{n \times n}$  with  $AA^T = A^T A = I$ . This section is devoted to proving the following theorem, which forms the basis for many of the remaining results in this paper.

► **Theorem 3** (informal). *The permanent of a real orthogonal matrix is #P-hard.*

The orthogonal matrices form a group under composition, the *real orthogonal group*, usually denoted  $O(n, \mathbb{R})$ . This is a subgroup of the unitary group,  $U(n, \mathbb{C})$ , which is itself a subgroup of the general linear group  $GL(n, \mathbb{C})$ . Notice then that the hardness result of Theorem 3 will carry over to unitary matrices and invertible matrices.<sup>7</sup>

Our result follows the outline of Aaronson’s linear optical proof [2] that the permanent is #P-hard. In particular, our result depends on the KLM construction [17], and a subsequent improvement by Knill [16], which will happen to have several important properties for our reduction.

Let us briefly summarize Aaronson’s argument. Suppose we are given a classical circuit  $C$ , and wish to compute  $\Delta_C$ , the number of satisfying assignments minus the number of unsatisfying assignments. Clearly, calculating  $\Delta_C$  is a #P-hard problem. The first thing to notice is that there exists a simple quantum circuit  $Q$  such that the amplitude  $\langle 0 \cdots 0 | Q | 0 \cdots 0 \rangle$  is proportional to  $\Delta_C$ . The KLM protocol of Theorem 2 implies that there exists a postselected linear optical experiment simulating  $Q$ . This results in the following chain which relates  $\Delta_C$  to a permanent.

$$\text{per}(U_{I,I}) = \langle I | \varphi(U) | I \rangle \propto \langle 0 \cdots 0 | Q | 0 \cdots 0 \rangle \propto \Delta_C.$$

Notice that Aaronson’s result does not imply that the permanent of  $U \in U(n, \mathbb{C})$  is #P-hard since  $U_{I,I}$  is a *submatrix* of  $U$ . If, however,  $|S\rangle = |T\rangle = |1, \dots, 1\rangle$ , then  $U_{S,T} = U$  so the analogous chain relates  $\Delta_C$  directly to the permanent of  $U$ , which is a complex unitary matrix. In fact, this is exactly what we will arrange by modifying the KLM protocol. Furthermore, we will be careful to use *real* matrices exclusively during all gadget constructions, which will result in  $U$  being real, finishing the proof of Theorem 3.

<sup>7</sup> See Corollary 17 for a complete list of classical Lie groups for which our result generalizes.

In the following subsections, we will focus on the exact details of the reduction and emphasize those points where our construction differs from that of Aaronson.

### 3.1 Constructing the Quantum Circuit

Let  $C : \{0, 1\}^n \rightarrow \{0, 1\}$  be a classical Boolean circuit of polynomial size and let

$$\Delta_C := \sum_{x \in \{0, 1\}^n} (-1)^{C(x)}.$$

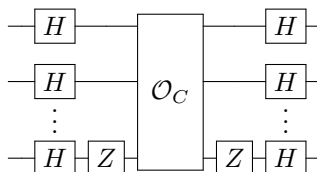
In this section, we prove the following:

► **Theorem 4.** *Given  $C$ , there exists a  $p(n)$ -qubit quantum circuit  $Q$  such that*

$$\langle 0|^{\otimes p(n)} Q | 0 \rangle^{\otimes p(n)} = \frac{\Delta_C}{2^n}$$

where  $p(n)$  is some polynomial in  $n$ . Furthermore,  $Q$  can be constructed in polynomial time with a polynomial number of real single-qubit gates and CSIGN gates.

To prove the theorem, it will suffice to implement  $\mathcal{O}_C$ , the standard oracle instantiation of  $C$  on  $n + 1$  qubits. That is,  $\mathcal{O}_C|x, b\rangle = |x, b \oplus C(x)\rangle$  for all  $x \in \{0, 1\}^n$  and  $b \in \{0, 1\}$ . The circuit for  $Q$  is depicted below, where  $H$  is the Hadamard gate and  $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$  is the Pauli  $\sigma_Z$  gate.



From this construction, we have

$$\langle 0|^{\otimes p(n)} Q | 0 \rangle^{\otimes p(n)} = \frac{1}{2^n} \left( \sum_{x \in \{0, 1\}^n} \langle x | \langle - | \right) \mathcal{O}_C \left( \sum_{x \in \{0, 1\}^n} |x\rangle |-\rangle \right) = \frac{\Delta_C}{2^n}.$$

Therefore, to complete the proof, it suffices to construct  $\mathcal{O}_C$  from CSIGN and single-qubit gates. For now let us assume we have access to the Toffoli gate as well. Since  $C$  is a classical Boolean function of polynomial complexity,  $\mathcal{O}_C$  can be implemented with a polynomial number of Toffoli and NOT gates<sup>8</sup> and a polynomial number of ancillas starting in the  $|0\rangle$  state [28].

Let us describe, briefly, one way to construct  $\mathcal{O}_C$ . Suppose we are given the circuit  $C$  as a network of polynomially many NAND gates. For each wire, with the exception of the input wires, we create an ancilla initially in state  $|0\rangle$  and use the NOT gate to put it in state  $|1\rangle$ . For each NAND gate (in topological ordering, i.e., such that no gate is applied before its inputs have been computed), we apply a Toffoli gate targeting the ancilla associated with the output wire, and controlled by the qubits associated with its input wires (whether they are the output of an earlier NAND gate, or an actual input). Hence, the target qubit is in state  $|1\rangle$  unless both control qubits are in state  $|1\rangle$ , simulating a NAND gate. Once we have

<sup>8</sup> Because we require that all ancillas start in the  $|0\rangle$  state, we also need the NOT gate to create  $|1\rangle$  ancillas.

applied all the gates of  $C$ , the output of the function will exist in the final ancilla register. We can now apply the same sequence of gates (ignoring the final Toffoli gate) in reverse order, which returns all other ancillas and inputs to their original value. This completes the construction.

Finally, we must construct the Toffoli gate from single-qubit gates and CSIGN gates. Unfortunately, Aaronson’s proof [2] uses a classic construction of the Toffoli gate which uses complex single-qubit gates (see, for example, Nielsen and Chuang [20]). This will later give rise to linear optical circuits with complex matrix representations as well.<sup>9</sup> Therefore, we will restrict ourselves to CSIGN and *real* single-qubit gates in our construction of the Toffoli gate.<sup>10</sup>

► **Lemma 5.** *There exists a circuit of CSIGN, Hadamard, and  $R_{\pi/4}$  gates which implements a Toffoli gate exactly, where*

$$R_{\pi/4} = \frac{1}{2} \begin{pmatrix} \sqrt{2 + \sqrt{2}} & -\sqrt{2 - \sqrt{2}} \\ \sqrt{2 - \sqrt{2}} & \sqrt{2 + \sqrt{2}} \end{pmatrix}.$$

We prove this lemma in Appendix C. This completes the proof of Theorem 4.

### 3.2 Postselected Linear Optical Gadgets

We will construct a postselected linear optical circuit  $L$  which will simulate the qubit circuit  $Q$  on the all zeros input via a modified version of the KLM protocol. The following chain of relations will hold:<sup>11</sup>

$$\text{per}(L) = \langle 1, \dots, 1 | \varphi(L) | 1, \dots, 1 \rangle \propto \langle 0 \cdots 0 | Q | 0 \cdots 0 \rangle \propto \Delta_C.$$

The first step was to convert from a classical circuit to a quantum circuit. Below we formalize the second step: converting from a quantum circuit to a linear optical circuit.

► **Theorem 6.** *Given an  $n$ -qubit quantum circuit  $Q$  with a polynomial number of CSIGN and real single-qubit gates, there exists a linear optical circuit  $L \in \mathcal{O}(4n + 2\Gamma, \mathbb{R})$  such that*

$$\langle 1, \dots, 1 | \varphi(L) | 1, \dots, 1 \rangle = \left( \frac{1}{3} \sqrt{\frac{2}{3}} \right)^\Gamma \left( \frac{-1}{\sqrt{6}} \right)^n \langle 0 |^{\otimes n} Q | 0 \rangle^{\otimes n},$$

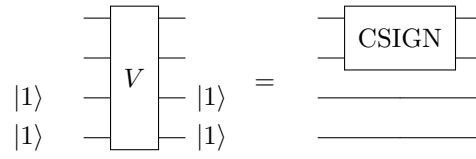
where  $\Gamma$  is the number of CSIGN gates in  $Q$ . Furthermore,  $L$  can be computed in time polynomial in  $n$ .

We now give an explicit construction of  $L$  using the original KLM protocol, subsequent improvements by Knill [16], and a new gadget unique to our problem. First, let us recall our main issue with using the original KLM protocol: to prove that *orthogonal* matrices are #P-hard, we must have that all modes start and end with exactly one photon. There

<sup>9</sup> Actually, the proof of Aaronson [2] claims that the final linear optical matrix consists entirely of real-valued entries even though the matrices of the individual single-qubit gates have complex entries. In fact, the matrix *does* have complex entries, but our construction for Toffoli suffices to fix this error.

<sup>10</sup> Although it is known that the Toffoli gate and the set of real single-qubit gates suffice to densely generate the orthogonal matrices (i.e.,  $\mathcal{O}(2^n)$  for every  $n > 0$ ) [24], it will turn out to be both simpler and necessary to have an exact decomposition. In particular, we will need an exact construction of the Toffoli gate in Section 4 where we discuss the computation of permanents in finite fields.

<sup>11</sup> To clarify,  $|0 \cdots 0\rangle$  is a tensor product of qubits in the state  $|0\rangle$  and  $|1, \dots, 1\rangle$  is a Fock state with 1 photon in every mode.



■ **Figure 1** Applying a postselected  $V$  gadget to generate CSIGN.

are two instances in which the original KLM protocol requires a mode to be empty at the beginning and end of the computation. First, the  $\text{NS}_1$  gate postselects on the Fock state  $|0, 1\rangle$ , and second, KLM protocol works in a dual-rail encoding. Therefore, half of the modes in the original KLM protocol start and end empty.

To overcome the first obstacle, we appeal to subsequent work of Knill [16], in which the  $\text{NS}_1$  gadget construction for CSIGN is replaced by a single 4-mode gadget  $V$ , which directly implements CSIGN with two modes postselected in state  $|1, 1\rangle$ . From the matrix gadget

$$V = \frac{1}{3\sqrt{2}} \begin{pmatrix} -\sqrt{2} & -2 & 2 & 2\sqrt{2} \\ 2 & -\sqrt{2} & -2\sqrt{2} & 2 \\ -\sqrt{6+2\sqrt{6}} & \sqrt{6-2\sqrt{6}} & -\sqrt{3+\sqrt{6}} & \sqrt{3-\sqrt{6}} \\ -\sqrt{6-2\sqrt{6}} & -\sqrt{6+2\sqrt{6}} & -\sqrt{3-\sqrt{6}} & -\sqrt{3+\sqrt{6}} \end{pmatrix}$$

we can directly calculate the transition amplitudes of the circuit:

$$\begin{aligned} \langle 0, 0, 1, 1 | \varphi(V) | 0, 0, 1, 1 \rangle &= \frac{1}{3} \sqrt{\frac{2}{3}} & \langle 0, 1, 1, 1 | \varphi(V) | 1, 0, 1, 1 \rangle &= 0 \\ \langle 0, 1, 1, 1 | \varphi(V) | 0, 1, 1, 1 \rangle &= \frac{1}{3} \sqrt{\frac{2}{3}} & \langle 1, 0, 1, 1 | \varphi(V) | 0, 1, 1, 1 \rangle &= 0 \\ \langle 1, 0, 1, 1 | \varphi(V) | 1, 0, 1, 1 \rangle &= \frac{1}{3} \sqrt{\frac{2}{3}} & \langle 2, 0, 1, 1 | \varphi(V) | 1, 1, 1, 1 \rangle &= 0 \\ \langle 1, 1, 1, 1 | \varphi(V) | 1, 1, 1, 1 \rangle &= -\frac{1}{3} \sqrt{\frac{2}{3}} & \langle 0, 2, 1, 1 | \varphi(V) | 1, 1, 1, 1 \rangle &= 0 \end{aligned}$$

We now argue that these transition amplitudes suffice to generate a postselected CSIGN. Consider the linear optical circuit depicted in Figure 1: the first two inputs of the  $V$  gadget are applied to the dual rail modes which contain a photon whenever their corresponding input qubits of the CSIGN gate are in state  $|1\rangle$ ; the next two modes are postselected in the  $|1, 1\rangle$  state. First, because we postselect on the final two modes ending in the state  $|1, 1\rangle$ , we only need to consider those transitions for which those two modes end in that state. Secondly, because we use “fresh” ancillary modes for every CSIGN gate, we can always assume that those two modes start in the  $|1, 1\rangle$  state. This already vastly reduces the number of cases we must consider.

Finally, we wish to know what will happen when the first two modes start in the states  $|0, 0\rangle$ ,  $|0, 1\rangle$ ,  $|1, 0\rangle$ , and  $|1, 1\rangle$ . Our construction will ensure that there is never more than one photon per mode representing one of the dual-rail encoded qubits. For instance, the transition amplitudes of  $V$  show that whenever the first two modes of the circuit each start with a photon, there is 0 probability (after postselection) that those photons transition to a state in which one of those modes contains 2 photons and the other contains no photons.

We find that all other amplitudes behave exactly as we would expect for CSIGN. Since each of the acceptable transitions (e.g. from the state  $|0, 1\rangle$  to the state  $|0, 1\rangle$ ) has equal magnitude, we only have left to check that  $V$  flips the sign of the state whenever the input modes are both in the  $|1\rangle$  state, which is indeed the case. Importantly, because  $\varphi$  is a homomorphism, we can analyze each such gate separately. Therefore, using the above we can now construct a linear optical circuit where all of our postselected modes for CSIGN start and end with exactly one photon.



## 19:10 Permanent Hardness from Linear Optics

We now turn our attention to the dual-rail encoding. Instead of changing the dual-rail encoding of the KLM protocol directly, we will start with one photon in every mode and apply a linear optical gadget to convert to a dual-rail encoding. Of course, the number of photons in the circuit must be conserved, so we will dump these extra photons into  $n$  modes separate from the modes of the dual-rail encoding. Specifically, each logical qubit is our scheme is initially represented by four modes in the state  $|1, 1, 1, 1\rangle$ . We construct a gadget that moves a photon from the first mode to the third mode, postselecting on a single photon in the last mode. That is, under postselection, we get the transition

$$|1, 1, 1, 1\rangle \rightarrow |0, 1, 2, 1\rangle,$$

where the first two modes now represent a logical  $|0\rangle$  qubit in the dual-rail encoding, the third mode stores the extra photon (which we will reclaim later), and the last mode is necessary for postselection. We call the gadget for this task the *encoding gadget*  $E$ , and it is applied to the first, third, and fourth mode of the above state. The matrix<sup>12</sup> for  $E$  is

$$E = \frac{1}{\sqrt{6}} \begin{pmatrix} \sqrt{2} & -\sqrt{2} & \sqrt{2} \\ 0 & \sqrt{3} & \sqrt{3} \\ -2 & -1 & 1 \end{pmatrix}$$

from which we get the following transition amplitudes

$$\langle 1, 1, 1 | \varphi(E) | 1, 1, 1 \rangle = 0, \quad \langle 2, 0, 1 | \varphi(E) | 1, 1, 1 \rangle = 0, \quad \langle 0, 2, 1 | \varphi(E) | 1, 1, 1 \rangle = \frac{1}{\sqrt{3}}.$$

After applying the encoding gadget to each logical qubit, we can implement the KLM protocol as previously discussed.<sup>13</sup> Therefore, the relevant amplitude in the computation of  $Q$  is now proportional to amplitude of the Fock state which has  $n$  groups of modes in the state  $|0, 1, 2, 1\rangle$  and  $2\Gamma$  modes in the state  $|1\rangle$ . Because we want to return to a state which has one photon in every mode, we must reverse the encoding step.<sup>14</sup> For this purpose, we construct a *decoding gadget*  $D$ , which will not require any extra postselected modes. We apply the gadget to the second and third modes of the logical qubit such that the two photons in the third mode split with some nonzero probability. The matrix for  $D$  is

$$D = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

from which the transition condition  $\langle 1, 1 | \varphi(D) | 0, 2 \rangle = -1/\sqrt{2}$  follows. Nearly any two-mode linear optical gate would suffice here, but  $D$ , the familiar Hadamard gate, maximizes the norm of the amplitude on state  $|1, 1\rangle$ . If the logical qubit is in state  $|1\rangle$ , then  $D$  is applied to the three-photon state  $|1, 2\rangle$ . Therefore, the resulting amplitude on the two-photon state

<sup>12</sup>To find  $E$ , we first define a set of constraints on transition amplitudes. The following equations must hold for this particular encoding gadget to exist:  $\langle 1, 1, 1 | \varphi(E) | 1, 1, 1 \rangle = 0$ ,  $\langle 2, 0, 1 | \varphi(E) | 1, 1, 1 \rangle = 0$ ,  $\langle 0, 2, 1 | \varphi(E) | 1, 1, 1 \rangle \neq 0$ . That is, starting from the state  $|1, 1, 1\rangle$ , there is some nonzero amplitude on the state  $|0, 2, 1\rangle$  and zero amplitude on the states  $|1, 1, 1\rangle$  and  $|2, 0, 1\rangle$ . We then solve these constraints using MATHEMATICA.

<sup>13</sup>One might wonder why we cannot simply apply the encoding gadget to the *entire* input, thus circumventing the need to use Knill's more complicated  $V$  gadget to implement CSIGN. Examining Theorem 2 carefully, we see that all the postselection actually happens at the end of the computation. One might be concerned that once we measured the state  $|0, 1\rangle$  to implement  $\text{NS}_1$ , those modes would remain in that state. Nevertheless, it *is* possible to compose the gadgets in such a way to allow for postselection on  $|0\rangle$  while maintaining that the desired amplitude is still on the  $|1, \dots, 1\rangle$  state. We omit such a design since  $V$  will turn out to have some nice properties, including its minimal usage of ancillary modes.

<sup>14</sup>Notice that postselection was required for the encoding gadget, so it does not have a natural inverse.



$|1, 1\rangle$  is zero by conservativity. To complete the proof of the theorem, let the linear optical circuit  $L$  simply be the composition of the encoding gadget, the KLM scheme, and the decoding gadget.

### 3.3 Main Result

We are finally ready to prove the Theorem 3, which we restate formally below.

► **Theorem 3.** *The permanent of a real orthogonal matrix is #P-hard. Specifically, given a polynomially sized Boolean circuit  $C$ , there exist integers  $a, b \in \mathbb{Z}$  and a real orthogonal matrix  $L$  over a finite Galois extension  $\mathbb{Q}(\alpha)$  (where  $\alpha = \sqrt{2 + \sqrt{2}} + \sqrt{3 + \sqrt{6}}$ ) computable in polynomial time such that*

$$\text{per}(L) = 2^a 3^b \Delta_C.$$

**Proof.** We reduce from the problem of calculating  $\Delta_C$  for some polynomially sized Boolean circuit  $C$  on  $n$  bits. By Theorem 4, we first construct the quantum circuit  $Q$  from CSIGN and single-qubit gates such that  $\langle 0 |^{\otimes p(n)} Q | 0 \rangle^{\otimes p(n)} = \Delta_C / 2^n$ . Let  $\Gamma$  be the number of CSIGN gates in  $Q$ . We then convert the qubit circuit  $Q$  to a linear optical circuit  $L$  on  $4p(n) + 2\Gamma$  modes using Theorem 6. Notice that we can assume without loss of generality that  $p(n)$  and  $\Gamma$  are both even since we can always add an extra qubit to the circuit  $Q$  and/or apply an extra CSIGN gate to the  $|00\rangle$  state. Combined with the fact that the output amplitudes of linear optical experiments can be described by permanents via the  $\varphi$ -transition formula, we have the following chain of consequences

$$\begin{aligned} \text{per}(L) &= \langle 1, \dots, 1 | \varphi(L) | 1, \dots, 1 \rangle \\ &= \left( \frac{1}{3} \sqrt{\frac{2}{3}} \right)^\Gamma \left( \frac{-1}{\sqrt{6}} \right)^{p(n)} \langle 0 |^{\otimes p(n)} Q | 0 \rangle^{\otimes p(n)} \\ &= \left( \frac{1}{3} \sqrt{\frac{2}{3}} \right)^\Gamma \left( \frac{-1}{\sqrt{6}} \right)^{p(n)} \left( \frac{1}{2^n} \right) \Delta_C \\ &= 2^a 3^b \Delta_C, \end{aligned}$$

where the last equality comes from the fact that  $\Gamma$  and  $p(n)$  are even. ◀

We now turn to the question of how to represent entries of the orthogonal matrix. First, the problem is clearly still hard if we generalize the matrix to arbitrary algebraic numbers (say, represented implicitly with integer polynomials) instead of only  $\mathbb{Q}(\alpha)$ . More practically, the entries may be represented as floating point numbers, such that the matrix is only approximately orthogonal due to rounding error. To this end, we state without proof the following corollary:

► **Corollary 7.** *Given  $\tilde{A} \in \mathbb{Q}^{n \times n}$  such that  $\|A - \tilde{A}\|_\infty \leq 2^{-cn}$  for some orthogonal matrix  $A \in \mathbb{R}^{n \times n}$ , the problem of computing  $\text{per}(\tilde{A})$  to within additive  $2^{-cn}$  precision is #P-hard for some constant  $c$ .*

## 4 Permanents over Finite Fields

Valiant’s foundational work on #P is well-known, but his contemporary work on the relationship between the permanent and the class we now know as  $\text{Mod}_k \text{P}$  is less appreciated. In another 1979 paper [32], Valiant showed that the permanent modulo  $p$  is  $\text{Mod}_p \text{P}$ -complete,

## 19:12 Permanent Hardness from Linear Optics

except when  $p = 2$ , in which case the permanent coincides with the determinant because  $1 \equiv -1 \pmod{2}$ .

► **Theorem 8** (Valiant [32]). *The problem of computing  $\text{per}(M) \bmod p$  for a square matrix  $M \in \mathbb{F}_p^{n \times n}$  is  $\text{Mod}_p\text{P}$ -complete for any prime  $p \neq 2$  (and in  $\text{NC}^2$  otherwise).*

As discussed in Appendix B,  $\text{Mod}_p\text{P}$ -hardness provides evidence for the difficulty of computing the permanent, even modulo a prime. In particular, an efficient algorithm for the problem would collapse the polynomial hierarchy.

In the spirit of our result on real orthogonal matrices, we ask whether the permanent is still hard for orthogonal matrices in a finite field. We are not the first to consider the problem; there is the following surprising theorem of Kogan [18] in 1996.

► **Theorem 9** (Kogan [18]). *Let  $\mathbb{F}$  be any field of characteristic 3. There is a polynomial time algorithm to compute the permanent of any orthogonal matrix over  $\mathbb{F}$ .*

In other words, for orthogonal matrices, the permanent is easy to compute for fields of characteristic 2 (since it is easy in general), but it is also easy for fields of characteristic 3 (by a much more elaborate argument)! Could it be that the permanent is easy for all finite fields of some other characteristic? No, it turns out. Using the gadgets from Section 3, we prove a converse to Theorem 9.

► **Theorem 10.** *Let  $p \neq 2, 3$  be a prime. There exists a finite field of characteristic  $p$ , namely  $\mathbb{F}_{p^4}$ , such that the permanent of an orthogonal matrix in  $\mathbb{F}_{p^4}$  is  $\text{Mod}_p\text{P}$ -hard.*

We prove the theorem by carefully porting Theorem 3 to the finite field setting. Recall that Theorem 3 takes a circuit  $C$  and constructs a sequence of gadgets  $G_1, \dots, G_m$  such that

$$\text{per}(G_1 \cdots G_m) = 2^a 3^b \Delta_C, \quad (1)$$

for some  $a, b \in \mathbb{Z}$ . In general, there is no way to convert such an identity on real numbers into one over finite fields, but all of our gadgets are built out of *algebraic* numbers. In particular, all of the entries are in some algebraic field extension  $\mathbb{Q}(\alpha)$  of the rationals, where  $\alpha \approx 4.182173283$  is the largest real root of irreducible polynomial

$$f(x) = x^{16} - 40x^{14} + 572x^{12} - 3736x^{10} + 11782x^8 - 17816x^6 + 11324x^4 - 1832x^2 + 1.$$

Each element in  $\mathbb{Q}(\alpha)$  can be written as a polynomial (of degree less than 16) in  $\alpha$  over the rationals. In Appendix D.1, we give explicit canonical representations for a set of numbers which generate (via addition, subtraction and multiplication, but *not* division) the entries of all our gadgets.

Each entry of a gadget  $G_i$  is a polynomial in  $\alpha$  with rational coefficients, so observe that we can take a common denominator for the coefficients and write the entry as an integer polynomial divided by some positive integer. By the same token, we can take a common denominator for the entries of a gadget  $G_i$ , and write it as  $\frac{1}{k_i} \hat{G}_i$  where  $\hat{G}_i$  is a matrix over  $\mathbb{Z}[\alpha]$ , and  $k_i$  is a positive integer.

Now we would like to take Equation 1 modulo a prime  $p$ . In principle, we can pull  $k_1, \dots, k_m$  out of the permanent, multiply through by  $Z = (k_1 \cdots k_m)^{n2^{|a|}3^{|b|}}$  to remove all fractions on both sides, and obtain an equation of the form

$$K \text{per}(\hat{G}_1 \cdots \hat{G}_m) = K' \Delta_C,$$

where  $K, K'$  are integers. Then the entire equation is over  $\mathbb{Z}[\alpha]$ , so if we reduce all the coefficients modulo  $p$ , we get an equation over  $\mathbb{F}_p[\alpha]$ .

We show in Appendix D.1 that for each gadget we use, the denominator  $k_i$  may have prime divisors 2, 3, and 23, but no others. Hence, as long as  $p \neq 2, 3, 23$  (and in the case  $p = 23$ , there is an alternative representation we can use, see Appendix D), we can divide through by  $Z$ , pull it back inside the permanent as the  $\frac{1}{k_i}$ s, and distribute each  $\frac{1}{k_i}$  into the corresponding  $\hat{G}_i$ . This gives

$$\text{per}(G_1 \cdots G_m) \equiv 2^a 3^b \Delta_C \pmod{p},$$

the equivalent of Equation 1, but over  $\mathbb{F}_p[\alpha]$ . In particular,  $G_1, \dots, G_m$  are now orthogonal matrices in  $\mathbb{F}_p[\alpha]$ , and  $\Delta_C$  has been reduced modulo  $p$ .

Note that  $\mathbb{F}_p[\alpha] \cong \mathbb{F}_p[x]/(f(x))$  is a ring, not a field. If  $f(x)$  were irreducible modulo  $p$  then it would be a field, but this will never happen for our  $f$ . Consider the following lemma.

► **Lemma 11.** *Let  $q$  be a prime power. Suppose  $\mathbb{F}_q$  is the subfield of order  $q$  contained in the finite field  $\mathbb{F}_{q^2}$ . Then every element in  $\mathbb{F}_q$  has a square root in  $\mathbb{F}_{q^2}$ .*

**Proof.** Let  $a$  be an arbitrary element of  $\mathbb{F}_q$ . By definition,  $a$  has a square root if the polynomial  $f(x) := x^2 - a$  has a root. If  $f$  has a root in  $\mathbb{F}_q$  then we are done. Otherwise,  $f$  is irreducible, but has a root in  $\mathbb{F}_q[x]/\langle f(x) \rangle \cong \mathbb{F}_{q^2}$ . ◀

By Lemma 11, the square roots of 2 and 6 are in  $\mathbb{F}_{p^2}$ , and therefore so are  $2 + \sqrt{2}$  and  $3 + \sqrt{6}$ . Then *their* square roots are in  $\mathbb{F}_{p^4}$ , so  $\alpha = \sqrt{2 + \sqrt{2}} + \sqrt{3 + \sqrt{6}}$  is in  $\mathbb{F}_{p^4}$ . All the other roots of  $f$  can be expressed as polynomials in  $\alpha$  (see Appendix D.2), so they are all in  $\mathbb{F}_{p^4}$ . It follows that  $f$  factors over  $\mathbb{F}_p$  as a product of irreducible polynomials, each of degree 1, 2, or 4.

Suppose  $g$  is some irreducible factor of  $f$ . The ideal  $(g(x))$  contains  $(f(x))$ , so there exists a ring homomorphism  $\sigma$  from  $\mathbb{F}_p[x]/(f(x))$  to  $\mathbb{F}_p[x]/(g(x))$ . Note that  $\mathbb{F}_p[x]/(g(x))$  is a field because  $g(x)$  is irreducible over  $\mathbb{F}_p$ . Also,  $\sigma$  fixes  $\mathbb{F}_p$ , so we obtain

$$\text{per}(\sigma(G_1) \cdots \sigma(G_m)) = \sigma(\text{per}(G_1 \cdots G_m)) = 2^a 3^b \Delta_C$$

as an equation over the field  $\mathbb{F}_p[x]/(g(x))$ . For each  $i$ ,  $\sigma(G_i)$  is orthogonal in  $\mathbb{F}_p[x]/(g(x))$  as well:

$$\sigma(G_i)\sigma(G_i)^T = \sigma(G_i G_i^T) = \sigma(I) = I.$$

It follows that  $M := \sigma(G_1) \cdots \sigma(G_m)$  is orthogonal.

Depending on the degree of  $g$ , the field  $\mathbb{F}_p[x]/(g(x))$  is isomorphic to  $\mathbb{F}_p$ ,  $\mathbb{F}_{p^2}$ , or  $\mathbb{F}_{p^4}$ . But  $\mathbb{F}_{p^4}$  contains  $\mathbb{F}_p$  and  $\mathbb{F}_{p^2}$ , so  $M$  can be lifted to a matrix over  $\mathbb{F}_{p^4}$ . Given the permanent of  $M$  in  $\mathbb{F}_{p^4}$ , we can easily solve for  $\Delta_C$ , so this completes the proof of Theorem 10.

Theorem 10 shows that for any prime  $p \neq 2, 3$  there is some finite field of characteristic  $p$  where computing permanents (of orthogonal matrices) is hard. In particular,  $p = 2$  and  $p = 3$  are the only cases where the permanent of an orthogonal matrix is easy to compute in *every* finite field of characteristic  $p$ , assuming the polynomial hierarchy does not collapse. We will now show that there are primes  $p$  for which this problem is hard in *any* field of characteristic  $p$ , by showing that it is hard to compute in  $\mathbb{F}_p$  (which is contained in every other field of characteristic  $p$ ).

► **Theorem 12.** *For all but finitely many primes  $p$  that split completely in  $\mathbb{Q}(\alpha)$ , computing the permanent of an orthogonal matrix over  $\mathbb{F}_p$  is  $\text{Mod}_p\text{P}$ -complete. This is a sequence of primes with density  $\frac{1}{16}$  beginning*

191, 239, 241, 337, 383, 433, 673, 863, 911, 1103, 1151, 1249, 1583, 1871, 1873, 2017, ...

## 19:14 Permanent Hardness from Linear Optics

**Proof.** Recall that in the proof of Theorem 10, if  $g$  is an irreducible factor of  $f$ , then the result applies over the field  $\mathbb{F}_p[x]/(g(x)) \cong \mathbb{F}_{p^{\deg g}}$ . We show that  $g$  is degree at most 4, but in special cases this can be improved. In particular, we want  $g$  to be degree 1 (i.e., a linear factor) for our orthogonal matrix to be over  $\mathbb{F}_p$ .

First, observe that  $\mathbb{Q}(\alpha)$  is a Galois extension of  $\mathbb{Q}$ . That is, every root of the minimal polynomial for  $\alpha$  is in  $\mathbb{Q}(\alpha)$ . See Appendix D.2 for details. We apply Chebotarev’s density theorem [30], which says that if  $K$  is a finite Galois extension of  $\mathbb{Q}$  of degree  $n$ , then the density of primes which split completely in  $K$  is  $\frac{1}{n}$ . We take  $K = \mathbb{Q}(\alpha)$ , a degree 16 extension of  $\mathbb{Q}$ .

For our purposes, a prime  $p$  splits completely if and only if the ideal  $(p)$  factors into 16 distinct maximal ideals in the ring of integers of  $\mathbb{Q}(\alpha)$ . For all but finitely many such primes,<sup>15</sup> we also have that  $f$  (the minimal polynomial for  $\alpha$ ) factors into distinct linear terms modulo  $p$  by Dedekind’s theorem. Furthermore, since  $\mathbb{Q}(\alpha)$  is a Galois extension,  $f$  will split into equal degree factors. Hence, if any factor is linear, then *all* the factors are linear.

Therefore, according to Chebotarev’s theorem,  $(1/16)$ th of all primes split completely and yield the desired hardness result. We verified the list of primes given in the theorem computationally. ◀

Note that as a consequence of the proof above theorem, we can also prove a hardness result over  $\mathbb{F}_{p^2}$  for  $3/16$  of all primes. We leave open how hard it is to compute the permanent of an orthogonal matrix over  $\mathbb{F}_p$  for the remaining  $15/16$  of all primes. Other linear optical gadgets can be used for CSIGN instead of  $V$ , resulting in different field extensions where different primes split. For instance, there exists an orthogonal gadget for KLM’s  $\text{NS}_1$  gate for which computing the permanent modulo 97 is hard (see Appendix F). However, it seems impossible to design linear optical gadgets that do not involve 2 or 3 photons at a time, in which case writing down  $\varphi(L)$  requires  $\sqrt{2}$  and  $\sqrt{3}$ . By quadratic reciprocity, these square roots only exist if  $p \equiv \pm 1 \pmod{24}$  (i.e., for about a quarter of all primes), so the remaining primes may require some other technique.

## 5 Expanding Permanent Hardness

In this section, we try to fill in some of the remaining landscape of matrix permanents. In particular, we will focus on the permanents of positive semidefinite (PSD) matrices and their connection to boson sampling. We will conclude by listing some matrix variants and their accompanying permanent complexities, many of which are simple consequences of the reduction in Section 3.

### 5.1 Positive Semidefinite Matrix Permanents

Permanents of PSD matrices have recently become relevant to the expanding theory of boson sampling [21]. Namely, permanents of PSD matrices describe the output probabilities of a boson sampling experiment in which the input is a tensor product of thermal states. Suppose we have a thermal state with  $m$  modes. The  $i$ th mode of the system starts in a state of the form

---

<sup>15</sup> Actually, we can compute these primes explicitly as those that divide the index of  $\mathbb{Z}[\alpha]$  in the ring of integers of  $\mathbb{Q}(\alpha)$ . For our choice of field, this number is  $19985054955504338544361472 = 2^{75}23^2$ .

$$\rho_i = (1 - \tau_i) \sum_{n=0}^{\infty} \tau_i^n |n\rangle\langle n|$$

where  $\tau_i = \langle n_i \rangle / (\langle n_i \rangle + 1)$  and  $\langle n_i \rangle$  is average number of photons one observes when measuring  $\rho_i$ . In particular, notice that  $\tau_i \geq 0$ .

Let  $U$  be a unitary matrix representing the linear optical network applied to our thermal state. Define  $D$  to be the diagonal matrix with  $\tau_1, \dots, \tau_m$  along the diagonal, and let  $A = UDU^\dagger$ . Since  $\tau_i \geq 0$  for all  $i$ ,  $A$  is PSD. We can calculate the probability of detecting one photon in each mode:<sup>16</sup>

$$\langle 1, \dots, 1 | \left( \varphi(U) \left( \bigotimes_{i=1}^m \rho_i \right) \varphi(U)^\dagger \right) | 1, \dots, 1 \rangle = \frac{\text{per}(A)}{\prod_{i=1}^m (1 + \langle n_i \rangle)}.$$

One might then reasonably ask, “how hard is it to compute such probabilities?” The following theorem answers that question in the exact case.

► **Theorem 13.** *The permanent of a positive-definite matrix in  $\mathbb{Z}^{n \times n}$  is #P-hard. This implies #P-hardness for the larger class of positive semidefinite matrices.*

**Proof.** It is well-known that the permanent of a 0-1 matrix is #P-hard [31]. Therefore, let  $B \in \{0, 1\}^{n \times n}$  and consider the matrix

$$\Lambda_B = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}$$

Since  $\text{per}(B) \geq 0$ , we have  $\text{per}(B) = \sqrt{\text{per}(\Lambda_B)}$ . Also observe that  $\Lambda_B^T = \Lambda_B$ , so  $\Lambda_B$  is Hermitian, that is, diagonalizable with real eigenvalues. Furthermore, since  $B$  is a 0-1 matrix, its spectral radius is at most  $2n$ . Defining  $\Lambda_B(x) := \Lambda_B + xI$ , we see that  $\Lambda_B(x)$  is positive-definite for all  $x > 2n$ .

Notice now that  $\text{per}(\Lambda_B(x))$  is a degree- $2n$  polynomial in  $x$ . Therefore, given an oracle that calculates the permanent of a positive-definite matrix, we can interpolate a monic polynomial through the points  $x = 2n + 1, 2n + 2, \dots, 4n$  to recover the polynomial  $\text{per}(\Lambda_B(x))$ . Since  $\text{per}(\Lambda_B(0)) = \text{per}(\Lambda_B)$ , the permanent of a positive-definite matrix under *Turing reductions* is #P-hard.

We now only have left to prove that the above reduction can be condensed into a single call to the positive-definite matrix permanent oracle. Since the matrix  $B$  is a 0-1 matrix, the polynomial  $\text{per}(\Lambda_B(x))$  has positive integer coefficients, the largest of which is at most  $(2n)!$ . Therefore, if  $x > (2n)!$ , then we can deduce the constant term of  $\text{per}(\Lambda_B(x))$  with a single oracle call. Clearly, this requires at most a polynomial increase in the bit length of the integers used in the reduction. ◀

Theorem 13 implies that there is some linear optical experiment one can perform with thermal input states for which calculating the exact success probability is computationally difficult. We would like to say that this also precludes an efficient classical sampling algorithm (unless PH collapses), as is done in work by Aaronson and Arkhipov [3] and Bremner, Jozsa, Shepherd [8]. Unfortunately, those arguments rely on the fact that even finding an *approximation* to their output probabilities is difficult, but the following theorem heavily suggests that such a result cannot exist.

<sup>16</sup> A similar formula arises for detecting 1 photon in each of  $k$  distinct modes and 0 photons in the remaining  $m - k$  modes.

► **Theorem 14** (Rahimi-Keshari, Lund, Ralph [21]). *There exists an efficient classical sampling algorithm for Boson Sampling with thermal input states. Furthermore, multiplicatively approximating the permanent of a PSD matrix is in the class  $\text{FBPP}^{\text{NP}}$ .*

Intuitively, such an algorithm exists because it is possible to write the permanent of a PSD matrix as an integral<sup>17</sup> of a nonnegative function, on which we can use Stockmeyer’s approximate counting algorithm [25]. Such a representation as a sum of positive terms also implies that the permanent of a PSD matrix is nonnegative.

Notice that this also justifies our use of techniques distinct from the linear optical approach. Suppose we can encode the answer to a  $\text{GapP}$ -hard problem into the permanent of a PSD matrix as we do with real orthogonal matrices, then multiplicatively approximating the permanent of a PSD matrix would also be  $\text{GapP}$ -hard under Turing reductions (see Theorem 30 in Appendix E). On the other hand, Theorem 14 says that such a multiplicative approximation *does* exist, so

$$\text{PH} \subseteq \text{P}^{\text{GapP}} \subseteq \text{BPP}^{\text{NP}} \subseteq \Sigma_3^{\text{P}}.$$

Therefore, either such a reduction does not exist or the polynomial hierarchy collapses to the third level.

## 5.2 More Permanent Consequences of the Main Result

In this section, we try to give a sense in which our proof for the hardness of the permanent for real orthogonal matrices leads to new hardness results for many classes of matrices. The structure of this section is as follows: we will first *restrict* as much as possible the class of matrices for which the permanent is  $\#\text{P}$ -hard; we will then observe that the permanent for any larger class of matrices must also be hard, which will show hardness for many natural classes of matrices.

We call matrix  $A$  an *involution* if  $A = A^{-1}$ .

► **Theorem 15.** *Let  $A$  be a real orthogonal involution with  $\text{per}(A) \geq 0$ . The permanent of  $A$  is  $\#\text{P}$ -hard.*

**Proof.** Let  $C : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function for which we want to calculate  $\Delta_C$ . We will construct a new circuit  $C' : \{0, 1\}^{n+1} \rightarrow \{0, 1\}$  such that for  $x \in \{0, 1\}^n$  and  $b \in \{0, 1\}$  we have  $C'(x, b) = C(x) \vee b$ . It is not hard to see then that  $\Delta_{C'} = \Delta_C + 2^n$ . Importantly, this implies that  $\Delta_{C'} \geq 0$ .

Now let us leverage the reduction in Theorem 3 to build a real orthogonal matrix  $B$  such that  $\text{per}(B) \propto \Delta_{C'}$ . As in the proof of Theorem 13, let

$$\Lambda_B = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}.$$

Since  $\Delta_{C'} \geq 0$ , we have  $\text{per}(B) \geq 0$ , which implies that  $\text{per}(B) = \sqrt{\text{per}(\Lambda_B)}$ . However, since  $B$  is orthogonal, we have that  $\Lambda_B^2 = I$ , so  $\Lambda_B$  is an involution. Furthermore,  $\Lambda_B = \Lambda_B^T$ , so

---

<sup>17</sup> Suppose we have PSD matrix  $A = CC^\dagger$  where  $C = \{c_{i,j}\}$ . Then the permanent of  $A$  can be expressed as the following expected value over complex Gaussians:

$$\text{per}(A) = \mathbb{E}_{x \in \mathcal{G}_{\mathbb{C}}(0,1)^n} \left[ \prod_{i=1}^n \left| \sum_{j=1}^n c_{i,j} x_j \right|^2 \right].$$

$\Lambda_B$  is a real orthogonal matrix. Therefore, the permanent of real orthogonal involutions is #P-hard. ◀

We call a matrix  $A$  *special* if  $\det(A) = 1$ . Furthermore, a matrix  $A$  is *symplectic* if  $A^T \Omega A = \Omega$  where  $\Omega = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}$ . We strengthen Theorem 15 to provide the smallest class of matrices for which we know the permanent is #P-hard.

► **Theorem 16.** *Let  $A$  be a real special orthogonal symplectic involution with  $\text{per}(A) \geq 0$ . The permanent of  $A$  is #P-hard.*

**Proof.** Let  $B$  be a real orthogonal involution, and let  $I_n$  be the  $n \times n$  identity matrix. Consider the matrix

$$I_2 \otimes B = \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix}.$$

Notice that

$$\det(I_2 \otimes B) = \det(B)^2 = \det(B^2) = \det(I_n) = 1,$$

where we use that  $B^2 = I_n$  is an involution for the third equality. Therefore,  $I \otimes B$  is special. It is also easy to verify that  $I_2 \otimes B$  is real orthogonal symplectic involution. Assuming  $\text{per}(B) \geq 0$ , we have  $\text{per}(B) = \sqrt{\text{per}(I_2 \otimes B)}$ . Combining the above with Theorem 15, we get that the permanent of real special orthogonal involutions is #P-hard. ◀

Since the set of  $n \times n$  real special orthogonal matrices form a group  $\text{SO}(n, \mathbb{R})$ , we immediately get #P-hardness for all the matrix groups containing it.

► **Corollary 17.** *The permanent of an  $n \times n$  matrix  $A$  in any of the classical Lie groups over the complex numbers is #P-hard. That is, it is hard for the following matrix groups:*

**General linear:**  $A \in \text{GL}(n)$  iff  $\det(A) \neq 0$

**Special linear:**  $A \in \text{SL}(n)$  iff  $\det(A) = 1$

**Orthogonal:**  $A \in \text{O}(n)$  iff  $AA^T = I_n$

**Special orthogonal:**  $A \in \text{SO}(n)$  iff  $AA^T = I_n$  and  $\det(A) = 1$

**Unitary:**  $A \in \text{U}(n)$  iff  $AA^\dagger = I_n$

**Special unitary:**  $A \in \text{SU}(n)$  iff  $AA^\dagger = I_n$  and  $\det(A) = 1$

**Symplectic:**  $A \in \text{Sp}(2n)$  iff  $A^T \Omega A = \Omega$  where  $\Omega = \begin{pmatrix} 0 & I_n \\ -I_n & 0 \end{pmatrix}$

**Proof.** Since  $\text{SO}(n, \mathbb{R})$  is a subgroup of all the stated Lie groups besides the symplectic group  $\text{Sp}(2n)$ , their permanents are #P-hard by Theorem 16. Theorem 16 handles the symplectic case separately. ◀

## 6 Open Problems

This paper gives many new classes of matrices for which the permanent is hard. Nevertheless, there exist classes of matrices which have unknown permanent complexity, and proving #P-hardness or otherwise remains a central open problem. For instance, is computing the permanent of an orthogonal matrix modulo a prime  $p$  hard for all  $p \neq 2, 3$ ? Notice that our result only gives  $\text{Mod}_p$ P-hardness for 1/16th of all primes.

Another interesting open question about permanents concerns the complexity of multiplicatively approximating permanents of PSD matrices. Although we show the exact version



of this problem to be  $\#P$ -hard in this paper, we know that an  $\text{FBPP}^{\text{NP}}$  algorithm exists [21]. Could this problem actually just be in  $P$ ? Is there any more insight to be gained by viewing PSD permanents as probabilities of certain boson sampling experiments? For instance, Chakhmakhchyan, Cerf, and Garcia-Patron [10] have recently detailed conditions on the eigenvalues of a PSD matrix for which a linear optical sampling algorithm gives a better *additive* approximation to the permanent than the classic approximation algorithm of Gurvits [14].

---

## References

- 1 S. Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proc. Roy. Soc. London*, A461(2063):3473–3482, 2005. quant-ph/0412187.
- 2 S. Aaronson. A linear-optical proof that the permanent is  $\#P$ -hard. *Proc. Roy. Soc. London*, A467(2088):3393–3405, 2011. arXiv:1109.1674.
- 3 S. Aaronson and A. Arkhipov. The computational complexity of linear optics. *Theory of Computing*, 9(4):143–252, 2013. Conference version in Proceedings of ACM STOC’2011. ECCS TR10-170, arXiv:1011.3245.
- 4 S. Aaronson, D. Grier, and L. Schaeffer. The classification of reversible bit operations. *arXiv preprint arXiv:1504.05155*, 2015.
- 5 Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh A. Huang. Quantum computability. *SIAM J. Comput.*, 26(5):1524–1540, 1997. doi:10.1137/S0097539795293639.
- 6 A. Ben-Dor and S. Halevi. Zero-one permanent is  $\#P$ -complete, a simpler proof. In *ISTCS*, pages 108–117, 1993.
- 7 S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information processing letters*, 18(3):147–150, 1984.
- 8 M. Bremner, R. Jozsa, and D. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. Roy. Soc. London*, A467(2126):459–472, 2010. arXiv:1005.1407.
- 9 E. R. Caianiello. On quantum field theory, 1: explicit solution of Dyson’s equation in electrodynamics without use of Feynman graphs. *Nuovo Cimento*, 10:1634–1652, 1953.
- 10 L. Chakhmakhchyan, N. J. Cerf, and R. Garcia-Patron. A quantum-inspired algorithm for estimating the permanent of positive semidefinite matrices. *arXiv preprint arXiv:1609.02416*, 2016.
- 11 A. Drucker and R. de Wolf. Quantum proofs for classical theorems. *Theory of Computing Graduate Surveys*, pages 1–54, 2011. arXiv:0910.3376, ECCS TR03-048.
- 12 S. Fenner, F. Green, S. Homer, and R. Pruim. Quantum NP is hard for PH. In *Proceedings of 6th Italian Conference on theoretical Computer Science*, pages 241–252. Citeseer, 1998.
- 13 Stephen A. Fenner, Lance Fortnow, and Stuart A. Kurtz. Gap-definable counting classes. *J. Comput. Syst. Sci.*, 48(1):116–148, 1994. doi:10.1016/S0022-0000(05)80024-8.
- 14 Leonid Gurvits. On the complexity of mixed discriminants and related problems. In Joanna Jedrzejowicz and Andrzej Szepietowski, editors, *Mathematical Foundations of Computer Science 2005, 30th International Symposium, MFCS 2005, Gdansk, Poland, August 29 - September 2, 2005, Proceedings*, volume 3618 of *Lecture Notes in Computer Science*, pages 447–458. Springer, 2005. doi:10.1007/11549345\_39.
- 15 C. K. Hong, Z. Y. Ou, and L. Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Phys. Rev. Lett.*, 59(18):2044–2046, 1987.
- 16 E. Knill. Quantum gates using linear optics and postselection. *Physical Review A*, 66(5), 2002. doi:10.1103/PhysRevA.66.052306.
- 17 E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409:46–52, 2001. See also quant-ph/0006088.



- 18 G. Kogan. Computing permanents over fields of characteristic 3: Where and why it becomes difficult. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 108–114. IEEE, 1996.
- 19 G. Kuperberg. How hard is it to approximate the Jones polynomial? *Theory of Computing*, 11(6):183–219, 2015.
- 20 M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- 21 S. Rahimi-Keshari, A. P. Lund, and T. C. Ralph. What can quantum optics say about computational complexity theory? *Physical review letters*, 114(6):060501, 2015.
- 22 Terry Rudolph. Simple encoding of a quantum circuit amplitude as a matrix permanent. *Physical Review A*, 80(5):054302, 2009.
- 23 Rimli Sengupta. Cancellation is exponentially powerful for computing the determinant. *Information Processing Letters*, 62(4):177–181, 1997.
- 24 Y. Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computation. *Quantum Information and Computation*, 3(1):84–92, 2002. quant-ph/0205115.
- 25 L. J. Stockmeyer. The complexity of approximate counting. In *Proc. ACM STOC*, pages 118–126, 1983.
- 26 S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.
- 27 Seinosuke Toda and Mitsunori Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 21(2):316–328, 1992. doi:10.1137/0221023.
- 28 T. Toffoli. Reversible computing. In *Proc. Intl. Colloquium on Automata, Languages, and Programming (ICALP)*, pages 632–644. Springer, 1980.
- 29 L. Troyansky and N. Tishby. Permanent uncertainty: On the quantum evaluation of the determinant and the permanent of a matrix. In *Proceedings of PhysComp*, 1996.
- 30 N. Tschebotareff. Die bestimmung der dichtigkeit einer menge von primzahlen, welche zu einer gegebenen substitutionsklasse gehören. *Mathematische Annalen*, 95(1):191–228, 1926. doi:10.1007/BF01206606.
- 31 L. G. Valiant. The complexity of computing the permanent. *Theoretical Comput. Sci.*, 8(2):189–201, 1979.
- 32 Leslie G. Valiant. Completeness classes in algebra. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.
- 33 Leslie G. Valiant. Negation can be exponentially powerful. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 189–196. ACM, 1979. doi:10.1145/800135.804412.

## A Linear optics as a symmetric subspace

The Fock space  $\Phi_{m,k}$  can alternatively be described as a linear subspace of  $(\mathbb{C}^m)^{\otimes k}$ , the Hilbert space of  $k$  qudits with local dimension  $m$ .

A single photon can be in one of  $m$  modes, so is described as a unit vector in  $\mathbb{C}^m$ . Therefore, transformations on single photons are unitary matrices in  $U(m)$ . Linear optical states with multiple photons are described by the symmetric tensor. That is, for  $v_1, v_2, \dots, v_k \in \mathbb{C}^m$ , let

$$v_1 \odot v_2 \odot \dots \odot v_k = \frac{1}{k!} \sum_{\sigma \in S_k} v_{\sigma(1)} \otimes v_{\sigma(2)} \otimes \dots \otimes v_{\sigma(k)}$$

## 19:20 Permanent Hardness from Linear Optics

be their symmetric tensor. Notice that the symmetric tensor is invariant under permutations; that is,  $v_1 \odot v_2 \odot \dots \odot v_k = v_{\sigma(1)} \odot v_{\sigma(2)} \odot \dots \odot v_{\sigma(k)}$  for any  $\sigma \in S_k$ . This captures the physical intuition that the photons are indistinguishable.

We can extend the usual inner product to the symmetric setting:

$$\begin{aligned}
 & \langle v_1 \odot \dots \odot v_k, w_1 \odot \dots \odot w_k \rangle \\
 &= \left( \frac{1}{k!} \sum_{\sigma \in S_k} v_{\sigma(1)}^\dagger \otimes \dots \otimes v_{\sigma(k)}^\dagger \right) \left( \frac{1}{k!} \sum_{\rho \in S_k} w_{\rho(1)} \otimes \dots \otimes w_{\rho(k)} \right) \\
 &= \left( \frac{1}{k!} \right)^2 \sum_{\sigma, \rho \in S_k} \left( v_{\sigma(1)}^\dagger w_{\rho(1)} \right) \cdots \left( v_{\sigma(k)}^\dagger w_{\rho(k)} \right) \\
 &= \frac{1}{k!} \sum_{\rho \in S_k} \langle v_1, w_{\rho(1)} \rangle \cdots \langle v_k, w_{\rho(k)} \rangle \\
 &= \frac{1}{k!} \text{per}(\langle v_i, w_j \rangle)_{i,j}.
 \end{aligned}$$

We are now ready to define an orthonormal basis for  $\Phi_{m,k}$ . Let  $e_1, \dots, e_m$  be the standard basis for  $\mathbb{C}^m$ , where  $e_i$  represents a photon in mode  $i$ . The basis vectors for  $\Phi_{m,k}$  will be  $k$ -fold symmetric tensors of the  $e_i$  vectors. Let  $v_1 \odot \dots \odot v_k$  be one such symmetric tensor with  $v_j \in \{e_1, \dots, e_m\}$ . Let  $s_i = |\{v_j \mid v_j = e_i\}|$ ; that is, there are  $s_i$  photons in mode  $i$ . We will denote the corresponding basis vector in  $\Phi_{m,k}$  as  $|s_1, \dots, s_m\rangle$ . Formally,

$$|s_1, \dots, s_m\rangle = \sqrt{\frac{k!}{s_1! s_2! \cdots s_m!}} (v_1 \odot v_2 \odot \dots \odot v_k).$$

Notice that if we specify the symmetric tensor by the  $s_i$  in this way, we lose the relative ordering of the elements  $v_1, \dots, v_k$ . Recall, however, that any choice will do since the symmetric tensor is invariant under permutation.

► **Theorem 18.** *The elements  $|s_1, \dots, s_m\rangle$  such that  $\sum_{i=1}^m s_i = k$  form an orthonormal basis for  $\Phi_{m,k}$ .*

**Proof.** First, it should be clear that every symmetrized basis vector of  $(\mathbb{C}^m)^{\otimes k}$  corresponds to some element  $|s_1, \dots, s_m\rangle$  such that  $\sum_{i=1}^m s_i = k$ . We need now only show orthonormality. For states  $|s_1, \dots, s_m\rangle$  and  $|t_1, \dots, t_m\rangle$ , we have

$$\begin{aligned}
 \langle t_1, \dots, t_m \mid s_1, \dots, s_m \rangle &= \frac{k!}{\sqrt{s_1! \cdots s_m! t_1! \cdots t_m!}} \langle v_1 \odot \dots \odot v_k, w_1 \odot \dots \odot w_k \rangle \\
 &= \frac{\text{per}(\langle v_i, w_j \rangle)_{i,j}}{\sqrt{s_1! \cdots s_m! t_1! \cdots t_m!}}
 \end{aligned}$$

Since the  $e_i$  form an orthonormal basis for  $\mathbb{C}^m$ , we have  $\langle v_i, w_j \rangle = 1$  when  $v_i = w_j$  and 0 otherwise. Therefore, if there exists  $i$  such that  $s_i \neq t_i$ , then  $\text{per}(\langle v_i, w_j \rangle)_{i,j} = 0$ . Otherwise, the value of this permanent is equal to the number of permutations  $\sigma \in S_k$  such that  $v_{\sigma(j)} = w_j$  for all  $j$ . In other words, these permutations only permute the photons within each mode. Since there are  $s_i$  many photons in mode  $i$ , there are  $s_i!$  many permutations of photons in that mode. Therefore,  $\text{per}(\langle v_i, w_j \rangle)_{i,j} = s_1! s_2! \cdots s_m!$ , which completes the proof. ◀

Finally, we must describe the transformations of the space  $\Phi_{m,k}$ . These are just those transformations that act identically on all photons. For  $A \in \mathbb{C}^{m \times m}$ , we write  $\varphi(A) = A^{\otimes k}$

as its  $k$ -fold tensor product. This notation is often convenient because it suppresses the parameter  $k$ . Notice that  $\varphi(A)$  fixes the subspace  $\Phi_{m,k}$  since

$$A^{\otimes k}(v_1 \odot \dots \odot v_k) = \frac{1}{k!} \sum_{\sigma \in S_k} Av_{\sigma(1)} \otimes \dots \otimes Av_{\sigma(k)} = w_1 \odot \dots \odot w_k,$$

where  $w_i = Av_i \in \mathbb{C}^m$ . We get the following other important properties of  $\varphi$  from this definition:

► **Fact 19.** *If  $U$  is unitary, then  $\varphi(U)$  is unitary.*

**Proof.** If  $U \in U(m)$ , then  $U^{\otimes k} \in U(m^k)$ . ◀

► **Fact 20.**  *$\varphi$  is a group homomorphism:  $\varphi(AB) = \varphi(A)\varphi(B)$  for  $A, B \in \mathbb{C}^{m \times m}$ .*

**Proof.**  $(AB)^{\otimes k} = A^{\otimes k}B^{\otimes k}$ . ◀

Finally, we ready to state and prove the  $\varphi$ -transition formula.

► **Theorem 21.** *For  $A \in \mathbb{C}^{m \times m}$ ,  $|S\rangle = |s_1, \dots, s_m\rangle \in \Phi_{m,k}$ , and  $|T\rangle = |t_1, \dots, t_m\rangle \in \Phi_{m,k}$ , we have*

$$\langle t_1, \dots, t_m | \varphi(A) | s_1, \dots, s_m \rangle = \frac{\text{per}(A_{S,T})}{\sqrt{s_1! \dots s_m! t_1! \dots t_m!}}$$

provided  $\sum_{i=1}^m s_i = \sum_{i=1}^m t_i = k$ . Let  $A_{S,T}$  be the matrix obtained by taking  $s_i$  copies of the  $i$ th row and  $t_i$  copies of the column  $i$  in  $A$  for all  $i \in \{1, 2, \dots, m\}$ .

**Proof.** By definition, we have

$$\langle t_1, \dots, t_m | \varphi(A) | s_1, \dots, s_m \rangle = k! \cdot \frac{\langle v_1 \odot \dots \odot v_k, Aw_1 \odot \dots \odot Aw_k \rangle}{\sqrt{s_1! \dots s_m! t_1! \dots t_m!}} = \frac{\text{per}(\langle v_i, Aw_j \rangle)_{i,j}}{\sqrt{s_1! \dots s_m! t_1! \dots t_m!}}$$

where  $v_i, w_j \in \{e_1, \dots, e_m\}$ ,  $s_i = \{w_j \mid w_j = e_i\}$ , and  $t_i = \{v_j \mid v_j = e_i\}$ . Notice that if  $v_i = e_k$ , then the  $i$ th row of the matrix  $(\langle v_i, Aw_j \rangle)_{i,j}$  corresponds to the  $k$ th row of  $A$ . Following this reasoning, we get that  $(\langle v_i, Aw_j \rangle)_{i,j} = A_{S,T}$ . ◀

## B Counting Classes

Let us introduce the complexity classes we use in this paper. Note that the permanent is a function, so computing it is a function problem. Hence, we will sometimes need the class FP to stand in for P when we are talking about function problems.

► **Definition 22.** FP is the class of functions computable by deterministic Turing machines in polynomial time.

Of course, computing the permanent is, in general, thought to be intractable (i.e., not in FP). We use a variety of different classes to capture the difficulty of computing the permanent (depending on the kind of matrix, underlying field, etc.), but the most important class is #P:

► **Definition 23.** #P is the class of function problems of the form “compute the number of accepting paths of a polynomial-time non-deterministic Turing machine.” For example, given a classical circuit of NAND gates as input, the problem of computing the number of satisfying assignments is in #P (and indeed, is #P-complete).

Since  $\#P$  is a class of function problems (more specifically, counting problems), we often consider  $P^{\#P}$  to compare  $\#P$  to decision classes. Observe that  $P^{\#P} = P^{PP}$  since, on the one hand, the  $\#P$  oracle can count paths to simulate  $PP$ , and on the other hand, we can use the  $PP$  oracle to binary search (on the number of accepting paths) to count exactly. We add that  $P^{\#P} \subseteq PSPACE$  is an upper bound for  $\#P$ , and Toda's theorem [26] gives  $PH \subseteq P^{\#P}$ .

Fenner, Fortnow, and Kurtz [13] define a very closely related class,  $GapP$ , which is also relevant to us.

► **Definition 24.**  $GapP$  is the class of function problems of the form “compute the number of accepting paths *minus* the number of rejecting paths of a polynomial-time non-deterministic Turing machine.”

We have  $GapP \supseteq \#P$  since we can take a  $\#P$  problem (manifest as a non-deterministic Turing machine) and at the end of each rejecting path, add a non-deterministic branch which accepts in one half and rejects in the other. In the other direction, any  $GapP$  problem can be solved with at most two calls to a  $\#P$  oracle (one for accepting paths, one for rejecting), and a subtraction. Hence, for most of our results we neglect the difference.

Nonetheless,  $GapP$  and  $\#P$  are different. For one, functions in  $\#P$  are non-negative (and integral) by definition, whereas functions in  $GapP$  can take negative values. The distinction is also important in the context of approximation; Stockmeyer's approximate counting gives a multiplicative approximation to any  $\#P$  problem in  $BPP^{NP}$ , whereas it is known that multiplicative approximation to a  $GapP$ -hard problem remains  $GapP$ -hard under Turing reductions (see Theorem 30).

One cannot even get very bad multiplicative approximations to  $GapP$ -hard problems. Even the worst multiplicative approximation will distinguish zero from non-zero outputs, and this problem is captured by the class  $C=P$ , defined below.

► **Definition 25.**  $C=P$  is the class of decision problems solvable by a non-deterministic polynomial-time machine which accepts if it has the same number of accepting paths as rejecting paths.

A good upper bound for  $C=P$  is simply  $PP$ . This is easily seen once we have the following theorem.

► **Theorem 26.** *Suppose  $f_1, f_2 \in \Sigma^* \rightarrow \mathbb{Z}$  are functions computable in  $GapP$ . Then  $f_1 + f_2$ ,  $-f_1$ , and  $f_1 f_2$  are computable in  $GapP$ .*

**Proof.** Let  $M_1$  and  $M_2$  be non-deterministic machines witnessing  $f_1 \in GapP$  and  $f_2 \in GapP$  respectively. Then the machines for  $f_1 + f_2$ ,  $-f_1$ , and  $f_1 f_2$  are defined as follows.

1. For  $f_1 + f_2$ , non-deterministically branch at the start, then run  $M_1$  in one branch and  $M_2$  in the other.
2. For  $-f_1$ , take the complement of  $M_1$ . That is, make every accepting path reject, and make every rejecting path accept.
3. For  $f_1 f_2$ , run  $M_1$  to completion, then run  $M_2$  to completion (in every branch of  $M_1$ ). Accept if the two machines produce the same outcome, otherwise reject.

The last construction may require some explanation. Let  $a_1, a_2$  be the number of accepting paths of  $M_1$  and  $M_2$  respectively, and similarly let  $b_1, b_2$  be the numbers of rejecting paths. Then there are  $a_1 a_2 + b_1 b_2$  accepting paths for the new machine and  $a_1 b_2 + a_2 b_1$  rejecting paths, so as a  $GapP$  machine it computes

$$a_1 a_2 - a_1 b_2 - a_2 b_1 + b_1 b_2 = (a_1 - b_1)(a_2 - b_2) = f_1(x) f_2(x). \quad \blacktriangleleft$$

Theorem 26 implies that  $C=P \subseteq PP$  because we can square and negate the gap. In other words, we can find a machine such that the gap is always negative (i.e., strictly less than half of all paths accept) unless the original machine had gap zero, in which case the gap is still zero (or, WLOG, very slightly positive). It is also worth noting that  $coC=P$  is known to equal  $NQP$ , by a result of Fenner et al. [12].

► **Definition 27.** The class  $NQP$  contains decision problems solvable by a polynomial-time quantum Turing machine (or, equivalently, a uniform, polynomial-size family of quantum circuits) where we accept if there is any nonzero amplitude on the accept state at the end of the computation.

Quantum classes with exact conditions on the amplitudes (e.g.,  $NQP$  or  $EQP$ ) tend to be very sensitive to the gate set, or QTM transition amplitudes allowed. Adleman, Demarrais, and Huang [5] are careful to define  $NQP$  for the case where the transition amplitudes are algebraic and real.

Finally, we specify computational hardness for our finite field problems using a mod  $k$  decision version of  $\#P$ .

► **Definition 28.** For any integer  $k \geq 2$ , let  $Mod_kP$  be the class of decision problems solvable by a polynomial time non-deterministic machine which rejects if the number of accepting paths is divisible by  $k$ , and accepts otherwise. In the special case  $k = 2$ ,  $Mod_kP$  is also known as “parity  $P$ ”, and denoted  $\oplus P$ .

Clearly  $P^{\#P}$  is an upper bound for  $Mod_kP$ . We are finally ready to state the main hardness result for these counting classes, namely, the celebrated theorem of Toda [26] and a subsequent generalization by Toda and Ogiwara [27]. There are many important consequences of Toda’s work, but we only require the following formulation.

► **Theorem 29 (Toda’s Theorem [26, 27]).** *Let  $A$  be one of the counting classes  $Mod_kP$ ,  $C=P$ ,  $\#P$ ,  $PP$ , or  $GapP$ . Then  $PH \subseteq BPP^A$ .*

This means in particular that, if a problem is hard for any of these classes, then there is no efficient algorithm for the problem unless  $PH$  collapses.

## C Real Construction of Toffoli (Proof of Lemma 5)

In this appendix we prove Lemma 5 from Section 3. Let us first define  $R_\theta$  as the rotation by  $\theta$  about the  $Y$ -axis. That is,  $R_\theta = \cos(\theta/2)I - i \sin(\theta/2)Y$  where  $Y$  is the Pauli  $\sigma_Y$  matrix. For our purposes, we only require the following two matrices:

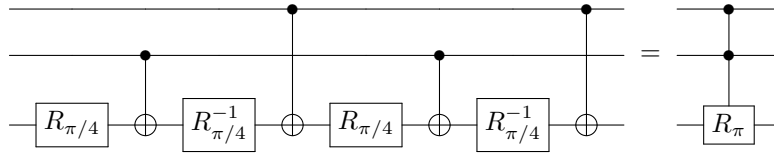
$$R_{\pi/4} = \frac{1}{2} \begin{pmatrix} \sqrt{2+\sqrt{2}} & -\sqrt{2-\sqrt{2}} \\ \sqrt{2-\sqrt{2}} & \sqrt{2+\sqrt{2}} \end{pmatrix} \quad R_\pi = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Let us now recall the statement of the lemma:

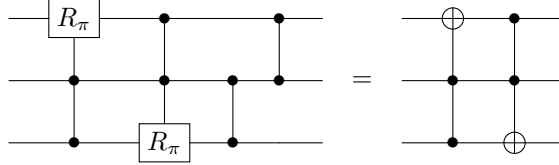
► **Lemma 5.** *There exists a circuit of CSIGN, Hadamard, and  $R_{\pi/4}$  gates which implements a Toffoli gate exactly.*

**Proof.** We construct the Toffoli gate from the CSIGN, Hadamard, and  $R_{\pi/4}$  gates in three steps:

1. **Construct a controlled-controlled- $R_\pi$  gate (CC- $R_\pi$ ) from CSIGN and  $R_{\pi/4}$  gates.** CC- $R_\pi$  is a three-qubit gate that applies  $R_\pi$  to the third qubit if the first two qubits are in the state  $|11\rangle$ . Notice that CC- $R_\pi$  is already a kind of “poor man’s” Toffoli gate. If it



■ **Figure 2** Generating  $CC-R_{\pi}$  from the CNOT and  $R_{\pi/4}$  gates.



■ **Figure 3** Generating non-affine classical gate from  $CC-R_{\pi}$  and CSIGN.

were not for the minus sign in the  $R_{\pi}$  gate, we would be done. The construction is given in Figure 2. Observe that if either of the two control qubits is zero, then any CNOT gate controlled by that qubit can be ignored. The remaining gates will clearly cancel to the identity. Furthermore, if the two control qubits are in the state  $|11\rangle$ , then on the last qubit, we apply the operation  $XR_{\pi/4}^{-1}XR_{\pi/4}XR_{\pi/4}^{-1}XR_{\pi/4}$ . Since  $XR_{\pi/4}^{-1}X = R_{\pi/4}$ ,

$$XR_{\pi/4}^{-1}XR_{\pi/4}XR_{\pi/4}^{-1}XR_{\pi/4} = R_{\pi/4}^4 = R_{\pi}.$$

Notice that this construction uses CNOT gates, but observe that a CNOT is a CSIGN gate conjugated by the Hadamard gate:

$$(I \otimes H) \text{CSIGN}(I \otimes H) = \text{CNOT}.$$

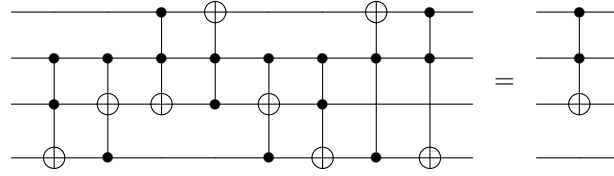
2. **Construct a non-affine classical reversible gate from CSIGN and  $CC-R_{\pi}$  gates.** By classical, we simply mean that the gate maps each computational basis state to another computational basis state (i.e., states of the form  $|x\rangle$  for  $x \in \{0, 1\}^n$ ). If this transformation is non-affine, then it suffices to generate Toffoli (perhaps with some additional ancilla qubits) by Aaronson et al. [4]. The construction is shown in Figure 3.
3. **Use the non-affine gate to generate Toffoli.** We give an explicit construction in Figure 4. Notice that the fourth qubit is an ancillary qubit starting in the  $|0\rangle$  state.<sup>18</sup> ◀

## D Gadget Details

As discussed above in Section 3 and Section 4, our results on orthogonal matrices depend on a collection of gadgets. In the real orthogonal setting (Section 3), each gadget is a real orthogonal matrix with algebraic entries, and all entries have clear, compact expressions in terms of radicals. However, in Section 4, we wish to reuse the same gadgets over finite fields, and radicals are no longer the best representation.

Instead, we will show that our (real) gadget matrices have entries in  $\mathbb{Q}(\alpha)$ , the algebraic field extension of the rational numbers by  $\alpha$ , where  $\alpha = \sqrt{2 + \sqrt{2}} + \sqrt{3 + \sqrt{6}} \approx 4.182173283$

<sup>18</sup>Indeed, this ancillary qubit is necessary because the non-affine gate in Figure 3 is an even permutation and the Toffoli gate is an odd permutation on three bits.



■ **Figure 4** Generating Toffoli gate from non-affine gate in Figure 3.

is the largest real root of irreducible polynomial

$$f(x) = x^{16} - 40x^{14} + 572x^{12} - 3736x^{10} + 11782x^8 - 17816x^6 + 11324x^4 - 1832x^2 + 1.$$

More specifically, we will write every entry as a polynomial in  $\alpha$ , with rational coefficients and degree less than 16.

This is a cumbersome representation for hand calculation, but there are some advantages. First, it eliminates any ambiguity about, for instance, which square root of 2 to use in a finite field. Second, we can check the various conditions our gadgets need to satisfy in the field  $\mathbb{Q}(\alpha)$ , and then argue that the verification generalizes to  $\mathbb{F}_p(\alpha)$ , with a few caveats. So, without further ado, we present polynomials for a set of reals which generate all the entries of our gadgets.

### D.1 Gadget entries

Since  $\pm \frac{1}{\sqrt{2}}$  are the only entries in  $D$ , our decoder gadget, we show how to express those entries as polynomials in  $\alpha$ .

$$\frac{1}{\sqrt{2}} = \frac{1}{11776} (\alpha^{14} - 53\alpha^{12} + 1077\alpha^{10} - 10561\alpha^8 + 51555\alpha^6 - 115791\alpha^4 + 95207\alpha^2 - 8379),$$

For our encoder gadget  $E$ , we also must also express  $\frac{1}{\sqrt{3}}$  as an element in  $\mathbb{Q}(\alpha)$ . Note that  $\frac{1}{\sqrt{6}}$  can be obtained as  $\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{3}}$ .

$$\frac{1}{\sqrt{3}} = \frac{1}{11776} (\alpha^{14} - 53\alpha^{12} + 1077\alpha^{10} - 10561\alpha^8 + 51555\alpha^6 - 115791\alpha^4 + 95207\alpha^2 - 8379).$$

Showing that the entries of the  $R_{\pi/4}$  gate are in  $\mathbb{Q}(\alpha)$  requires the following:

$$\begin{aligned} \sqrt{2 + \sqrt{2}} &= \frac{1}{5888} (-123\alpha^{15} + 4932\alpha^{13} - 70785\alpha^{11} + 464494\alpha^9 \\ &\quad - 1470141\alpha^7 + 2209176\alpha^5 - 1357287\alpha^3 + 193302\alpha) \end{aligned}$$

$$\begin{aligned} \sqrt{2 - \sqrt{2}} &= \frac{1}{5888} (216\alpha^{15} - 8711\alpha^{13} + 126234\alpha^{11} - 841629\alpha^9 \\ &\quad + 2733428\alpha^7 - 4270353\alpha^5 + 2799098\alpha^3 - 466411\alpha) \end{aligned}$$

Finally, we have the  $V$  gate. We already have the  $\frac{1}{3\sqrt{2}}$  in front, and the various multiples of  $\sqrt{2}$  inside, so we just need  $\sqrt{3 \pm \sqrt{6}}$  and  $\sqrt{6 \pm 2\sqrt{6}}$ . These are related by a factor of  $\sqrt{2}$ , so it suffices to give  $\sqrt{3 \pm \sqrt{6}}$ .

$$\begin{aligned} \sqrt{3 + \sqrt{6}} &= \frac{1}{5888} (123\alpha^{15} - 4932\alpha^{13} + 70785\alpha^{11} - 464494\alpha^9 \\ &\quad + 1470141\alpha^7 - 2209176\alpha^5 + 1357287\alpha^3 - 187414\alpha) \end{aligned}$$

$$\begin{aligned} \sqrt{3 - \sqrt{6}} &= \frac{1}{256} (15\alpha^{15} - 598\alpha^{13} + 8505\alpha^{11} - 55084\alpha^9 \\ &\quad + 171665\alpha^7 - 256518\alpha^5 + 161671\alpha^3 - 25624\alpha) \end{aligned}$$

## 19:26 Permanent Hardness from Linear Optics

The numbers above, combined with  $\frac{1}{2}$  and  $\frac{1}{3}$ , generate all the entries of our real orthogonal gadgets. Note that the denominators in front of the polynomials above (e.g., 11776, 5888, 256, 3, etc.) all divide  $35328 = 2^9 \cdot 3 \cdot 23$ . In other words, this representation is a bad choice for fields of characteristic 2, 3, or 23 because, in those cases, division by 35328 is division by 0. Aside from this restriction, the representation is well-defined for any field containing some root  $\alpha$  of the polynomial  $p$ .

We should not be surprised that the representation fails for fields of characteristic 2 or 3 because our matrices contain, for instance, the entries  $\frac{1}{3}$  and  $\frac{1}{\sqrt{2}}$ . We also know the permanent of an orthogonal matrix is easy to compute in fields of characteristic 2 or 3, so it is actually no surprise to find this obstacle to our hardness proof.

On the other hand, we can find no explanation for the requirement  $p \neq 23$ ; it appears to be a quirk of the algebraic number  $\alpha$ . In fact, a different choice fails for different primes. Consider  $\beta \approx 5.596386846$ , the largest real root of

$$x^{16} - 56x^{14} - 32x^{13} + 1084x^{12} + 960x^{11} - 9224x^{10} - 8928x^9 + 37702x^8 + 33920x^7 - 73736x^6 - 53216x^5 + 63932x^4 + 23488x^3 - 21560x^2 + 3808x - 191.$$

This appendix is long enough without doing all the same steps for  $\beta$ , so let us claim without proof that  $\mathbb{Q}(\beta) = \mathbb{Q}(\alpha)$ . Furthermore, when we represent the matrix entries as polynomials in  $\beta$  (we omit the details), the denominators prohibit the use of this representation for fields of characteristic 2, 3, 191, and 3313, but *not* 23. Hence, for all primes  $p$  other than 2 or 3, there is some representation that works for that prime.

## D.2 Galois Extension

We need  $\mathbb{Q}(\alpha)$  to be a Galois extension to apply Chebotarev's theorem, which we use to prove Theorem 12. Another helpful consequence is that if  $\alpha$  is in some field, then all the roots of  $f$  are also in the field since they can be expressed as polynomials in  $\alpha$ .

The most direct way to prove  $\mathbb{Q}(\alpha)$  is a Galois extension is to write all 16 roots of  $f$  in terms of  $\alpha$ . Since  $f$  is an even polynomial, half of the roots are just the negatives of the other half, so we restrict our attention to the 8 positive roots.

Root	Polynomial
0.0234	$\frac{1}{5888} (-129\alpha^{15} + 5043\alpha^{13} - 69381\alpha^{11} + 425303\alpha^9 - 1214867\alpha^7 + 1629561\alpha^5 - 919335\alpha^3 + 122941\alpha)$
0.4866	$\frac{1}{2944} (123\alpha^{15} - 4932\alpha^{13} + 70785\alpha^{11} - 464494\alpha^9 + 1470141\alpha^7 - 2209176\alpha^5 + 1357287\alpha^3 - 190358\alpha)$
1.1057	$\frac{1}{2944} (-234\alpha^{15} + 9343\alpha^{13} - 133200\alpha^{11} + 865713\alpha^9 - 2709218\alpha^7 + 4054545\alpha^5 - 2537860\alpha^3 + 391327\alpha)$
1.5073	$\frac{1}{5888} (561\alpha^{15} - 22465\alpha^{13} + 321849\alpha^{11} - 2108561\alpha^9 + 6681723\alpha^7 - 10170267\alpha^5 + 6517531\alpha^3 - 1055763\alpha)$
1.5690	$\frac{1}{5888} (-93\alpha^{15} + 3779\alpha^{13} - 55449\alpha^{11} + 377135\alpha^9 - 1263287\alpha^7 + 2061177\alpha^5 - 1441811\alpha^3 + 278997\alpha)$
2.5897	$\frac{1}{2944} (111\alpha^{15} - 4411\alpha^{13} + 62415\alpha^{11} - 401219\alpha^9 + 1239077\alpha^7 - 1845369\alpha^5 + 1180573\alpha^3 - 198025\alpha)$
3.0997	$\frac{1}{5888} (339\alpha^{15} - 13643\alpha^{13} + 197019\alpha^{11} - 1306123\alpha^9 + 4203569\alpha^7 - 6479529\alpha^5 + 4156385\alpha^3 - 653825\alpha)$
4.1821	$\alpha$

## E Approximation

Much like in Aaronson's paper [2], our hardness reductions for *exactly* computing the permanent lead naturally to hardness of approximation results as well. Approximation results comes in two flavors: additive and multiplicative. For example, Gurvits' algorithm [14] approximates the permanent of a matrix  $A$  up to  $\pm\epsilon\|A\|^n$  additive error. We will focus



strictly on multiplicative approximation. That is, the result of the approximation should be between  $\frac{1}{k} \text{per}(A)$  and  $k \text{per}(A)$  for some  $k$ .

We give approximation results only for real orthogonal matrices since it is unclear how to even define multiplicative approximation in a finite field. All of our results follow from the fact that we actually prove  $\text{GapP}$ -hardness (since we compute the gap,  $\Delta_C$ , rather than just the number of satisfying assignments). None of the results use anything specific to permanents; they are all  $\text{GapP}$  folklore, but we state them as permanent results for clarity.

► **Theorem 30.** *Suppose  $A$  is an oracle that approximates the permanent of a real orthogonal matrix to any multiplicative factor. In other words,  $A$  is an oracle for the sign (zero, positive, or negative) of the permanent. Then  $\text{GapP} \subseteq \text{FP}^A$ .*

**Proof.** We give an  $\text{FP}^A$  algorithm for computing  $\Delta_C$  for a classical circuit  $C$ . Since this problem is  $\text{GapP}$ -hard, we get  $\text{GapP} \subseteq \text{FP}^A$ .

By earlier results, we can construct a real orthogonal matrix with permanent proportional to  $\Delta_C$ . Then we can apply the oracle to compute the sign of the permanent, and hence the sign of  $\Delta_C$ . This is helpful, but we can do better.

Recall that we can add or subtract two  $\text{GapP}$  functions (see Appendix B), so for any integer  $k$ , we can construct a circuit  $C_k$  such that  $\Delta_{C_k} = \Delta_C - k$ . Then we can apply  $A$  to give us the sign of  $\Delta_{C_k}$ , or equivalently, compare  $\Delta_C$  to  $k$ . In other words, we can use  $A$  to binary search for the value of  $\Delta_C$ , which we know to be an integer in the range  $-2^n$  and  $2^n$ . ◀

Recall that  $\text{C=P}$  is the class of decision problems of solvable by a non-deterministic polynomial-time machine which accepts if it has the same number of accepting paths as rejecting paths. By Toda's theorem,  $\text{PH} \subseteq \text{BPP}^{\text{C=P}}$ .

► **Theorem 31.** *Suppose  $A$  is an oracle that approximates the absolute value of permanent of a real orthogonal matrix to any multiplicative factor. That is,  $A$  tells us whether the permanent is zero. Then  $\text{P}^{\text{C=P}} \subseteq \text{P}^A$ .*

**Proof.** The problem of computing whether  $\Delta_C = 0$  for a classical circuit  $C$  is  $\text{C=P}$ -hard. But clearly we can construct a real, orthogonal matrix from the circuit with permanent proportional to  $\Delta_C$ , and then apply  $A$  to determine if the permanent is zero, and hence whether  $\Delta_C$  is zero. Therefore  $\text{P}^{\text{C=P}} \subseteq \text{P}^A$ . ◀

Finally, we show that even a very poor approximation to the absolute value of the permanent still allows us to calculate the exact value of the permanent via a boosting argument.

► **Theorem 32.** *Suppose  $A$  is an oracle that approximates the absolute value of the permanent of an  $n \times n$  real orthogonal matrix to within a  $2^{n^{1-\varepsilon}}$  factor for some  $\varepsilon > 0$ . Then  $\text{GapP} \subseteq \text{FP}^A$ .*

**Proof.** We give an  $\text{FP}^A$  algorithm for computing  $\Delta_C$  of a classical circuit. Since this problem is  $\text{GapP}$ -hard, we get  $\text{GapP} \subseteq \text{FP}^A$ .

As in Theorem 30, we can construct a circuit  $C_k$  such that  $\Delta_{C_k} = \Delta_C - k$  for any integer  $k$ . By applying oracle  $A$  to the real orthogonal matrix corresponding to  $C_k$ , we can get a multiplicative estimate for  $|\Delta_C - k|$ . Let us assume for the moment that  $A$  gives a multiplicative approximation to within a factor of 2, and improve this to  $2^{n^{1-\varepsilon}}$  later.

Suppose we are given an interval  $[a, b]$  guaranteed to contain  $\Delta_C$ . For instance,  $\Delta_C$  is initially in  $[-2^n, 2^n]$ . Apply  $A$  to find an estimate for  $\Delta_{C_a} = \Delta_C - a$ . Suppose the

approximation we get is  $x^*$ . Then we have

$$a + \frac{1}{2}x^* \leq \Delta_C \leq a + 2x^*.$$

So  $\Delta_C$  is in the interval  $[a + \frac{1}{2}x^*, a + 2x^*] \cap [a, b]$ . One can show that this interval is longest when  $a + 2x^* = b$ , where it has length  $\frac{3}{4}(b - a)$ . Since the interval length decreases by a constant factor each step, we only need  $O(n)$  steps to shrink it from  $[-2^n, 2^n]$  to length  $< 1$ , and determine  $\Delta_C$ .

Finally, suppose we are given an oracle which gives an approximation to within a multiplicative factor  $2^{n^{1-\epsilon}}$ . Theorem 26 in Appendix B lets us construct a circuit  $C^m$  (not to be confused with  $C_k$ ) such that  $\Delta_{C^m} = (\Delta_C)^m$ . The circuit is essentially  $m$  copies of  $C$ , so we can only afford to do this for  $k$  polynomial in the size of  $C$ , otherwise our algorithm is too slow.

The point of  $C^m$  is that a factor  $\beta$  approximation to  $\Delta_{C^m}$  gives a factor  $\beta^{1/m}$  approximation of  $\Delta_C$  by taking  $m$ th roots. This is excellent for reducing a constant approximation factor, but when  $\beta$  grows with  $n$ , we must account for the fact that the size of  $C^m$  grows with  $n$  as well. In particular, the size of  $C^m$  scales with  $m$ , and the dimension of the matrix in our construction scales linearly with  $m$  as well.

So, for our algorithm to succeed, we need  $\beta(nm)^{1/m} \leq 2$  or

$$\beta(nm) \leq 2^m$$

for  $m$  a polynomial in  $n$ . Suppose we can afford  $m = n^c$  copies of  $C$ . Then we succeed when  $\beta(n^{1+c}) \leq 2^{n^c}$ , or

$$\beta(n) \leq 2^{n^{1-\frac{1}{c+1}}}.$$

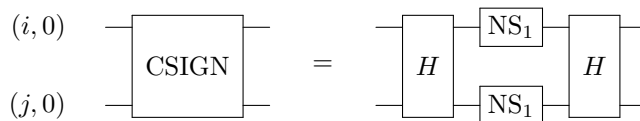
Within the scope of polynomial time algorithms, we can make  $\frac{1}{c+1}$  less than any  $\epsilon$ , and thereby handle any  $2^{n^{1-\epsilon}}$  approximation factor. ◀

The core ideas in both Theorem 30 and 32 were already noticed by Aaronson [2], but we give slightly better error bounds for the latter theorem.

## **F** Orthogonal Matrices mod 97 are #P-hard via NS<sub>1</sub>-approach

It is natural to ask whether Theorem 12 can be extended to more primes, or all primes. In other words, is there some prime  $p \neq 2, 3$  such that it is easy to compute the permanent modulo  $p$ , even though computing the permanent over  $\mathbb{F}_{p^4}$  is hard? In this appendix, we present a different construction for CSIGN gates (in fact, the construction originally used by KLM) which works in  $\mathbb{F}_{97}$ , where the  $V$  gate does not. We conclude that there is at least one more prime, namely  $p = 97$ , where the permanent is hard.

The original KLM construction builds an CSIGN gate from what they call an NS<sub>1</sub> gate, instead of directly using a  $V$  gate. Logically, the NS<sub>1</sub> gate acts on one mode and does nothing to 0 or 1 photon, but flips the sign for 2 photons. The construction of CSIGN from NS<sub>1</sub> is shown in Figure 5. If  $|1, 1\rangle$  is the input state, the Hadamard gate turns it into a linear combination of  $|2, 0\rangle$  and  $|0, 2\rangle$ , which then change phase by the NS<sub>1</sub> gate, and get recombined into  $-|1, 1\rangle$  by the Hadamard gate. Otherwise, there are not enough photons for the NS<sub>1</sub> gates to do anything, and the Hadamard gates cancel, so the gate does nothing (as a CSIGN should).



■ **Figure 5** Generating CSIGN from  $H$  and  $NS_1$  [17].

It turns out it is impossible to construct an  $NS_1$  gate without at least two postselected modes, so the KLM  $NS_1$  is a three mode gate where the last two modes start and end (via postselection) in state  $|0, 1\rangle$ . Unfortunately, the KLM  $NS_1$  gate postselects on a mode having zero photons, which is undesirable for our application. Therefore, we construct our own  $NS_1$  gate shown below. It postselects on the last two modes being  $|1, 1\rangle$  and has entirely real entries.

The gate is

$$NS_1 = \frac{1}{6} \begin{pmatrix} 6 - 18\gamma & -\sqrt{6}\sqrt{9\gamma - \sqrt{6 - 3\gamma} - 2} & -\sqrt{6}\sqrt{9\gamma + \sqrt{6 - 3\gamma} - 2} \\ -\sqrt{6}\sqrt{9\gamma - \sqrt{6 - 3\gamma} - 2} & 9\gamma + \sqrt{24 - 45\gamma} & -3\sqrt{2 - 4\gamma} \\ -\sqrt{6}\sqrt{9\gamma + \sqrt{6 - 3\gamma} - 2} & -3\sqrt{2 - 4\gamma} & 9\gamma - \sqrt{24 - 45\gamma} \end{pmatrix}$$

where  $\gamma \triangleq \frac{1}{18} (\sqrt{33} + 3) \approx 0.4858090359$ .

One can verify the following identities hold.

$$\begin{aligned} \langle 0, 1, 1 | \phi(NS_1) | 0, 1, 1 \rangle &= \gamma, \\ \langle 1, 1, 1 | \phi(NS_1) | 1, 1, 1 \rangle &= \gamma, \\ \langle 2, 1, 1 | \phi(NS_1) | 2, 1, 1 \rangle &= -\gamma. \end{aligned}$$

That is, with amplitude  $\gamma$  the postselection succeeds, and the three mode gate behaves like an  $NS_1$  gate on the first mode.


The field extension containing this gate is of higher degree than  $\mathbb{Q}(\alpha)$ , so we have not computed it explicitly. If we proved the equivalent of Theorem 12 in that extension, we would expect the density to be worse. However, this construction of an CSIGN works for at least one prime where  $V$  does not, namely  $p = 97$ .

# Retracted: Two-Player Entangled Games are NP-Hard

Anand Natarajan<sup>1</sup>

Center for Theoretical Physics, MIT, Cambridge, USA


anand@natarajans.edu

 <https://orcid.org/0000-0003-3648-3844>

Thomas Vidick<sup>2</sup>

Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, USA

vidick@cms.caltech.edu

 <https://orcid.org/0000-0002-6405-365X>

---

## Abstract

*The article, published on June 4th, 2018 in the CCC 2018 proceedings, has been retracted by agreement between the authors, the editor(s), and the publisher Schloss Dagstuhl / LIPIcs. The retraction has been agreed due to an error in the proof of the main result. This error is carried over from an error in the referenced paper “Three-player entangled XOR games are NP-hard to approximate” by Thomas Vidick (SICOMP ’16). That paper was used in an essential way to obtain the present result, and the error cannot be addressed through an erratum. See Retraction Notice on page 19.*

We show that it is NP-hard to approximate, to within an additive constant, the maximum success probability of players sharing quantum entanglement in a two-player game with classical questions of logarithmic length and classical answers of constant length. As a corollary, the inclusion  $\text{NEXP} \subseteq \text{MIP}^*$ , first shown by Ito and Vidick (FOCS’12) with three provers, holds with two provers only. The proof is based on a simpler, improved analysis of the low-degree test of Raz and Safra (STOC’97) against two entangled provers.

**2012 ACM Subject Classification** Theory of computation → Quantum complexity theory, Theory of computation → Interactive proof systems, Theory of computation → Complexity classes

**Keywords and phrases** low-degree testing, entangled nonlocal games, multi-prover interactive proof systems

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.20

## 1 Introduction

Interactive proofs are a fundamental concept in theoretical computer science, with applications to complexity theory, cryptography, and more. A classic result [19, 24] shows that interaction is a powerful resource: the class IP of problems that a polynomial-time verifier can solve with access to a single, untrusted prover is equal to PSPACE. A subsequent line of works culminating in [3] showed that even more power can be gained by interacting with *multiple* provers: the class MIP of problems decidable by a polynomial-time verifier interacting with multiple non-communicating provers is equal to NEXP. This result was an important catalyst in the discovery of the PCP theorem [2, 1], a seminal result in complexity theory that has had broad-ranging implications for hardness of approximation [9, 12]. More recently, increasingly efficient probabilistically checkable proofs (PCPs) have played a major role in the design of protocols for delegated computation of space [11] or time-bounded [18] circuits.

---

<sup>1</sup> Supported by NSF CAREER Grant CCF-1452616

<sup>2</sup> Supported by NSF CAREER Grant CCF-1553477, AFOSR YIP award number FA9550-16-1-0495, and the IQIM, an NSF Physics Frontiers Center (NSF Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028).



© Anand Natarajan and Thomas Vidick;  
licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 20; pp. 20:1–20:19

Leibniz International Proceedings in Informatics



LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



What happens when one considers a verifier that has the increased power of quantum polynomial-time computation, or provers that may use the non-local properties of quantum entanglement? In the single-prover setting, it is a highly non-trivial result that a quantum verifier, having the ability to exchange quantum messages with the prover, cannot decide more languages than a classical polynomial-time verifier:  $\text{QIP} = \text{IP} = \text{PSPACE}$  [15].

The story for multi-prover interactive proof systems is more complex. Cleve et al. [6] were the first to explore the consequences of entanglement for complexity theory. The class  $\text{MIP}^*$  is the class of languages having multi-prover interactive proofs between a classical polynomial-time verifier and quantum provers who may share entanglement. (The class  $\text{QMIP}^*$  allows a quantum verifier and quantum messages; it is known that  $\text{QMIP}^* = \text{MIP}^*$  [23].) It has been known since the early days of quantum mechanics [8], and more specifically the work of Bell [4], that allowing spatially isolated provers to perform local measurements on a shared entangled state may allow them to generate correlations between their (classical) outputs that cannot be reproduced by any local model, even using shared randomness. In general, quantum strategies have a higher success probability than classical ones, and this can affect both the completeness and soundness parameters of a proof system. As a result, the only trivial lower bound on  $\text{MIP}^*$  is  $\text{IP}$ , since the verifier can ignore one of the provers, and there are no trivial upper bounds, as the size of entangled-prover strategies can be arbitrary. Cleve et al. [6] showed that entanglement could at least in some cases lead to a collapse of a complexity class based on an interactive proofs: they studied XOR proof systems and showed that  $\oplus\text{MIP}^* \subseteq \text{PSPACE}$  (for any constant completeness-soundness gap), while it follows from Håstad’s work [12] that  $\oplus\text{MIP} = \text{NEXP}$  (for some choice of constant completeness and soundness parameters).

Nevertheless, a sequence of works established techniques to limit the power of entangled provers, eventually leading to a proof that  $\text{MIP} \subseteq \text{MIP}^*$  [14] for proof systems involving four provers, a single round of interaction, and sufficiently large, but constant, answer size. The result is a corollary of the inclusion  $\text{NEXP} \subseteq \text{MIP}^*$ , whose proof follows the same structure as Babai et al.’s celebrated proof [3] that  $\text{NEXP} \subseteq \text{MIP}$ . The main technical component of the proof is an analysis of the soundness of Babai et al.’s multilinearity test with entangled provers. The result was later refined in [26], who obtained a scaled-down version that applies to multiplayer games specified in explicit form: the main result of [26] is that it is NP-hard to approximate the value of a three-player entangled game specified in explicit form (in contrast to an interactive proof system, which is specified by a family of circuits for the verifier). The proof rests on an analysis of the soundness of the “plane-vs-point” low-degree test [22], an improvement over Babai et al.’s multilinearity test, with entangled provers.

A rather intriguing limitation of the results in [14, 26] is that they only apply to games, or interactive proof systems, with three or more entangled players, or provers. Even though in any interaction the verifier in the proof systems considered in those works only exchanges messages with two out of the three provers,<sup>3</sup> the proof seems to crucially require that the joint Hilbert space supporting the provers’ strategies can be decomposed in at least three tensor factors. Most importantly, this requirement is used in the proof of the “self-improvement lemma” that is key to control the accumulation of approximation errors in the inductive analysis of both the multilinearity and low-degree tests. Intuition for the requirement that there are three players is based on the phenomenon of monogamy of entanglement: it has been known at least since the work of Toner [25] that this kind of “embedding” of a two-player

---

<sup>3</sup> More precisely, all tests considered, including the low-degree test, take the form: (i) the verifier selects two provers at random, and calls them “Alice” and “Bob”; (ii) the verifier plays a two-prover game with Alice and Bob.

game in a three-player game can effectively limit the players' ability to take advantage of their shared entanglement, in some cases drastically lowering their maximum success probability in the game. Could it be that the two-prover entangled value of the game can be approximated in polynomial time, while the three-player entangled value is NP-hard?

We answer this question by showing that the same plane-vs-point low-degree test analyzed in [26] remains sound even when it is played with two, instead of three, entangled provers. As a consequence, we obtain the first non-trivial hardness results for the class  $\text{MIP}^*(2, 1)$  of two-prover one-round entangled proof systems. (The best prior result is hardness for inverse-exponential completeness-soundness gap [13], which cannot be amplified by a polynomial-time verifier using e.g. parallel repetition.)

► **Theorem 1.** *The inclusion  $\text{NEXP} \subseteq \text{MIP}^*(2, 1)$  holds. Furthermore, it still holds when  $\text{MIP}^*(2, 1)$  is restricted to one-round proof systems with constant answer size.*

Theorem 1 is obtained by scaling up a stronger NP-hardness result for two-player entangled projection games,<sup>4</sup> see Theorem 15 and Corollary 16 in Section 4.

Theorem 1 shows that allowing the provers to share entanglement does not weaken the power of two-prover one-round interactive proof systems. As mentioned earlier, entanglement may also have the effect of increasing the complexity of such proof systems, by allowing the verifier to implement protocols whose completeness can only be achieved by provers sharing entanglement. In fact, this is known to occur when the completeness-soundness gap is allowed to be exponentially small. In this regime, it was shown by [10] that the class  $\text{QMIP}^*$  of multi-prover interactive proof systems with a quantum verifier and messages contain  $\text{QMA}_{\text{EXP}}$ , the quantum analogue of NEXP, and subsequent works by Ji [16, 17] improved this result to show that  $\text{MIP}^*$  with exponentially small gap contains NEXP (nondeterministic doubly-exponential time). However, it remained an open question whether a similar phenomenon occurs when the completeness-soundness gap is a constant.

In a subsequent work [21], building on the soundness analysis of the two-player low-degree test presented in this paper, we were able to answer (a version of) this question in the affirmative, showing the first constant-gap QMA-hardness results for entangled-player games. Specifically, we show that it is QMA-hard, under randomized reductions, to give a constant additive approximation to the maximum success probability of a players sharing entanglement in a multiplayer game specified in explicit form. The reduction in [21] yields a game with 7 players and one round of interaction. Interestingly, the analysis of this 7 player game, which uses the quantum error-correcting code framework of [10, 16], relies essentially on the soundness of the low-degree test with *two* entangled players. This is a further application of the techniques of this work, beyond the hardness for two-player games achieved in Theorem 1.

The main ingredient needed to obtain Theorem 1, and our main technical contribution, is a soundness analysis of the plane-vs-point low-degree test in the presence of two entangled provers. The analysis that we provide is both conceptually and technically simpler than the analysis in [26]. Although our proof relies on elementary reductions from [26], we present it in a modular way which, we hope, will make it more easily accessible, and more conveniently re-usable, than the proof in [26]. In the following subsection we describe the low-degree test and give a high-level overview of our analysis.

<sup>4</sup> The reduction proceeds in a standard way by using a succinctly represented instance of the 3-SAT problem as starting point; we omit the details.

---

Out of the two provers, choose one at random to be Alice and the other to be Bob.

1. Let  $d, m$  be integer and  $q$  a prime power given as input.
  2. Select a random point  $x \in \mathbb{F}_q^m$  and two random directions  $y_1, y_2 \in \mathbb{F}_q^m$ . If  $y_1$  and  $y_2$  are not linearly independent, accept; otherwise, let  $s$  be the plane spanned by the two lines parallel to  $y_1, y_2$  passing through  $x$ .
  3. Send  $s$  to Alice and  $x$  to Bob. Receive  $g$ , a specification of a degree- $d$  polynomial restricted to  $s$ , from Alice, and  $a \in \mathbb{F}_q$  from Bob.
  4. Accept if and only if  $g(x) = a$ .
- 

■ **Figure 1** The  $(d, m, q)$ -low-degree test.

### 1.1 The low-degree test

We recall the “plane-vs-point” low-degree test from [26] in Figure 1. The test is essentially the same as the classical test from [22]. It asks one prover for the restriction of a low-degree  $m$ -variate polynomial  $g$  to a random two-dimensional subspace  $s$  of  $\mathbb{F}_q^m$ , where  $\mathbb{F}_q$  is the finite field with  $q$  elements,  $q$  a prime power, and the other prover for the evaluation of  $g$  at a random  $x \in s$ ; the prover’s answers are checked for consistency.

Since the test treats both provers symmetrically, for the purposes of the soundness analysis we may reduce to the case where the provers share a permutation-invariant state and use the same collection of measurement operators. The following states the result of our analysis of the test. It extends Theorem 3.1 in [26] to the case of two provers.<sup>5</sup> In the theorem, we use the notation  $\langle A, B \rangle_\Psi$  for  $\langle \Psi | A \otimes B | \Psi \rangle$ .

► **Theorem 2.** *There exists a  $\delta = \text{poly}(\varepsilon)$  and a constant  $c > 0$  such that the following holds. Let  $\varepsilon > 0$ ,  $m, d$  integers, and  $q$  a prime power such that  $q \geq (dm/\varepsilon)^c$ . For any strategy for the players using entangled state  $|\Psi\rangle$  and projective measurements  $\{A_s^r\}_r$  that succeeds in the  $(d, m, q)$ -low-degree test with probability at least  $1 - \varepsilon$ , there exists a POVM  $\{S^g\}_g$ , where  $g$  ranges over  $m$ -variate polynomials over  $\mathbb{F}_q$  of total degree at most  $d$ , such that the following hold:*

1. *Approximate consistency with  $A$ :*

$$\mathbb{E}_s \sum_g \sum_{r \neq g|_s} \langle A_s^r, S^g \rangle_\Psi \leq \delta,$$

where the expectation is over a random two-dimensional subspace  $s$  of  $\mathbb{F}_q^m$ , as chosen by the verifier in the test;

2. *Self-consistency:*

$$\sum_g \langle S^g, (\text{Id} - S^g) \rangle_\Psi \leq \delta.$$

The proof of Theorem 2 follows the same structure as the proof of Theorem 3.1 in [26]. The proof is by induction on the number of variables  $m$ . The base case  $m = 2$  is trivial, since there is a single subspace  $s$ , and the provers’ associated POVM  $\{A^r\}$  can directly play the role of  $\{S^g\}$  in the theorem. Suppose then that the theorem is true for a value  $(m - 1)$  such that  $m - 1 \geq 2$ . To show that the theorem holds for  $m$  there are three main steps, which mirror the classical analysis of the low-degree test:

---

<sup>5</sup> The self-consistency condition is not explicitly stated in [26] but (as we will show) it follows easily from the proof.



1. (Section 6.3 of [26]) By the induction hypothesis, for every  $(m - 1)$ -dimension hyperplane  $s$  in  $\mathbb{F}_q^m$  there is a POVM  $\{Q_s^g\}_g$  with outcomes  $g$  in the set of degree- $d$  polynomials on  $s$ , such that on average over the choice of a uniformly random  $s$  and  $x \in s$  the POVM  $\{Q_s^g\}$  is consistent with  $\{A_x^a\}$ .
2. (Section 6.4 of [26]) For any  $k \geq 1$ , measurements  $\{Q_s^g\}_g$  associated with  $k$  parallel subspaces  $s_1, \dots, s_k$  are “pasted” together to yield a combined measurement  $\{Q_{(s_i)}^{(g_i)}\}$  that returns a  $k$ -tuple of degree- $d$  polynomials  $g_i$  defined on  $s_i$ . This is proved by induction on  $k$ .
3. (Section 6.5 of [26]) Finally, taking  $k$  to be sufficiently large compared to  $d$ , the measurement  $\{Q_{(s_i)}^{(g_i)}\}$  is consolidated into a single global measurement  $\{S^g\}$  that satisfies the conclusion of the theorem for the  $m$ -variate case.

These three steps remain unchanged in the current proof. At only very few places in [26] is the presence of three provers used; in most cases this is only a matter of convenience and is easily avoided. For completeness, in Appendix A we explicitly list those places and how the use of three provers can be avoided.

As already mentioned the critical point in the proof where three provers, or rather the existence of three tensor factors in the provers’ Hilbert space, is used, is to control the error increase throughout the induction. As shown by the analysis, if the measurements  $\{Q_s^g\}$  produced by the induction hypothesis are  $\delta$ -consistent with  $\{A_x^a\}$ , then the resulting  $S^g$  will be  $O(\delta^c)$ -consistent with the same  $\{A_x^a\}$ , for some constant  $c < 1$ . For poly-logarithmic  $m$  such an increase is unmanageable.

The key step in the analysis consists in establishing a “self-improvement lemma”, which resets the consistency error to some constant baseline at each step of the induction. This is called the “consolidation procedure” in [26]. A similar self-improvement was already at the heart of Babai et al.’s proof of  $\text{MIP} \subseteq \text{NEXP}$ ; variants thereof have found uses outside of complexity theory, such as in property testing.

Our main technical contribution is a simpler, self-contained proof of a variant of the consolidation procedure from [26] (stated as Proposition 5.8 in that paper), which applies to strategies with two provers only. The procedure shows that the consistency error sustained by any POVM, when measured against a structure called a “robust triple” in [26], can be automatically improved. Our variant is based on a simpler notion than the robust triples from [26], that we call “global consistency”. We believe that our formulation of self-improvement, and its analysis (which crucially relies on semidefinite duality), should be of broad interest. At a high level, the result relies on a procedure that, given a collection of positive semidefinite operators  $\{A_i\}$ , identifies a measurement  $\{T_i\}$ , i.e.  $T_i \geq 0$  and  $\sum_i T_i = \text{Id}$ , that “optimally coincides” with the  $\{A_i\}$  (see Lemma 13 for a precise formulation).

Throughout we assume familiarity with the notation and proof structure from [26], though we recall the most important notions in Section 2. In particular we formally define robust triples and global consistency, and show that the former notion implies the latter, so that our result can be directly used in lieu of Proposition 5.8 in the analysis of [26]. In Section 3 we prove our replacement for Proposition 5.8, Proposition 12. The proof of (the scaled-down version of) Theorem 1 follows from the analysis of the test using similar reductions as in [26]; we briefly explain how in Section 4.



## 2 Preliminaries

### 2.1 Notation

We use  $\mathcal{H}$  to denote a finite-dimensional Hilbert space, and  $L(\mathcal{H})$  for the linear operators on  $\mathcal{H}$ . Subscripts  $\mathcal{H}_A, \mathcal{H}_B$  indicate distinct spaces. For  $|\Psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$  and  $A \in L(\mathcal{H}_A)$ ,  $B \in L(\mathcal{H}_B)$  we write  $\langle A, B \rangle_\Psi = \langle \Psi | A \otimes B | \Psi \rangle$ . Note that we do not conjugate  $A$  or  $B$ . Given two families of operators  $\{A_x^a\}$  and  $\{B_x^a\}$  on  $\mathcal{H}_A$ , where  $x \in \mathcal{X}$  and  $a \in \mathcal{A}$  range over finite sets, and  $0 \leq \delta \leq 1$ , we write  $A_x^a \approx_\delta B_x^a$  for

$$\mathbb{E}_x \sum_a \langle (A_x^a - B_x^a)^2, \text{Id} \rangle_\Psi = O(\delta).$$

The expectation over  $x$  will usually be taken with respect to the uniform distribution. The distinction between taking an expectation (over  $x$ ) or a summation (over  $a$ ) will always be clear from context.

### 2.2 Measurements

Throughout, we consider a bipartite state  $|\Psi\rangle \in \mathcal{H} \otimes \mathcal{H}$  assumed to be invariant under permutation of the two registers. All operators we consider act on the finite-dimensional space  $\mathcal{H}$ .

► **Definition 3.** A *sub-measurement*  $\{M^a\}_a$  is a collection of positive semidefinite operators satisfying  $M = \sum_a M^a \leq \text{Id}$ . We say that a sub-measurement is  $\eta$ -complete if

$$\langle M, \text{Id} \rangle_\Psi \geq 1 - \eta;$$

$\eta$  is called the *completeness error*. If  $M = \text{Id}$  then we say that  $\{M^a\}_a$  is a measurement, in which case the completeness error is zero.<sup>6</sup>

The following definition appears in [26].

► **Definition 4.** Let  $\mathcal{X}$  and  $\mathcal{A}$  be finite sets. Let  $\{M_x^a\}_a$  be a family of sub-measurements indexed by  $x \in \mathcal{X}$  and with outcomes  $a \in \mathcal{A}$ . For each  $x$ , let  $M_x = \sum_a M_x^a$ . We say that  $\{M_x^a\}$  is

■  $\varepsilon$ -self-consistent if

$$\mathbb{E}_x \sum_{a \neq a'} \langle M_x^a, M_x^{a'} \rangle_\Psi \leq \varepsilon,$$

■  $\gamma$ -projective if

$$\mathbb{E}_x \langle M_x, (\text{Id} - M_x) \rangle_\Psi \leq \gamma.$$

■ Let  $\{T^g\}$  be a sub-measurement with outcomes in the set of all functions  $g: \mathcal{X} \rightarrow \mathcal{A}$ . We say that  $\{M_x^a\}$  and  $\{T^g\}$  are  $\delta$ -consistent if

$$\mathbb{E}_x \sum_{g, a: a \neq g(x)} \langle T^g, M_x^a \rangle_\Psi \leq \delta.$$

<sup>6</sup> The converse does not necessarily hold, as  $|\Psi\rangle$  may not have full support.

We consider families of functions such that distinct functions have few points of intersection. The following definition is the reformulation of the definition of an error-correcting code, that is adapted to our notation using functions (where the codeword associated to a function is the evaluation table of the function, and vice-versa).

► **Definition 5.** Let  $\mathcal{X}$  and  $\mathcal{A}$  be finite sets,  $\mathcal{G}$  a set of functions from  $\mathcal{X}$  to  $\mathcal{A}$ , and  $0 \leq \kappa \leq 1$ . We say that  $(\mathcal{X}, \mathcal{A}, \mathcal{G})$  is  $\kappa$ -structured if for any two distinct  $g, g' \in \mathcal{G}$ ,

$$\Pr_{x \in \mathcal{X}} (g(x) = g'(x)) \leq \kappa,$$

where the probability is taken under the uniform distribution on  $\mathcal{X}$ .

The following lemma states useful properties of consistency.

► **Lemma 6.** Let  $(\mathcal{X}, \mathcal{A}, \mathcal{G})$  be  $\kappa$ -structured. Let  $\{A_x^a\}_{a \in \mathcal{A}}$  be a family of measurements indexed by  $x \in \mathcal{X}$  that is  $\varepsilon$ -self-consistent. Let  $\{T^g\}_{g \in \mathcal{G}}$  be a sub-measurement that is  $\delta$ -consistent with  $\{A_x^a\}$ . Then

- $\{T^g\}$  is  $\delta'$ -self-consistent, for  $\delta' = O(\sqrt{\varepsilon} + \sqrt{\delta} + \kappa)$ ;
- Let  $T = \sum_g T^g$ , and suppose  $\{T^g\}$  is  $\gamma$ -projective. Then

$$TA_x^a \approx_{\sqrt{\varepsilon} + \sqrt{\delta} + \gamma + \kappa} A_x^a T.$$

**Proof.** We sketch the proof. For the first item,

$$\begin{aligned} \sum_{g \neq g'} \langle T^g, T^{g'} \rangle_{\Psi} &= \mathbb{E}_x \sum_a \sum_{g \neq g'} \langle T^g, T^{g'} A_x^a \rangle_{\Psi} \\ &\approx \sqrt{\delta} \mathbb{E}_x \sum_{g \neq g'} \langle T^g, T^{g'} A_x^{g(x)} \rangle_{\Psi} \\ &\approx \sqrt{\varepsilon} \mathbb{E}_x \sum_{g \neq g'} \langle T^g A_x^{g(x)}, T^{g'} \rangle_{\Psi} \\ &\approx \sqrt{\delta} \mathbb{E}_x \sum_{g \neq g'} \mathbf{1}_{g(x)=g'(x)} \langle T^g A_x^{g(x)}, T^{g'} \rangle_{\Psi} \\ &\approx \sqrt{\delta} \mathbb{E}_x \sum_{g \neq g'} \mathbf{1}_{g(x)=g'(x)} \langle T^g, T^{g'} \rangle_{\Psi} \\ &\leq \kappa. \end{aligned}$$

For the second item, it suffices to lower bound

$$\begin{aligned} \mathbb{E}_x \sum_a \langle TA_x^a TA_x^a, \text{Id} \rangle_{\Psi} &\approx \sqrt{\varepsilon} \mathbb{E}_x \sum_a \sum_g \langle TA_x^a T^g, A_x^a \rangle_{\Psi} \\ &\approx \sqrt{\delta} \mathbb{E}_x \sum_a \sum_g \langle TA_x^a T^g, A_x^{g(x)} \rangle_{\Psi} \\ &\approx \sqrt{\delta} \mathbb{E}_x \sum_{a, a'} \sum_g \langle TA_x^a T^g, A_x^{a'} \rangle_{\Psi} \\ &= \langle T^2, \text{Id} \rangle_{\Psi}. \end{aligned}$$

The claimed bound then follows by expanding  $\mathbb{E}_x \sum_a (TA_x^a - A_x^a T)^2$  and regrouping terms. ◀

### 2.3 Global consistency

The analysis of the low-degree test amounts to arguing that a set of measurement operators which produce outcomes that are locally consistent can be combined into a single measurement which returns a global object consistent with each of the local measurements: it is possible to recombine local views. In [26] the notion of local consistency used is called a “robust triple”. For convenience we recall the definition.

► **Definition 7** (Definition 5.2 in [26]). Let  $G = (V, E)$  be a graph,  $S$  a finite set,  $\mathcal{G} \subseteq \{g : V \rightarrow S\}$  a set of functions and for every  $v \in V$ ,  $\{A_v^a\}_{a \in S}$  a measurement with outcomes in  $S$ . Given  $\delta > 0$  and  $0 < \mu \leq 1$ , we say that  $(G, \{A_v^a\}, \mathcal{G})$  is a  $(\delta, \mu)$ -robust triple if:

1. (self-consistency) The family of measurements  $\{A_v^a\}$  is  $\delta$ -self-consistent;
2. (small intersection)  $(V, S, \mathcal{G})$  is  $\delta$ -structured;
3. (stability) For any sub-measurement  $\{R^g\}_{g \in \mathcal{G}}$  it holds that

$$\mathbb{E}_{v \in V} \mathbb{E}_{v' \in N(v)} \sum_g \langle R^g, (A_v^{g(v)} - A_{v'}^{g(v')})^2 \rangle_\Psi \leq \delta,$$

where  $N(v)$  is the set of neighbors of  $v$  in  $G$ ;

4. (expansion)  $G$  has mixing time  $O(\mu^{-1})$ . Precisely, if for any  $v \in V$  we let  $p_k(v)$  denote the distribution on  $V$  that results from starting a  $k$ -step random walk at  $v$ , then for any  $\delta > 0$  and some  $k = O(\log(1/\delta) \log(1/\mu))$  it holds that  $\mathbb{E}_{v \in V} \|p_k(v) - |V|^{-1}\|_1 \leq \delta$ .

We observe that the only way in which items 3. and 4. from the definition are used for the self-improvement results is through [26, Claim 5.3], which states the following.

► **Claim 8** (Claim 5.3 in [26]). *Suppose  $(G, A, \mathcal{G})_\Psi$  is a  $(\delta, \mu)$ -robust triple. Then there exists a  $\delta' = O(\delta^{1/2} \log^2(1/\delta) \log^2(1/\mu))$  such that for any sub-measurement  $\{R^g\}_{g \in \mathcal{G}}$ ,*

$$\sum_g \langle R^g, A^g - (A^g)^2 \rangle_\Psi \leq \delta', \quad (1)$$

where  $A^g = \mathbb{E}_{v \in V} A_v^{g(v)}$ .

It is more direct, and more general, to use condition (1) directly as part of the definition, as this allows us to set aside any notion of an expanding graph.

► **Definition 9.** Let  $(\mathcal{X}, \mathcal{A}, \mathcal{G})$  be  $\kappa$ -structured. Let  $\{A_x^a\}_{a \in \mathcal{A}}$  be a family of measurements indexed by  $x \in \mathcal{X}$  and with outcomes  $a \in \mathcal{A}$ . For  $g \in \mathcal{G}$ , let  $A^g = \mathbb{E}_x A_x^{g(x)}$ . Let  $|\Psi\rangle$  be a permutation-invariant bipartite state. For  $0 \leq \varepsilon, \delta \leq 1$  we say that  $(\{A_x^a\}, \mathcal{G})$  is  $(\varepsilon, \delta)$ -globally consistent on  $|\Psi\rangle$  if:

1.  $\kappa = O(\varepsilon)$ ;
2. The family  $\{A_x^a\}$  is  $\varepsilon$ -self-consistent;
3. There exists a positive semidefinite operator  $Z$  such that

$$\forall g \in \mathcal{G}, 0 \leq A^g - (A^g)^2 \leq Z, \quad \text{and} \quad \langle Z, \text{Id} \rangle_\Psi \leq \delta.$$

It is not hard to verify that condition 3. in the definition is equivalent to (1). This can be seen by writing the bound  $\delta$  in the condition as the optimum of a semidefinite program, and taking the dual. This is done in a similar way to the analysis of the semidefinite program (2). The only difference is that the latter considers consistency when the state  $|\Psi\rangle$  is maximally entangled. Formally, we have the following lemma.

► **Lemma 10.** *Let  $|\Psi\rangle \in \mathcal{H} \otimes \mathcal{H}$  be a state invariant under permutation of its two registers, such that the reduced density of  $|\Psi\rangle$  on either register has full support. Let  $\{A_i\}$  be a family of positive semidefinite operators on  $\mathcal{H}$  with  $A_i \leq \text{Id}$  for all  $i$ . Then the following primal and dual semidefinite program satisfy strong duality, and hence have the same optimum value:*

Primal SDP

$$\sup \sum_i \langle T_i, A_i \rangle_\Psi$$

$$\text{s.t. } T_i \geq 0 \quad \forall i,$$

$$\sum_i T_i \leq \text{Id}.$$

Dual SDP

$$\inf \langle Z, \text{Id} \rangle_\Psi$$

$$\text{s.t. } Z \geq A_i \quad \forall i,$$

$$Z \geq 0.$$

**Proof.** Both the primal and dual are strictly feasible, as can be seen by taking e.g.  $T_i \propto \text{Id}$  such that  $\sum_i T_i = \text{Id}/2$ , and  $Z = 2\text{Id}$ . ◀

Taking  $A_i$  in Lemma 10 to equal  $A^g - (A^g)^2$ , the primal value being less than  $\delta'$  is equivalent to (1), while the dual value being less than  $\delta'$  is equivalent to item 3. in Definition 9.

For later use we note that self-consistency of  $\{A_x^a\}$  implies self-consistency of the operators  $A^g$  introduced in Definition 9, in the following sense.

► **Lemma 11.** *Let  $\{A_x^a\}$  be a family of measurements that is  $\varepsilon$ -self-consistent. Then for any sub-measurement  $\{R^g\}$ ,*

$$\sum_g \langle A^g, R^g \rangle_\Psi \approx_{\sqrt{\varepsilon}} \sum_g \langle \text{Id}, R^g A^g \rangle_\Psi.$$

**Proof.** Write

$$\begin{aligned} \sum_g \langle A^g, R^g \rangle_\Psi &= \sum_g \mathbb{E}_x \langle A_x^{g(x)}, R^g \rangle_\Psi \\ &= \sum_{g,a} \mathbb{E}_x \langle A_x^{g(x)}, R^g A_x^a \rangle_\Psi \\ &\approx_{\sqrt{\varepsilon}} \sum_g \mathbb{E}_x \langle A_x^{g(x)}, R^g A_x^{g(x)} \rangle_\Psi \\ &\approx_{\sqrt{\varepsilon}} \sum_g \mathbb{E}_x \langle \text{Id}, R^g A_x^{g(x)} \rangle_\Psi. \end{aligned}$$

### 3 Self-improvement with two provers

The main result on self-improvement from [26] is stated as Proposition 5.8 in that paper. Our main technical result, Proposition 12 below, improves upon Proposition 5.8 in the following respects:

- Proposition 12 allows performing self-improvement with two provers only;
- Proposition 12 only requires the notion of consistency introduced in Definition 9, which as argued in Section 2.3 is less restrictive than the notion of robust triple used in [26];

■ The proof of Proposition 12 is simpler and yields better parameters.

We state the proposition and give its proof here. In Section 4 we show how the proposition is used to obtain the hardness results.

► **Proposition 12.** *There exists universal constants  $\varepsilon_0, \delta_0, t_0 > 0$  such that the following holds. Let  $(\mathcal{X}, \mathcal{A}, \mathcal{G})$  be  $\kappa$ -structured. Let  $\{A_x^a\}_{a \in \mathcal{A}}$  be a family of measurements indexed by  $x \in \mathcal{X}$ , and  $|\Psi\rangle$  a bipartite permutation-invariant state. Suppose that the following conditions hold:*

1.  $(\{A_x^a\}, \mathcal{G})$  is  $(\varepsilon, \delta)$ -globally consistent on  $|\Psi\rangle$ , for some  $0 \leq \varepsilon \leq \varepsilon_0, 0 \leq \delta \leq \delta_0$ ;
2. *There exists a function  $t = t(\varepsilon', \delta')$  and  $\varepsilon'_0, \delta'_0 > 0$  such that for any  $0 \leq \varepsilon' \leq \varepsilon'_0$  and  $0 \leq \delta' \leq \delta'_0$  it holds that  $t(\varepsilon', \delta') \leq t_0$ , and such that the following holds. For any  $(\varepsilon', \delta')$  and state  $|\Phi\rangle$  such that  $(\{A_x^a\}, \mathcal{G})$  is  $(\varepsilon', \delta')$ -globally consistent on  $|\Phi\rangle$ , there exists a measurement  $\{Q^g\}_{g \in \mathcal{G}}$  that is  $t(\varepsilon', \delta')$ -consistent with  $\{A_x^a\}$ .*

*Then there exists a measurement  $\{R^g\}_{g \in \mathcal{G}}$  that is  $\delta'$ -consistent with  $\{A_x^a\}$ , for some  $\delta' = O(\sqrt{r(\varepsilon, \delta)})$ , where  $r(\varepsilon, \delta)$  is the function defined in Lemma 13.*

The key “improvement” provided by the proposition is that, while the function  $t$  is only assumed to be bounded by a fixed constant for sufficiently small values of the arguments, the proposition returns a measurement  $\{R^g\}$  that has an explicit consistency  $\delta'$  with  $\{A_x^a\}$ , where  $\delta'$  is polynomial in  $\varepsilon$  and  $\delta$ , irrespective of  $t$  (indeed  $t$  need not approach 0 as  $\varepsilon, \delta$  approach 0).

We note that, in our language, [26, Proposition 5.8] considers a family of globally consistent pairs  $(\{A_{t,x}^a\}, \mathcal{G}_t)$ , parametrized by some finite set  $t \in T$ , and makes both the assumptions and the conclusions of Proposition 12 in an averaged sense, for uniformly random  $t \in T$ . For simplicity we state and prove the proposition for  $|T| = 1$ . The case of general  $T$  is needed for the inductive application of the Proposition towards the proof of Theorem 2. We sketched the inductive step in the introduction. We refer to [26] for details of the derivation of Theorem 2 from Proposition 12, which is identical to the derivation of [26, Theorem 3.1] from [26, Proposition 5.8], up to minor modifications that we review in Appendix A.

The main step in the proof of the proposition is provided by the following lemma, which is analogous to [26, Claim 5.4]. The semidefinite program considered in the proof of the lemma, and its analysis, are our main points of departure from the proof in [26]. Indeed, the proof of an upper bound on the completeness error of the sub-measurement  $\{S^g\}$  constructed in the proof of the lemma is the main point where the existence of a three-fold tensor product decomposition of the Hilbert space is most crucially used in [26].

► **Lemma 13.** *There exists a function  $r(\varepsilon, \delta) = O(\sqrt{\varepsilon} + \sqrt{\delta})$  such that the following holds for all  $0 \leq \varepsilon, \delta, \eta \leq 1$ . Let  $(\mathcal{X}, \mathcal{A}, \mathcal{G})$  be  $\kappa$ -structured. Let  $\{A_x^a\}_{a \in \mathcal{A}}$  be a family of measurements indexed by  $x \in \mathcal{X}$ . Let  $|\Psi\rangle$  be a permutation-invariant bipartite state and assume  $(\{A_x^a\}, \mathcal{G})$  are  $(\varepsilon, \delta)$ -globally consistent on  $|\Psi\rangle$ . Let  $\{Q^g\}_{g \in \mathcal{G}}$  be a sub-measurement that is  $\eta$ -consistent with  $\{A_x^a\}$  on  $|\Psi\rangle$ . Then there exists a sub-measurement  $\{S^g\}$  that is  $r(\varepsilon, \delta)$ -consistent with  $\{A_x^a\}$  and projective and has completeness error*

$$\langle \text{Id} - S, \text{Id} \rangle_\Psi \leq \langle \text{Id} - Q, \text{Id} \rangle_\Psi + \eta + r(\varepsilon, \delta).$$

**Proof.** For  $g \in \mathcal{G}$ , let  $A^g = E_x A_x^{g(x)}$ . We consider the following primal and dual semidefinite program, obtained from the semidefinite program in Lemma 10 by setting  $A_i$  to  $A^g$  and formally replacing the state  $|\Psi\rangle$  appearing in the SDP by the maximally entangled state<sup>7</sup>.

<sup>7</sup> Note that we are not assuming that the state  $|\Psi\rangle$  appearing in the hypothesis of Lemma 13 is maximally entangled. The purpose of defining the SDP (2) without reference to the state  $|\Psi\rangle$  is to make the resulting complementary slackness conditions (4) easier to work with.

The primal becomes

$$\begin{aligned} \omega = \sup \quad & \sum_g \text{Tr}(T^g A^g) \\ \text{s.t.} \quad & T^g \geq 0 \quad \forall g \in \mathcal{G}, \\ & \sum_g T^g \leq \text{Id}, \end{aligned} \quad (2)$$

and the dual

$$\begin{aligned} \min \quad & \text{Tr}(Z) \\ \text{s.t.} \quad & Z \geq A^g \quad \forall g \in \mathcal{G}, \\ & Z \geq 0. \end{aligned} \quad (3)$$

As shown in Lemma 10 both the primal and dual are strictly feasible, so that strong duality holds. Let  $\{T^g\}$  be an optimal primal solution. Without loss of generality,  $\sum_g T^g = \text{Id}$ , as any solution such that  $(\text{Id} - \sum_g T^g)A^{g'} \neq 0$  for any  $g'$  is clearly not optimal. The complementary slackness conditions, which follow from the KKT conditions for optimality, immediately imply

$$T^g Z = T^g A^g \quad \forall g \in \mathcal{G}. \quad (4)$$

For each  $g \in \mathcal{G}$  let

$$S^g = \mathbb{E}_x A_x^{g(x)} T^g A_x^{g(x)}.$$

Then  $\{S^g\}$  is a sub-measurement. We show that  $S^g$  satisfies the desired consistency, projectivity and completeness properties.

(i) *Consistency*: We have that

$$\mathbb{E}_x \sum_g \sum_{a \neq g(x)} \langle S^g, A_x^a \rangle_\Psi = \sum_g \langle S^g, (\text{Id} - A^g) \rangle_\Psi.$$

Using self-consistency of  $\{A_x^a\}$ ,

$$\begin{aligned} \sum_g \langle S^g, \text{Id} \rangle_\Psi &= \mathbb{E}_x \sum_g \langle A_x^{g(x)} T^g A_x^{g(x)}, \text{Id} \rangle_\Psi \\ &\approx_{\sqrt{\varepsilon}} \mathbb{E}_x \sum_g \langle T^g, A_x^{g(x)} \rangle_\Psi \\ &= \sum_g \langle T^g, A^g \rangle_\Psi. \end{aligned} \quad (5)$$

Similarly,

$$\begin{aligned} \sum_g \langle S^g, A^g \rangle_\Psi &= \mathbb{E}_x \sum_g \langle A_x^{g(x)} T^g A_x^{g(x)}, A^g \rangle_\Psi \\ &\approx_{\sqrt{\varepsilon}} \mathbb{E}_x \sum_g \langle T^g, A_x^{g(x)} A^g A_x^{g(x)} \rangle_\Psi. \end{aligned} \quad (6)$$

Using the Cauchy-Schwarz inequality,

$$\begin{aligned}
 \mathbb{E}_x \sum_g \langle T^g, (A^g - A_x^{g(x)}) A^g A_x^{g(x)} \rangle_\Psi &\leq \left( \mathbb{E}_x \sum_g \langle T^g, A_x^{g(x)} (A^g)^2 A_x^{g(x)} \rangle_\Psi \right)^{\frac{1}{2}} \\
 &\quad \cdot \left( \mathbb{E}_x \sum_g \langle T^g, (A^g - A_x^{g(x)})^2 \rangle_\Psi \right)^{\frac{1}{2}} \\
 &\leq \left( \sum_g \langle T^g, (A^g - (A^g)^2) \rangle_\Psi \right)^{\frac{1}{2}} \\
 &\leq \sqrt{\delta}, \tag{7}
 \end{aligned}$$

where the second inequality uses  $A_x^{g(x)} (A^g)^2 A_x^{g(x)} \leq \text{Id}$  for the first term, and expands the square and uses  $(A_x^{g(x)})^2 \leq A_x^{g(x)}$  for the second term, and the last inequality follows from item 3. in the definition of globally consistent. Combined with (5) and (6), we have shown

$$\mathbb{E}_x \sum_g \sum_{a \neq g(x)} \langle S^g, A_x^a \rangle_\Psi \approx_{\sqrt{\varepsilon} + \sqrt{\delta}} \sum_g \langle T^g, (A^g - (A^g)^3) \rangle_\Psi. \tag{8}$$

Writing

$$\begin{aligned}
 A^g - (A^g)^3 &= A^g - (A^g)^2 + \sqrt{A^g} (A^g - (A^g)^2) \sqrt{A^g} \\
 &\leq 2(A^g - (A^g)^2),
 \end{aligned}$$

since all terms commute and  $(A^g)^2 \leq A^g \leq \text{Id}$ , using item 3. in the definition of globally consistent the right-hand side of (8) is at most  $2\delta$ .

(ii) *Completeness:*

$$\begin{aligned}
 \sum_g \langle S^g, \text{Id} \rangle_\Psi &= \mathbb{E}_x \sum_g \langle A_x^{g(x)} T^g A_x^{g(x)}, \text{Id} \rangle_\Psi \\
 &\approx_{\sqrt{\varepsilon}} \mathbb{E}_x \sum_g \langle T^g, A_x^{g(x)} \rangle_\Psi \\
 &\approx_{\sqrt{\varepsilon}} \sum_g \langle T^g A^g, \text{Id} \rangle_\Psi \\
 &= \sum_g \langle T^g Z, \text{Id} \rangle_\Psi \\
 &= \langle Z, \text{Id} \rangle_\Psi,
 \end{aligned}$$

where the third line uses Lemma 11 and the penultimate equality follows from (4), and for the last we used  $\sum_g T^g = \text{Id}$ . We establish a lower bound on this last expression by introducing  $\{Q^g\}$ :

$$\begin{aligned}
 \langle Q, \text{Id} \rangle_\Psi - \eta &\leq \sum_g \langle Q^g, A^g \rangle_\Psi \\
 &\leq \sum_g \langle Q^g, Z \rangle_\Psi \\
 &\leq \langle \text{Id}, Z \rangle_\Psi,
 \end{aligned}$$

where the second inequality uses the dual constraint (3), and the third uses  $\sum_g Q^g \leq \text{Id}$ . It follows that

$$\sum_g \langle S^g, \text{Id} \rangle_\Psi \geq \langle Q, \text{Id} \rangle_\Psi - \eta - O(\sqrt{\varepsilon}).$$

(iii) *Projectivity*: By proceeding exactly as in (7), we can show

$$\begin{aligned}
\langle S, S \rangle_{\Psi} &= \sum_g \mathbb{E}_x \langle A_x^{g(x)} T^g A_x^{g(x)}, S \rangle_{\Psi} \\
&\approx_{\sqrt{\varepsilon} + \sqrt{\delta}} \sum_g \langle A^g T^g A^g, S \rangle_{\Psi} \\
&= \sum_{g, g'} \mathbb{E}_x \langle A^g T^g A^g, A_x^{g'(x)} T^{g'} A_x^{g'(x)} \rangle_{\Psi} \\
&\approx_{\sqrt{\varepsilon} + \sqrt{\delta}} \sum_{g, g'} \mathbb{E}_x \langle A_x^{g(x)} T^g A_x^{g(x)}, A_x^{g'(x)} T^{g'} A_x^{g'(x)} \rangle_{\Psi}.
\end{aligned}$$

Using self-consistency of  $\{A_x^a\}$ , from the above we get

$$\begin{aligned}
\langle S, S \rangle_{\Psi} &\approx_{\sqrt{\varepsilon} + \sqrt{\delta}} \sum_{g, g'} \mathbb{E}_x \langle T^g A_x^{g(x)}, A_x^{g'(x)} T^{g'} A_x^{g'(x)} \rangle_{\Psi} \\
&\approx_{\sqrt{\varepsilon} + \sqrt{\delta}} \sum_{g, g'} \langle T^g A^g, S^{g'} \rangle_{\Psi} \\
&= \langle Z, S \rangle_{\Psi}, \tag{9}
\end{aligned}$$

where the second line again uses similar arguments as (7) and the last line uses (4) and  $\sum_g T^g = \text{Id}$ . Using the dual constraint (3), we deduce

$$\begin{aligned}
\langle S, S \rangle_{\Psi} &\geq \sum_g \langle A^g, S^g \rangle_{\Psi} - O(\sqrt{\varepsilon} + \sqrt{\delta}) \\
&\approx_{\sqrt{\varepsilon} + \sqrt{\delta}} \langle S, \text{Id} \rangle_{\Psi},
\end{aligned}$$

where the second line follows from consistency of  $\{S^g\}$  and  $\{A_x^a\}$  shown in item (i). ◀

Based on Lemma 13, we give the proof of Proposition 12.

**Proof of Proposition 12.** Let  $\varepsilon, \delta$  be as in condition 1., and  $\{Q^g\}$  be the measurement whose existence follows from condition 2. in the proposition, when  $|\Phi\rangle = |\Psi\rangle$  and  $\varepsilon', \delta' = \varepsilon, \delta$ . By applying Lemma 13 to the state  $|\Psi\rangle$  and measurements  $\{A_x^a\}$  and  $\{Q^g\}$  we obtain a sub-measurement  $\{S^g\}$  that is  $\xi = r(\varepsilon, \delta)$ -projective and consistent with  $\{A_x^a\}$ . Among all sub-measurements that are  $\xi$ -projective and consistent with  $\{A_x^a\}$ , let  $\{T^g\}$  be one that minimizes the completeness error  $\theta = \langle \text{Id} - T, \text{Id} \rangle$ . Provided  $\varepsilon_0, \delta_0$  are small enough we may assume  $\theta \leq t(\varepsilon, \delta) + r(\varepsilon, \delta) \leq 1/4$ . If  $\theta = 0$  the measurement  $T$  is perfectly complete, and we are done as we can take the measurements  $R^g$  in the conclusion of the proposition to be equal to  $T^g$ . So, for the rest of the proof, we can assume that  $\theta > 0$ . To complete the proof we need to prove a better upper bound on  $\theta$ . Towards this, introduce a state

$$|\Phi\rangle = \frac{|\tilde{\Phi}\rangle}{\| |\tilde{\Phi}\rangle \|}, \quad \text{where} \quad |\tilde{\Phi}\rangle = (\text{Id} - T) \otimes (\text{Id} - T) |\Psi\rangle.$$

Given the assumption that  $\theta > 0$ , it follows that  $\| |\tilde{\Phi}\rangle \| > 0$ , and hence this state is well defined. Moreover we can estimate the norm of  $|\tilde{\Phi}\rangle$  as follows:

$$\begin{aligned}
\| |\tilde{\Phi}\rangle \|^2 &= \langle (\text{Id} - T)^2, (\text{Id} - T)^2 \rangle_{\Psi} \\
&= \langle \text{Id} - 2T + T^2, \text{Id} - 2T + T^2 \rangle_{\Psi} \\
&= 1 - 4\langle T, \text{Id} \rangle_{\Psi} + 4\langle T, T \rangle_{\Psi} + 2\langle T^2, \text{Id} \rangle_{\Psi} - 4\langle T^2, T \rangle_{\Psi} + \langle T^2, T^2 \rangle_{\Psi} \\
&= 1 - 4\langle T, (\text{Id} - T) \rangle_{\Psi} + 2\langle T^2, (\text{Id} - T) \rangle_{\Psi} - \langle T^2, T(\text{Id} - T) \rangle_{\Psi} - \langle T^2, T \rangle_{\Psi} \\
&\approx_{\sqrt{\xi}} 1 - \langle T, T^2 \rangle_{\Psi} \\
&\approx_{\sqrt{\xi}} 1 - \langle T, \text{Id} \rangle_{\Psi}, \tag{10}
\end{aligned}$$



## 20:14 Retracted: Two-Player Entangled Games are NP-Hard

where the last two approximations use the projectivity assumption on  $T$ .

► **Claim 14.** *There are  $\varepsilon' = O(\varepsilon + \sqrt{\xi})$  and  $\delta' = O(\delta + \sqrt{\xi})$  such that  $(\{A_x^a\}, \mathcal{G})$  is  $(\varepsilon', \delta')$ -globally consistent on  $|\Phi\rangle$ .*

**Proof.** We verify the properties in Definition 9. Item 1. is automatic. For item 2., self-consistency of  $\{A_x^a\}$  on  $|\Phi\rangle$ , write

$$\begin{aligned} \mathbb{E}_x \sum_a \langle A_x^a, A_x^a \rangle_{\tilde{\Phi}} &= \mathbb{E}_x \sum_a \langle A_x^a(\text{Id} - T) - TA_x^a(\text{Id} - T), (\text{Id} - T)A_x^a - (\text{Id} - T)A_x^a T \rangle_{\Psi} \\ &\approx \sqrt{\xi} \mathbb{E}_x \sum_a \langle A_x^a, A_x^a \rangle_{\Psi} - \langle T, A_x^a \rangle_{\Psi} + \langle A_x^a(\text{Id} - T), T \rangle_{\Psi} \\ &\approx \sqrt{\xi} 1 - \varepsilon - \langle T, \text{Id} \rangle_{\Psi}. \end{aligned}$$

Together with (10), it follows that  $\{A_x^a\}$  is  $\varepsilon'$ -self-consistent on  $|\Phi\rangle$ , for some  $\varepsilon' = O(\varepsilon + \sqrt{\xi})$ . For item 3. in the definition, let  $Z$  be such that  $A^g - (A^g)^2 \leq Z$  for all  $g \in \mathcal{G}$ , and  $\langle Z, \text{Id} \rangle_{\Psi} \leq \delta$ . Then

$$\begin{aligned} \langle Z, \text{Id} \rangle_{\tilde{\Phi}} &\approx \sqrt{\xi} \langle Z, (\text{Id} - T) \rangle_{\Psi} \\ &\leq \delta, \end{aligned}$$

and the property follows using (10). ◀

Applying condition 2. in the proposition to  $|\Phi\rangle$  and  $(\{A_x^a\}, \mathcal{G})$  we obtain a measurement  $\{Q^g\}$  that is  $\xi' = t(\varepsilon', \delta')$ -projective and consistent with  $\{A_x^a\}$  on  $|\Phi\rangle$ . Define a sub-measurement  $\{R^g\}$  by

$$R^g := TT^gT + (1 - T)Q^g(1 - T).$$

The completeness of this measurement on  $|\Psi\rangle$  is

$$\begin{aligned} \langle R, \text{Id} \rangle_{\Psi} &= \langle T^3, \text{Id} \rangle_{\Psi} + \langle (1 - T)^2, \text{Id} \rangle_{\Psi} \\ &\approx \sqrt{\xi} 1, \end{aligned} \tag{11}$$

since

$$\langle T^3, \text{Id} \rangle_{\Psi} \approx \sqrt{\xi} \langle T^2, \text{Id} \rangle_{\Psi} \approx \sqrt{\xi} \langle T, \text{Id} \rangle_{\Psi}.$$

To evaluate consistency with  $\{A_x^a\}$ ,

$$\begin{aligned} &\mathbb{E}_x \sum_g \sum_{a \neq g(x)} \langle R^g, A_x^a \rangle_{\Psi} \\ &= \mathbb{E}_x \sum_g \sum_{a \neq g(x)} (\langle TT^gT, A_x^a \rangle_{\Psi} + \langle (1 - T)Q^g(1 - T), A_x^a \rangle_{\Psi}) \\ &\approx \sqrt{\varepsilon + \sqrt{\xi} + \kappa} \mathbb{E}_x \sum_g \sum_{a \neq g(x)} (\langle TT^g, TA_x^a \rangle_{\Psi} + \langle (1 - T)Q^g, (1 - T)A_x^a \rangle_{\Psi}) \\ &= O(\sqrt{\xi}) + O(\sqrt{\xi'}) \|\tilde{\Phi}\|^2, \end{aligned}$$

where the second line uses the second item in Lemma 6 and the last  $\varepsilon = O(\xi)$ , given the definition of the function  $r$ . Using (11), if we complete  $\{R^g\}$  into a measurement  $\{\tilde{R}^g\}$  by adding an arbitrary term, the latter will have consistency  $\delta'' = O(\sqrt{\xi}) + O(\sqrt{\xi'}) \|\tilde{\Phi}\|^2$  with

$\{A_x^a\}$ . Applying Lemma 13 yields a sub-measurement  $\{V^g\}$  that is  $\xi = r(\varepsilon, \delta)$ -projective and consistent with  $\{A_x^a\}$ , and for which

$$\langle (\text{Id} - V), \text{Id} \rangle_\Psi = O(\sqrt{\xi}) + O(\sqrt{\xi'}) \|\tilde{\Phi}\|^2.$$

Recall that by assumption,  $\{T^g\}$  is the most complete measurement that is  $\xi$ -projective and consistent with  $A_x$ . Hence,  $\langle (\text{Id} - V), \text{Id} \rangle_\Psi \geq \langle (\text{Id} - T), \text{Id} \rangle_\Psi$ , so that

$$\theta \leq O(\sqrt{\xi}) + O(\sqrt{\xi'}) (\theta + O(\sqrt{\xi})).$$

Provided  $\varepsilon, \delta$  are small enough that  $O(\sqrt{\xi'}) = O(\sqrt{t(\varepsilon', \delta')})$ , with  $\varepsilon', \delta'$  as in Claim 14, is at most  $1/4$ , as can be assumed from the assumed upper bound  $t(\varepsilon', \delta') \leq t_0$  for  $\varepsilon' \leq \varepsilon_0$  and  $\delta' \leq \delta_0$  provided  $t_0$  is a small enough universal constant, we have obtained  $\theta = O(\sqrt{\xi}) = O(\sqrt{r(\varepsilon, \delta)})$ , as claimed.  $\blacktriangleleft$

#### 4 NP-hardness for two-player entangled games

Based on the result of the analysis of the low-degree test stated in Theorem 2 and following the same sequence of reductions — composition of the low-degree test with itself, to reduce answer size, and combination with the 3-SAT test — as in [26] we obtain the following analogue of [26, Theorem 4.1], which establishes NP-hardness for games with  $\text{poly}(\log \log n)$ -bit answers.

**► Theorem 15.** *There is an  $\varepsilon > 0$  such that the following holds. Given a 2-player game  $G$  in explicit form, it is NP-hard to distinguish between  $\omega(G) = 1$  and  $\omega^*(G) \leq 1 - \varepsilon$ . Furthermore, the problem is still NP-hard when restricting to games  $G$  of size  $n$  that are projection games for which questions and answers can be specified using  $O(\log n)$  bits and  $\text{poly}(\log \log n)$  bits respectively.*

In [26] this result is improved to obtain hardness for games with constant-bit answers by reducing the 3-SAT test, on which the proof of Theorem 15 is based, to the three-player QUADEQ test for testing satisfiability of a system of quadratic equations in binary variables. This amounts to composing a PCP based on low-degree polynomials with the “exponential PCP” based on the three-query linearity test of [5], and yields hardness for three-player games with binary answers. The same steps can be completed with two players only by using the technique of oracularization to transform the QUADEQ and linearity tests into two-player games. The idea of oracularization is that for every triple of questions  $(q_1, q_2, q_3)$  to be sent to the three players in the original test, the verifier sends the entire triple to a single player, Alice, and receives a triple of answers. The verifier also sends a randomly selected question from the triple to a second player, Bob. The verifier accepts if and only if Bob’s answer is consistent with Alice’s, and the triple of answers provided by Alice would have been accepted in the original test. For concreteness, we summarize the oracularized QUADEQ test in Figure 2. (Note that the third element in each of Alice’s question and answer triples is redundant and can be eliminated.)

It is easy to see that honest strategies pass the oracularized QUADEQ test with probability 1. To establish soundness of the test, i.e. to show an analogue of Lemma 3.5 of [26], we can follow essentially the same steps as in the proof of that lemma. The key step of the proof is to argue that, due to the soundness of the linearity test against entangled provers, there exist measurements on each prover’s space whose outcomes are linear functions that are consistent with the measurements applied in the test. For the oracularized test, we can perform this step using the soundness of the oracularized linearity test against entangled provers, which was analyzed in [20]. The rest of the proof proceeds unchanged. As a result we obtain the

---

Out of the two provers, choose one at random to be Alice and the other to be Bob.

1. With probability  $1/4$  each, do the following:
    - a. Send label  $\ell_1$  to the two players and perform the  $(n/2)$ -bit (oracularized) linearity test.
    - b. Same with label  $\ell_2$ .
    - c. Send labels  $(\ell_1, \ell_2)$  to the two players and perform the  $n$ -bit linearity test.
    - d. Same but perform the  $n^2$ -bit linearity test.
  2. Select random  $u, v \in \mathbb{F}_2^{n/2}$  and  $i \in [3]$ , and generate the three queries  $q_1 = (\ell_1, u)$ ,  $q_2 = (\ell_2, v)$ ,  $q_3 = (\ell_1, \ell_2, (u, v))$ . Send  $q_1, q_2$  to Alice, receiving answers  $a_1, a_2$ , and let  $a_3 = a_1 + a_2$ . Send  $q_i$  to Bob, receiving answer  $b$ . Accept if  $b = a_i$ .
  3. Select random  $u, v \in \mathbb{F}_2^n$  and  $i \in [3]$ , and generate the three queries  $q_1 = (\ell_1, \ell_2, u)$ ,  $q_2 = (\ell_1, \ell_2, v)$ ,  $q_3 = (\ell_1, \ell_2, u \otimes v)$ . Send  $q_1, q_2$  to Alice, receiving answers  $a_1, a_2$  and let  $a_3 = a_1 \cdot a_2$ . Send  $q_i$  to Bob, receiving answer  $b$ . Accept if  $b = a_i$ .
  4. Select a random vector  $v \in \mathbb{F}_2^K$  and let  $w = \sum_k w_k a^{(k)} \in \mathbb{F}_2^{n^2}$ . Send  $(\ell_1, \ell_2, w)$  to a randomly chosen player and check that the answer  $a = \sum_k w_k c^{(k)}$ .
- 

■ **Figure 2** The two-prover QUADEQ test. See Section [26, Section 3.4] for additional explanations regarding the notation.

following corollary, which establishes Theorem 1; it is completely analogous to [26, Corollary 4.3], except that due to the oracularization, the two provers now have to provide answers of two bits each instead of one.

► **Corollary 16.** *There is an  $\varepsilon > 0$  such that the following holds. Given a two-player projection game  $G$  in explicit form in which answers from one player is restricted to 2 bits, and answers from the other player to a single bit, it is NP-hard to distinguish between  $\omega(G) = 1$  and  $\omega^*(G) \leq 1 - \varepsilon$ .*

Using that the games  $G$  for which NP-hardness is shown in Corollary 16 are projection games, we may apply results on the parallel repetition of two-player entangled projection games [7] to amplify the completeness and soundness parameters from 1 and  $1 - \varepsilon$  to 1 and  $\delta$  respectively, for any  $\delta > 0$ , by repeating the game  $\text{poly}(\varepsilon^{-1} \log \delta^{-1})$  times and incurring a corresponding multiplicative factor blow-up in the length of questions and answers in the game.

---

## References

- 1 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.
- 2 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.
- 3 László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.
- 4 John S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1:195–200, 1964.
- 5 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47:549–595, 1993.
- 6 Richard Cleve, Peter Høyer, Ben Toner, and John Watrous. Consequences and limits of nonlocal strategies. 2004. [arXiv:quant-ph/0404076](https://arxiv.org/abs/quant-ph/0404076).
- 7 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.

- 8 Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935.
- 9 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.
- 10 Joseph Fitzsimons and Thomas Vidick. A multiprover interactive proof system for the local Hamiltonian problem. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 103–112. ACM, 2015.
- 11 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *Journal of the ACM (JACM)*, 62(4):27, 2015.
- 12 Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.
- 13 Tsuyoshi Ito, Hirotada Kobayashi, and Keiji Matsumoto. Oracularization and two-prover one-round interactive proofs against nonlocal strategies. In *Proceedings: Twenty-Fourth Annual IEEE Conference on Computational Complexity (CCC 2009)*, pages 217–228, July 2009.
- 14 Tsuyoshi Ito and Thomas Vidick. A multi-prover interactive proof for NEXP sound against entangled provers. *Proc. 53rd FOCS*, pages 243–252, 2012. [arXiv:1207.0550](#).
- 15 Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP = PSPACE. *Communications of the ACM*, 53(12):102–109, 2010. [arXiv:0907.4737](#).
- 16 Zhengfeng Ji. Classical verification of quantum proofs. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 885–898. ACM, 2016.
- 17 Zhengfeng Ji. Compression of quantum multi-prover interactive proofs. 2016. [arXiv:1610.03133](#).
- 18 Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 485–494. ACM, 2014. doi:10.1145/2591796.2591809.
- 19 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.
- 20 Anand Natarajan and Thomas Vidick. A quantum linearity test for robustly verifying entanglement. 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 1003–1015. ACM, 2017. doi:10.1145/3055399.3055468.
- 21 Anand Natarajan and Thomas Vidick. Low-degree testing for quantum states, and a quantum entangled games PCP. 2018. [arXiv:1801.03821v2](#).
- 22 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 475–484. ACM, 1997. doi:10.1145/258533.258641.
- 23 Ben Reichardt, Falk Unger, and Umesh Vazirani. A classical leash for a quantum system: Command of quantum systems via rigidity of CHSH games. *Nature*, 496(7446):456–460, 2013.
- 24 Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.
- 25 Ben Toner. Monogamy of non-local quantum correlations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 465(2101):59–69, 2009. doi:10.1098/rspa.2008.0149.
- 26 Thomas Vidick. Three-player entangled XOR games are NP-hard to approximate. In *Proc. 54th FOCS*, 2013. [arXiv:1302.1242](#).

## A

 Modified proofs from [26]

As noted in the introduction, the principal modifications to the soundness analysis of the low-degree test in [26] necessary to make it hold for two provers concern the self-improvement results of section 5. There are a few other steps of the proof of the main theorem in [26] that seem to require a tripartite tensor product factorization of the Hilbert space to be carried out. In all cases this is easily avoided by simple modification of the proof. Although they remain very elementary, in this appendix we describe the only two other non-trivial modifications needed. The first is in the proof of [26, Claim 6.10]. (We refer to the paper [26] for context, including an explanation of the notation; the following discussion is meant for a reader already familiar with the proofs in [26].)

► **Claim 17** (Claim 6.10 in [26]). *The measurements  $\{Q_s^g\}_{g \in \mathcal{P}_d(s)}$  satisfy*

$$\mathbb{E}_{s \in \mathcal{S}_{m-1}(\mathbb{F}_q^m)} \sum_{g \in \mathcal{P}_d(s)} \langle Q_s^g, (\text{Id} - Q_s^g) \rangle_\Psi = O(\varepsilon^{c_\ell}).$$

**Proof.** The proof is the same as in [26], except the third tensor factor is not needed — the second can be used for the same purpose:

$$\begin{aligned} \mathbb{E}_{s \in \mathcal{S}_{m-1}(\mathbb{F}_q^m)} \sum_{g, g' \in \mathcal{P}_d(s), g \neq g'} \langle Q_s^g, Q_s^{g'} \rangle_\Psi &\approx \mathbb{E}_{s \in \mathcal{S}_{m-1}(\mathbb{F}_q^m)} \mathbb{E}_{x \in S} \sum_{g, g' \in \mathcal{P}_d(s), g \neq g'} \langle Q_s^g, Q_s^{g'} A_x^{g(x)} \rangle_\Psi \\ &\quad + O(\varepsilon^{c_\ell}) \\ &\approx_{\varepsilon^{c_\ell}} \mathbb{E}_{s \in \mathcal{S}_{m-1}(\mathbb{F}_q^m)} \mathbb{E}_{x \in S} \sum_{g, g' \in \mathcal{P}_d(s), g \neq g'} \langle Q_s^g A_x^{g(x)}, Q_s^{g'} \rangle_\Psi \\ &\quad + O(\varepsilon^{c_\ell}) + O(\varepsilon) \\ &\approx O(\varepsilon^{c_\ell}) + O(\varepsilon). \end{aligned}$$

In the first line, we used the consistency between  $Q_s^g$  on the first prover and  $A_x^{g(x)}$  on the second; in the second line, we used the self-consistency of  $A$ ; and in the third, we used the consistency between  $Q_s^{g'}$  on the second prover and  $A_x^{g(x)}$  on the first prover. ◀

The second is in the proof of [26, Claim 6.14]. Here again, the use of a third tensor factor can be avoided by a simple modification. Specifically, the last set of centered equations on p.1056 (right below (6.22)) should be replaced with

$$\begin{aligned} \mathbb{E}_{(s_i)} \sum_{g, \deg(g) > d} \langle R_{(s_i)}^g, \text{Id} \rangle_\Psi &\approx_{\varepsilon^{c_\ell}} \mathbb{E}_{(s_i), z, \ell, \ell' \ni z} \sum_{g, \deg(g) > d} \sum_{\substack{h(\ell \cap s_i) = g(\ell \cap s_i) \\ h'(\ell' \cap s_i) = g(\ell' \cap s_i)}} \langle R_{(s_i)}^g, B_\ell^h B_{\ell'}^{h'} \rangle_\Psi \\ &\approx_{\varepsilon^{c_\ell}} \mathbb{E}_{(s_i), z, \ell, \ell' \ni z} \sum_{g, \deg(g) > d} \sum_{\substack{h(\ell \cap s_i) = g(\ell \cap s_i) \\ h'(\ell' \cap s_i) = g(\ell' \cap s_i)}} \langle R_{(s_i)}^g B_{\ell'}^{h'}, B_\ell^h \rangle_\Psi \\ &= O(\varepsilon^{d_c/2}) \end{aligned}$$

## Retraction Notice

The article, published on June 4th, 2018 in the CCC 2018 proceedings (LIPIcs, volume 102, <https://www.dagstuhl.de/dagpub/978-3-95977-069-9>), has been retracted by agreement between the authors, the editor(s), and the publisher Schloss Dagstuhl / LIPIcs. The retraction has been agreed due to an error in the proof of the main result, arising from an error in the cited paper “Three-player entangled XOR games are NP-hard to approximate” by Thomas Vidick (SICOMP '16). The error in that paper is in the proof of soundness of the plane vs point low-degree test against 3 entangled provers. The main technical result of the present article is a modification of that proof to hold against 2 entangled provers, and is affected by the same error. For more details on the nature of the error, and a description of which results in the literature are invalidated and which have been recovered by other techniques, see the erratum by Thomas Vidick for the SICOMP '16 paper available at <http://users.cms.caltech.edu/~vidick/errata.pdf>. (At a high level, the results that survive are “scaled up” complexity results about the complexity of MIP\*, whereas results in the “scaled down” setting such as the NP-hardness result of this article cannot be directly recovered with current techniques.)


*Dagstuhl Publishing – January 5, 2021.*

RETRACTED

# Complexity Classification of Conjugated Clifford Circuits

**Adam Bouland**<sup>1</sup>

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA  
adam@csail.mit.edu


 <https://orcid.org/0000-0002-8556-8337>

**Joseph F. Fitzsimons**<sup>2</sup>

Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372  
Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543  
joe.fitzsimons@nus.edu.sg

**Dax Enshan Koh**<sup>3</sup>

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA  
daxkoh@mit.edu

 <https://orcid.org/0000-0002-8968-591X>

---

## Abstract

Clifford circuits – i.e. circuits composed of only CNOT, Hadamard, and  $\pi/4$  phase gates – play a central role in the study of quantum computation. However, their computational power is limited: a well-known result of Gottesman and Knill states that Clifford circuits are efficiently classically simulable. We show that in contrast, “conjugated Clifford circuits” (CCCs) – where one additionally conjugates every qubit by the same one-qubit gate  $U$  – can perform hard sampling tasks. In particular, we fully classify the computational power of CCCs by showing that essentially any non-Clifford conjugating unitary  $U$  can give rise to sampling tasks which cannot be efficiently classically simulated to constant multiplicative error, unless the polynomial hierarchy collapses. Furthermore, by standard techniques, this hardness result can be extended to allow for the more realistic model of constant additive error, under a plausible complexity-theoretic conjecture. This work can be seen as progress towards classifying the computational power of all restricted quantum gate sets.

**2012 ACM Subject Classification** Theory of computation → Quantum complexity theory, Theory of computation → Computational complexity and cryptography

**Keywords and phrases** gate set classification, quantum advantage, sampling problems, polynomial hierarchy

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.21

**Related Version** A full version of the paper is available at <https://arxiv.org/abs/1709.01805>.

---

<sup>1</sup> AB was partially supported by the NSF GRFP under Grant No. 1122374, by a Vannevar Bush Fellowship from the US Department of Defense, and by an NSF Waterman award under grant number 1249349.

<sup>2</sup> JFF acknowledges support from the Air Force Office of Scientific Research under AOARD grant no. FA2386-15-1-4082. This material is based on research supported in part by the Singapore National Research Foundation under NRF Award No. NRF-NRFF2013-01.

<sup>3</sup> DEK is supported by the National Science Scholarship from the Agency for Science, Technology and Research (A\*STAR).



© Adam Bouland, Joseph F. Fitzsimons, and Dax E. Koh;  
licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 21; pp. 21:1–21:25

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**Acknowledgements** We thank Scott Aaronson, Mick Bremner, Alex Dalzell, Daniel Gottesman, Aram Harrow, Ashley Montanaro, Ted Yoder and Mithuna Yoganathan for helpful discussions.

## 1 Introduction

Quantum computers hold the promise of efficiently solving certain problems, such as factoring integers [53], which are believed to be intractable for classical computers. However, experimentally implementing many of these quantum algorithms is very difficult. For instance, the largest number factored to date using Shor’s algorithm is 21 [40]. These considerations have led to an intense interest in algorithms which can be more easily implemented with near-term quantum devices, as well as a corresponding interest in the difficulty of these computational tasks for classical computers. Some prominent examples of such models are constant-depth quantum circuits [56, 11] and non-adaptive linear optics [3].

In many of these constructions, one can show that these “weak” quantum devices can perform *sampling* tasks which cannot be efficiently simulated classically, even though they may not be known to be capable of performing difficult *decision* tasks. Such arguments were first put forth by Bremner, Jozsa, and Shepherd [14] and Aaronson and Arkhipov [3], who showed that exactly simulating the sampling tasks performed by weak devices is impossible assuming the polynomial hierarchy is infinite. The proofs of these results use the fact that the output probabilities of quantum circuits can be very difficult to compute – in fact they can be  $\text{GapP}$ -hard and therefore  $\#\text{P}$ -hard to approximate. In contrast the output probabilities of classical samplers can be approximated in  $\text{BPP}^{\text{NP}}$  by Stockmeyer’s approximate counting theorem [55], and therefore lie in the polynomial hierarchy. Hence a classical simulation of these circuits would collapse the polynomial hierarchy to the third level by Toda’s Theorem [57]. Similar hardness results have been shown for many other models of quantum computation [56, 43, 22, 10, 21, 8, 16, 39].

A curious feature of many of these “weak” models of quantum computation is that they can be implemented using non-universal gate sets. That is, despite being able to perform sampling problems which appear to be outside of  $\text{BPP}$ , these models are not themselves known to be capable of universal quantum computation. In short these models of quantum computation seem to be “quantum-intermediate” between  $\text{BPP}$  and  $\text{BQP}$ , analogous to the  $\text{NP}$ -intermediate problems which are guaranteed to exist by Ladner’s theorem [36]. From the standpoint of computational complexity, it is therefore natural to study this intermediate space between  $\text{BPP}$  and  $\text{BQP}$ , and to classify its properties.

One natural way to explore the space between  $\text{BPP}$  and  $\text{BQP}$  is to classify the power of all possible quantum gate sets over qubits. The Solovay-Kitaev Theorem states that all universal quantum gate sets, i.e. those which densely generate the full unitary group, have equivalent computational power [18]. Therefore the interesting gates sets to classify are those which are non-universal. However just because a gate set is non-universal does not imply it is weaker than  $\text{BQP}$  – in fact some non-universal gates are known to be capable of universality in an “encoded” sense, and therefore have the same computational power as  $\text{BQP}$  [32]. Other non-universal gate sets are efficiently classically simulable [25], while others seem to lie “between  $\text{BPP}$  and  $\text{BQP}$ ” in that they are believed to be neither universal for  $\text{BQP}$  nor efficiently classically simulable [14]. It is a natural open problem to fully classify all restricted gate sets into these categories according to their computational complexity.

This is a challenging problem, and to date there has only been partial progress towards this classification. One immediate difficulty in approaching this problem is that there is not a known classification of all possible non-universal gate sets. In particular this would require



classifying the discrete subgroups of  $SU(2^n)$  for all  $n \in \mathbb{N}$ , which to date has only been solved for  $n \leq 2$  [28]. Therefore existing results have characterized the power of modifications of known intermediate gate sets, such as commuting circuits and linear optical elements [9, 10, 47]. Others works have classified the classical subsets of gates [6, 27], or else given sufficient criteria for universality so as to rule out the existence of certain intermediate families [52]. A complete classification of this space “between BPP and BQP” would require a major improvement in our understanding of universality as well as the types of computational hardness possible between BPP and BQP.

One well-known example of a non-universal family of quantum gates is the Clifford group. Clifford circuits – i.e. circuits composed of merely CNOT, Hadamard and Phase gates – are a discrete subgroup of quantum gates which play an important role in quantum error correction [24, 13], measurement-based quantum computing [49, 50, 17], and randomized benchmarking [38]. However a well-known result of Gottesman and Knill states that circuits composed of Clifford elements are efficiently classically simulable [25, 5]. That is, suppose one begins in the state  $|0\rangle^{\otimes n}$ , applies polynomially many gates from the set CNOT, H, S, then measures in the computational basis. Then the Gottesman-Knill theorem states that one can compute the probability a string  $y$  is output by such a circuit in classical polynomial time. One can also sample from the same probability distribution on strings as this circuit as well. A key part of the proof of this result is that the quantum state at intermediate stages of the circuit is always a “stabilizer state” – i.e. the state is uniquely described by its set of stabilizers in the Pauli group – and therefore has a compact representation. Therefore the Clifford group is incapable of universal quantum computation (assuming  $\text{BPP} \neq \text{BQP}$ ).

In this work, we will study the power of a related family of non-universal gates, known as *Conjugated Clifford gates*, which we introduce below. These gates are non-universal by construction, but not known to be efficiently classically simulable either. Our main result will be to *fully classify* the computational power of this family of intermediate gate sets.

## 1.1 Our results

This paper considers a new “weak” model of quantum computation which we call “conjugated Clifford circuits” (CCCs). In this model, we consider the power of quantum circuits which begin in the state  $|0\rangle^{\otimes n}$ , and then apply gates from the set  $(U^\dagger \otimes U^\dagger)(\text{CNOT})(U \otimes U), U^\dagger H U, U^\dagger S U$  where  $U$  is a fixed one-qubit gate. In other words, we consider the power of Clifford circuits which are conjugated by an identical one-qubit gate  $U$  on each qubit. These gates manifestly perform a discrete subset of unitaries so this gate set is clearly not universal.

Although this transformation preserves the non-universality of the Clifford group, it is unclear if it preserves its computational power. The presence of generic conjugating unitaries (even the same  $U$  on each qubit, as in this model) breaks the Gottesman-Knill simulation algorithm [25], as the inputs and outputs of the circuit are not stabilizer states/measurements. Hence the intermediate states of the circuit are no longer efficiently representable by the stabilizer formalism. This, combined with prior results showing hardness for other modified versions of Clifford circuits [33, 34], leads one to suspect that CCCs may not be efficiently classically simulable. However prior to this work no hardness results were known for this model.

In this work, we confirm this intuition and provide two results in this direction. First, we provide a *complete classification* of the power of CCCs according to the choice of  $U$ . We do this by showing that *any*  $U$  which is not efficiently classically simulable by the

Gottesman-Knill theorem suffices to perform hard sampling problems with CCCs<sup>4</sup>. That is, for generic  $U$ , CCCs cannot be efficiently classically simulated to constant multiplicative error by a classical computer unless the polynomial hierarchy collapses. This result can be seen as progress towards classifying the computational complexity of restricted gate sets. Indeed, given a non-universal gate set  $G$ , a natural question is to classify the power of  $G$  when conjugated by the same one-qubit unitary  $U$  on each qubit, as this transformation preserves non-universality. Our work resolves this question for one of the most prominent examples of non-universal gate sets, namely the Clifford group. As few examples of non-universal gate sets are known<sup>5</sup>, this closes one of the major gaps in our understanding of intermediate gate sets. Of course this does not complete the complexity classification of all gate sets, as there is no known classification of all possible non-universal gate sets. However it does make progress towards this goal.

Second, we show that under an additional complexity-theoretic conjecture, classical computers cannot efficiently simulate CCCs to constant error in total variation distance. This is a more experimentally achievable model of error for noisy error-corrected quantum computations. The proof of this result uses standard techniques introduced by Aaronson and Arkhipov [3], which have also been used in other models [22, 15, 8, 16, 42, 39].

This second result is interesting for two reasons. First, it means our results may have relevance to the empirical demonstration of quantum advantage (sometimes referred to as “quantum supremacy”) [48, 8, 4], as our results are robust to noise. Second, from the perspective of computational complexity, it gives yet another conjecture upon which one can base the supremacy of noisy quantum devices. As is the case with other quantum supremacy proposals [3, 22, 15, 42, 39], in order to show that simulation of CCCs to additive error still collapses the polynomial hierarchy, we need an additional conjecture stating that the output probabilities of these circuits are hard to approximate on average. Our conjecture essentially states that for most Clifford circuits  $V$  and most one-qubit unitaries  $U$ , it is  $\#\text{P}$ -hard to approximate a constant fraction of the output probabilities of the CCC  $U^{\otimes n}V(U^\dagger)^{\otimes n}$  to constant multiplicative error. We prove that this conjecture is true in the worst case – in fact, for all non-Clifford  $U$ , there exists a  $V$  such that some outputs are  $\#\text{P}$ -hard to compute to multiplicative error. However, it remains open to extend this hardness result to the average case, as is the case with other supremacy proposals as well [3, 22, 15, 42, 39]. To the best of our knowledge our conjecture is independent of the conjectures used to establish other quantum advantage results such as boson sampling [3], Fourier sampling [22] or IQP [15, 16]. Therefore our results can be seen as establishing an alternative basis for belief in the advantage of noisy quantum devices over classical computation.

One final motivation for this work is that CCCs might admit a simpler fault-tolerant implementation than universal quantum computing, which we conjecture to be the case. It is well-known that many stabilizer error-correcting codes, such as the 5-qubit and 7-qubit codes [37, 19, 54], admit transversal Clifford operations [24]. That is, performing fault-tolerant Clifford operations on the encoded logical qubits can be done in a very simple manner – by simply performing the corresponding Clifford operation on the physical qubits. This is manifestly fault-tolerant, in that an error on one physical qubit does not “spread” to more than 1 qubit when applying the gate. In contrast, performing non-Clifford operations fault-tolerantly on such codes requires substantially larger (and non-transversal) circuits – and

---

<sup>4</sup> More precisely, we show that any  $U$  that cannot be written as a Clifford times a  $Z$ -rotation suffices to perform hard sampling problems with CCCs. See Theorem 7 for the exact statement.

<sup>5</sup> The only examples to our knowledge are matchgates, Clifford gates, diagonal gates, and subsets thereof.

therefore the non-transversal operations are often the most resource intensive. The challenge in fault-tolerantly implementing CCCs therefore lies in performing the initial state preparation and measurement. Initial preparation of non-stabilizer states in these codes is equivalent to the challenge of producing magic states, which are already known to boost Clifford circuits to universality using adaptive Clifford circuits [13, 12] (in contrast our construction would only need non-adaptive Clifford circuits with magic states). Likewise, measuring in a non-Clifford basis would require performing non-Clifford one-qubit gates prior to fault-tolerant measurement in the computational basis. Therefore the state preparation/measurement would be the challenging part of fault-tolerantly implementing CCCs in codes with transversal Cliffords. It remains open if there exists a code with transversal conjugated Cliffords<sup>6</sup> and easy preparation and measurement in the required basis. Such a code would not be ruled out by the Eastin-Knill Theorem [20], which states that the set of transversal gates must be discrete for all codes which correct arbitrary one qubit errors. Of course this is not the main motivation for exploring the power of this model – which is primarily to classify the space between BPP and BQP – but an easier fault-tolerant implementation could be an unexpected bonus of our results.

## 1.2 Proof Techniques

To prove these results, we use several different techniques.

### 1.2.1 Proof Techniques: classification of exact sampling hardness

To prove exact (or multiplicative) sampling hardness for CCCs for essentially all non-Clifford  $U$ , we use the notion of postselection introduced by Aaronson [2]. Postselection is the (non-physical) ability to discard all runs of the computation which do not achieve some particular outcomes. Our proof works by showing that postselecting such circuits allows them to perform universal quantum computation. Hardness then follows from known techniques [2, 14, 3].

One technical subtlety that we face in this proof, which is not present in other results, is that our postselected gadgets perform operations which are not closed under inversion. This means one cannot use the Solovay-Kitaev theorem to change quantum gate sets [18]. This is a necessary step in the proof that  $\text{PostBQP} = \text{PP}$  [2], which is a key part of the hardness proof (see [10]). Fortunately, it turns out that we can get away without inverses due to a recent inverse-free Solovay-Kitaev theorem of Sardharwalla *et al.* [51], which removes the needs for inverses if the gate set contains the Paulis. Our result would have been much more difficult to obtain without this prior result. To our knowledge this is the first application of their result to structural complexity.

A further difficulty in the classification proof is that the postselection gadgets we derive do not work for all non-Clifford  $U$ . In general, most postselection gadgets give rise to non-unitary operations, and for technical reasons we need to work with unitary postselection gadgets to apply the results of [51]. Therefore, we instead use several different gadgets which cover different portions of the parameter space of  $U$ 's. Our initial proof of this fact used a total

---

<sup>6</sup> Of course one can always “rotate” a code with transversal Clifford operations to obtain a code with transversal conjugated Cliffords. If the code previously had logical states  $|0\rangle_L, |1\rangle_L$ , then by setting the states  $|0\rangle'_L = U_L^\dagger |0\rangle_L$  and  $|1\rangle'_L = U_L^\dagger |1\rangle_L$ , one obtains a code in which the conjugated Clifford gates (conjugated by  $U$ ) are transversal. However having the ability to efficiently fault-tolerantly prepare  $|0\rangle_L$  in the old code does not imply the same ability to prepare  $|0\rangle'_L$  in the new code.

of seven postselection gadgets found by hand. We later simplified this to two postselection gadgets by conducting a brute-force search for suitable gadgets using Christopher Granade and Ben Criger’s QuaEC package [26]. We include this simplified proof in this writeup.

A final difficulty that one often faces with postselected universality proofs is that one must show that the postselection gadgets boost the original gate set to universality. In general this is a nontrivial task; there is no simple test of whether a gate set is universal, though some sufficient (but not necessary) criteria are known [52]. Prior gate set classification theorems have solved this universality problem using representation theory [9, 52] or Lie theory [10, 47]. However, in our work we are able to make use of a powerful fact: namely that the Clifford group plus *any* non-Clifford unitary is universal. This follows from results of Nebe, Rains and Sloane [44, 45, 1] classifying the invariants of the Clifford group<sup>7</sup>. As a result our postselected universality proofs are much simpler than in other gate set classification theorems.

### 1.2.2 Proof techniques: additive error

To prove hardness of simulation to additive error, we follow the techniques of [3, 15, 22, 42]. In these works, to show hardness of sampling from some probability distribution with additive error, one combines three different ingredients. The first is anti-concentration – showing that for these circuits, the output probabilities in some large set  $T$  are somewhat large. Second, one uses Markov’s inequality to argue that, since the simulation error sums to  $\epsilon$ , on some other large set of output probabilities  $S$ , the error must be below a constant multiple of the average. If  $S$  and  $T$  are both large, they must have some intersection – and on this intersection  $S \cap T$ , the imagined classical simulation is not only a simulation to additive error, but also to multiplicative error as well (since the output probability in question is above some minimum). Therefore a simulation to some amount  $\epsilon$  of additive error implies a multiplicative simulation to the output probabilities on a constant fraction of the outputs. The impossibility of such a simulation is then obtained by assuming that computing these output probabilities is multiplicatively hard on average. In particular, one assumes that it is a  $\#\text{P}$ -hard task to compute the output probability on  $|S \cap T|/2^n$ -fraction of the outputs. This leads to a collapse of the polynomial hierarchy by known techniques [3, 14].

We follow this technique to show hardness of sampling with additive error. In our case, the anticoncentration theorem follows from the fact that the Clifford group is a “2-design” [58, 59] – i.e. a random Clifford circuit behaves equivalently to a random unitary up to its second moment – and therefore must anticoncentrate, as a random unitary does (the fact that unitary designs anticoncentrate was also shown independently by several groups [29, 39, 31]). This is similar to the hardness results for IQP [15] and DQC1 [42], in which the authors also prove their corresponding anticoncentration theorems. In contrast it is open to prove the anticoncentration theorem used for Boson Sampling and Fourier Sampling [3, 22], though these models have other complexity-theoretic advantages<sup>8</sup>. Therefore the only assumption needed is the hardness-on-average assumption. We also show that our hardness assumption is true for worst-case inputs. This result follows from combining known facts about BQP with the classification theorem for exact sampling hardness.

<sup>7</sup> However we note that in our proofs we will only use the fact that the Clifford group plus any non-Clifford element is universal on a qubit. This version of the theorem admits a direct proof using the representation theory of  $SU(2)$ .

<sup>8</sup> For instance, for these models it is known to be  $\#\text{P}$ -hard to *exactly* compute most output probabilities of their corresponding circuit. This is a necessary but not sufficient condition for the supremacy conjectures to be true, which require it to be  $\#\text{P}$ -hard to *approximately* compute most output probabilities of their corresponding circuit. It remains open to show an exact average-to-worst case reduction for models which exhibit anticoncentration, such as our model, IQP, and DQC1.

### 1.3 Relation to other works on modified Clifford circuits

While we previously discussed the relation of our results to prior work on gate set classification and sampling problems, here we compare our results to prior work on Clifford circuits. We are not the first to consider the power of modified Clifford circuits. Jozsa and van den Nest [33] and Koh [34], categorized the computational power of a number of modified versions of Clifford circuits. The closest related result is the statement in [33] that if the input state to a Clifford circuit is allowed to be an arbitrary tensor product of one-qubit states, then such circuits cannot be efficiently classically simulated unless the polynomial hierarchy collapses. Their hardness result uses states of the form  $|0\rangle^{\otimes n/2}|\alpha\rangle^{\otimes n/2}$ , where  $|\alpha\rangle = \cos(\pi/8)|0\rangle + i\sin(\pi/8)|1\rangle$  is a magic state. They achieve postselected hardness via the use of magic states to perform T gates, using a well-known construction (see e.g. [13]). So in the [33] construction there are different input states on different qubits. In contrast, our result requires the same input state on every qubit – as well as measurement in that basis at the end of the circuit. This ensures our modified circuit can be interpreted as the action of a discrete gate set, and therefore our result has relevance for the classification of the power of non-universal gate sets.

## 2 Preliminaries

We denote the single-qubit *Pauli matrices* by  $X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ ,  $Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ ,  $Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ , and  $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ . The  $\pm 1$ -eigenstates of  $Z$  are denoted by  $|0\rangle$  and  $|1\rangle$  respectively. The *rotation operator* about an axis  $t \in \{x, y, z\}$  with an angle  $\theta \in [0, 2\pi)$  is

$$R_t(\theta) = e^{-i\theta\sigma_t/2} = \cos(\theta/2)I - i\sin(\theta/2)\sigma_t. \quad (1)$$

We will use the fact that any single-qubit unitary operator  $U$  can be written as

$$U = e^{i\alpha}R_z(\phi)R_x(\theta)R_z(\lambda), \quad (2)$$

where  $\alpha, \phi, \theta, \lambda \in [0, 2\pi)$  [46].

For linear operators  $A$  and  $B$ , we write  $A \propto B$  to mean that there exists  $\alpha \in \mathbb{C} \setminus \{0\}$  such that  $A = \alpha B$ . For linear operators, vectors or complex numbers  $a$  and  $b$ , we write  $a \sim b$  to mean that  $a$  and  $b$  differ only by a global phase, i.e. there exists  $\theta \in [0, 2\pi)$  such that  $a = e^{i\theta}b$ . For any subset  $S \subseteq \mathbb{R}$  and  $k \in \mathbb{R}$ , we write  $kS$  to refer to the set  $\{kn : n \in S\}$ . For example,  $k\mathbb{Z} = \{kn : n \in \mathbb{Z}\}$ . We denote the set of odd integers by  $\mathbb{Z}_{\text{odd}}$ . We denote the complement of a set  $S$  by  $S^c$ .

### 2.1 Clifford circuits and conjugated Clifford circuits

The  $n$ -qubit *Pauli group*  $\mathcal{P}_n$  is the set of all operators of the form  $i^k P_1 \otimes \dots \otimes P_n$ , where  $k \in \{0, 1, 2, 3\}$  and each  $P_j$  is a Pauli matrix. The  $n$ -qubit *Clifford group* is the normalizer of  $\mathcal{P}_n$  in the  $n$ -qubit unitary group  $\mathcal{U}_n$ , i.e.  $\mathcal{C}_n = \{U \in \mathcal{U}_n : U\mathcal{P}_nU^\dagger = \mathcal{P}_n\}$ .

The elements of the Clifford group, called *Clifford operations*, have an alternative characterization: an operation is a Clifford operation if and only if it can be written as a circuit comprising the following gates, called *basic Clifford gates*: *Hadamard*,  $\pi/4$  *phase*, and

## 21:8 Complexity Classification of Conjugated Clifford Circuits

*controlled-NOT* gates, whose matrix representations in the computational basis are

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad \text{and} \quad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

respectively. An example of a non-Clifford gate is the  $T$  gate, whose matrix representation is given by  $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ . We denote the group generated by the single-qubit Clifford gates by  $\langle S, H \rangle$ .

We will make use of the following fact about Clifford operations.

► **Fact 1.**  $R_z(\phi)$  is a Clifford operation if and only if  $\phi \in \frac{\pi}{2}\mathbb{Z}$ .

A *Clifford circuit* is a circuit that consists of computational basis states being acted on by the basic Clifford gates, before being measured in the computational basis. Without loss of generality, we may assume that the input to the Clifford circuit is the all-zero state  $|0\rangle^{\otimes n}$ . We define conjugated Clifford circuits (CCCs) similarly to Clifford circuits, except that each basic Clifford gate  $G$  is replaced by a conjugated basic Clifford gate  $(U^{\otimes k})^\dagger g U^{\otimes k}$ , where  $k = 1$  when  $g = H, S$  and  $k = 2$  when  $g = \text{CNOT}$ . In other words,

► **Definition 2.** Let  $U$  be a single-qubit unitary gate. A  $U$ -conjugated Clifford circuit ( $U$ -CCC) on  $n$  qubits is defined to be a quantum circuit with the following structure:

1. Start with  $|0\rangle^{\otimes n}$ .
2. Apply gates from the set  $\{U^\dagger H U, U^\dagger S U, (U^\dagger \otimes U^\dagger) \text{CNOT} (U \otimes U)\}$ .
3. Measure each qubit in the computational basis.

Because the intermediate  $U$  and  $U^\dagger$  gates cancel, we may equivalently describe a  $U$ -CCC as follows:

1. Start with  $|0\rangle^{\otimes n}$ .
2. Apply  $U^{\otimes n}$ .
3. Apply gates from the set  $\{H, S, \text{CNOT}\}$ .
4. Apply  $(U^\dagger)^{\otimes n}$ .
5. Measure each qubit in the computational basis.

### 2.2 Notions of classical simulation of quantum computation

Let  $\mathcal{P} = \{p_z\}_z$  and  $\mathcal{Q} = \{q_z\}_z$  be (discrete) probability distributions, and let  $\epsilon \geq 0$ . We say that  $\mathcal{Q}$  is a *multiplicative  $\epsilon$ -approximation* of  $\mathcal{P}$  if for all  $z$ ,

$$|p_z - q_z| \leq \epsilon p_z. \quad (3)$$

We say that  $\mathcal{Q}$  is an *additive  $\epsilon$ -approximation* of  $\mathcal{P}$  if

$$\frac{1}{2} \sum_z |p_z - q_z| \leq \epsilon. \quad (4)$$

Note that any multiplicative  $\epsilon$ -approximation is also an additive  $\epsilon/2$ -approximation, since summing Eq. (3) over all  $z$  produces Eq. (4). Here the factor of  $1/2$  is present so that  $\epsilon$  is the total variation distance between the probability distributions.

A *weak simulation with multiplicative (additive) error  $\epsilon > 0$*  of a family of quantum circuits is a classical randomized algorithm that samples from a distribution that is a

multiplicative (additive)  $\epsilon$ -approximation of the output distribution of the circuit. Note that from an experimental perspective, additive error is the more appropriate choice, since the fault-tolerance theorem merely guarantees additive closeness between the ideal and realized output distributions [7].

There are of course other notions of simulability of quantum circuits – such as strong simulation where one can compute individual output probabilities. We discuss these further in Section 6.

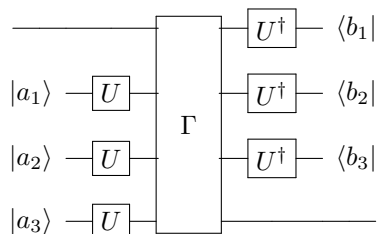
### 2.3 Postselection gadgets

Our results involve the use of postselection gadgets to simulate unitary operations. In this section, we introduce some terminology to describe these gadgets.

► **Definition 3.** Let  $U$  be a single-qubit operation. Let  $k, l \in \mathbb{Z}^+$  with  $k > l$ . A  $k$ -to- $l$   $U$ -CCC postselection gadget  $G$  is a postselected circuit fragment that performs the following procedure on an  $l$ -qubit system:

1. Introduce a set  $T$  of  $(k - l)$  ancilla registers in the state  $|a_1 \dots a_{k-l}\rangle$ , where  $a_1 \dots a_{k-l} \in \{0, 1\}^{k-l}$ .
2. Apply  $U^{\otimes(k-l)}$  to the set  $T$  of registers.
3. Apply a  $k$ -qubit Clifford operation  $\Gamma$  to both the system and ancilla.
4. Choose a subset  $S$  of  $(k - l)$  registers and apply  $(U^\dagger)^{\otimes(k-l)}$  to  $S$ .
5. Postselect on the subset  $S$  of qubits being in the state  $|b_1 \dots b_{k-l}\rangle$ , where  $b_1 \dots b_{k-l} \in \{0, 1\}^{k-l}$ .

An example of a 4-to-1  $U$ -CCC postselection gadget is the circuit fragment described by the following diagram:



Let  $G$  be a  $U$ -CCC postselection gadget as described in Definition 3. The *action*  $A(G)$  (also denoted  $A_G$ ) of  $G$  is defined to be the linear operation that it performs, i.e.

$$A(G) = A_G = \langle b_1 \dots b_l |_S \left( \prod_{i \in S} U_i^\dagger \right) \Gamma \left( \prod_{i \in T} U_i \right) |a_1 \dots a_l \rangle_T, \tag{5}$$

and the *normalized action* of  $G$ , when it exists, is

$$\tilde{A}_G = \frac{A_G}{(\det A_G)^{2^{-l}}}. \tag{6}$$

Note that the above normalization is chosen so that  $\det \tilde{A}_G = 1$ .

We say that a  $U$ -CCC postselection gadget  $G$  is *unitary* if there exists  $\alpha \in \mathbb{C} \setminus \{0\}$  and a unitary operator  $U$  such that  $A_G = \alpha U$ . It is straightforward to check that the following are equivalent conditions for gadget unitarity.



## 21:10 Complexity Classification of Conjugated Clifford Circuits

► **Lemma 4.** *A  $U$ -CCC postselection gadget  $G$  is unitary if and only if either one of the following holds:*

1. *There exists  $\gamma > 0$  such that  $A_G^\dagger A_G = \gamma I$ ,*
2.  *$\tilde{A}_G^\dagger \tilde{A}_G = I$ , i.e.  $\tilde{A}_G$  is unitary.*

Similarly, we say that a  $U$ -CCC postselection gadget  $G$  is *Clifford* if there exists  $\alpha \in \mathbb{C} \setminus \{0\}$  and a Clifford operator  $U$  such that  $A_G = \alpha U$ . The following lemma gives a necessary condition for a gadget to be Clifford.

► **Lemma 5.** *If  $G$  is a Clifford  $U$ -CCC postselection gadget, then*

$$A_G X A_G^\dagger \propto X \text{ or } A_G X A_G^\dagger \propto Y \text{ or } A_G X A_G^\dagger \propto Z, \quad (7)$$

and

$$A_G Z A_G^\dagger \propto X \text{ or } A_G Z A_G^\dagger \propto Y \text{ or } A_G Z A_G^\dagger \propto Z. \quad (8)$$

**Proof.** If  $G$  is a Clifford  $U$ -CCC postselection gadget, then there exists  $\alpha \in \mathbb{C} \setminus \{0\}$  and a Clifford operation  $\Gamma$  such that  $A_G = \alpha \Gamma$ . Since  $\Gamma$  is Clifford,  $\Gamma X \Gamma^\dagger$  is a Pauli operator. But  $\Gamma X \Gamma^\dagger \not\sim I$ , otherwise,  $X \sim I$ , which is a contradiction. Hence,  $\Gamma X \Gamma^\dagger \sim X$  or  $Y$  or  $Z$ , which implies Eq. (7). The proof of Eq. (8) is similar, with  $X$  replaced with  $Z$ . ◀

### 3 Weak simulation of CCCs with multiplicative error

#### 3.1 Classification results

In this section, we classify the hardness of weakly simulating  $U$ -CCCs as we vary  $U$ . As we shall see, it turns out that the classical simulation complexities of the  $U$ -CCCs associated with this notion of simulation are all of the following two types: the  $U$ -CCCs are either efficiently simulable, or are hard to simulate to constant multiplicative error unless the polynomial hierarchy collapses. To facilitate exposition, we will introduce the following terminology to describe these two cases: Let  $\mathcal{C}$  be a class of quantum circuits. Following the terminology in [34], we say that  $\mathcal{C}$  is in PWEAK if it is efficiently simulable in the weak sense by a classical computer. We say that  $\mathcal{C}$  is PH-supreme (or that it exhibits PH-supremacy) if it satisfies the property that if  $\mathcal{C}$  is efficiently simulable in the weak sense by a classical computer to constant multiplicative error, then the polynomial hierarchy (PH) collapses.

The approach we take to classifying the  $U$ -CCCs is to decompose each  $U$  into the form given by Eq. (2),

$$U = e^{i\alpha} R_z(\phi) R_x(\theta) R_z(\lambda), \quad (9)$$

and study how the classical simulation complexity changes as we vary  $\alpha, \phi, \theta$  and  $\lambda$ . Two simplifications can immediately be made. First, the outcome probabilities of the  $U$ -CCC are independent of  $\alpha$ , since  $\alpha$  appears only in a global phase. Second, the probabilities are also independent of  $\lambda$ . To see this, note that the outcome probabilities are all of the form:

$$|\langle b | R_z(-\lambda)^{\otimes n} V R_z(\lambda)^{\otimes n} | 0 \rangle|^2 = |\langle b | V | 0 \rangle|^2, \quad (10)$$

which is independent of  $\lambda$ . In the above expression,  $b \in \{0, 1\}^n$  and

$$V = R_x(-\theta)^{\otimes n} R_z(-\phi)^{\otimes n} \Gamma R_z(\phi)^{\otimes n} R_x(\theta)^{\otimes n}$$

for some Clifford circuit  $\Gamma$ . The equality follows from the fact that the computational basis states are eigenstates of  $R_z(\lambda)^{\otimes n}$  with unit-magnitude eigenvalues.



■ **Table 1** Complete complexity classification of  $U$ -CCCs (where  $U = R_z(\phi)R_x(\theta)$ ) with respect to weak simulation, as we vary  $\phi$  and  $\theta$ . The roman numerals in parentheses indicate the parts of Lemma 6 that are relevant to the corresponding box. All  $U$ -CCCs are either in PWEAK (i.e. can be efficiently simulated in the weak sense) or PH-supreme (i.e. cannot be simulated efficiently in the weak sense, unless the polynomial hierarchy collapses.)

$\phi \backslash \theta$	$\pi\mathbb{Z}$	$\frac{\pi}{2}\mathbb{Z}_{\text{odd}}$	$(\frac{\pi}{2}\mathbb{Z})^c$
$\frac{\pi}{2}\mathbb{Z}$	PWEAK (i, ii)	PWEAK (ii)	PH-supreme (iv)
$(\frac{\pi}{2}\mathbb{Z})^c$	PWEAK (i)	PH-supreme (iii)	PH-supreme (iv)

Hence, to complete the classification, it suffices to just restrict our attention to the two-parameter family  $\{R_z(\phi)R_x(\theta)\}_{\phi,\theta}$  of unitaries. We first prove the following lemma (see Table 1 for a summary):

► **Lemma 6.** *Let  $U = R_z(\phi)R_x(\theta)$ , where  $\phi, \theta \in [0, 2\pi)$ . Then*

■  *$U$ -CCCs are in PWEAK, if*  
(i)  $\phi \in [0, 2\pi)$  and  $\theta \in \pi\mathbb{Z}$ , or

(ii)  $\phi \in \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}$ .

■  *$U$ -CCCs are PH-supreme, if*  
(iii)  $\phi \notin \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ , or

(iv)  $\theta \notin \frac{\pi}{2}\mathbb{Z}$ .

We defer the proof of Lemma 6 to Sections 3.2 and 3.3. Lemma 6 allows us to prove our main theorem:

► **Theorem 7.** *Let  $U$  be a single-qubit unitary operator. Consider the following two statements:*

(A)  *$U$ -CCC is in PWEAK.*

(B) *There exists a single-qubit Clifford operator  $\Gamma \in \langle S, H \rangle$  and  $\lambda \in [0, 2\pi)$  such that<sup>9</sup>*

$$U \sim \Gamma R_z(\lambda). \tag{11}$$

Then,

1. (B) implies (A).
2. If the polynomial hierarchy is infinite, then (A) implies (B).

*In other words, if we assume that the polynomial hierarchy is infinite, then  $U$ -CCCs are PH-supreme if and only if they cannot be written in the form  $U \sim \Gamma R_z(\lambda)$ , where  $\Gamma$  is a Clifford circuit and  $R_z(\lambda)$  is a  $Z$ -rotation.*

**Proof.**

1. Since  $R_z(\lambda)|0\rangle \sim |0\rangle$ , it follows that for any  $\Gamma$ ,  $\Gamma R_z(\lambda)$ -CCCs have the same outcome probabilities as  $\Gamma$ -CCCs. But  $C$ -CCCs are efficiently simulable, by the Gottesman-Knill Theorem, since  $\Gamma \in \langle S, H \rangle$ . Hence,  $U$ -CCCs are in PWEAK.

<sup>9</sup> or alternatively, we could restrict the range of  $\lambda$  to be in  $[0, \pi]$ , since any factor of  $R_z(\pi/2) \sim S$  can be absorbed into the Clifford operator  $\Gamma$ .

## 21:12 Complexity Classification of Conjugated Clifford Circuits

2. Let  $U$  be such that  $U$ -CCCs are in PWEAK. Using the decomposition in Eq. (2), write  $U = e^{i\alpha} R_z(\phi) R_x(\theta) R_z(\lambda)$ . Since we assumed that the polynomial hierarchy is infinite, Lemma 6 implies that

a.  $\theta \in \pi\mathbb{Z}$ , or

b.  $\theta \in \frac{\pi}{2}\mathbb{Z}$  and  $\phi \in \frac{\pi}{2}\mathbb{Z}$ .

In Case (a),  $\theta \in 2\pi\mathbb{Z}$  or  $\pi\mathbb{Z}_{\text{odd}}$ . If  $\theta \in 2\pi\mathbb{Z}$ , then

$$U \sim R_z(\phi) R_x(2\pi\mathbb{Z}) R_z(\gamma) = I \cdot R_z(\phi + \gamma),$$

which is of the form given by Eq. (11). If  $\pi\mathbb{Z}_{\text{odd}}$ , then

$$U \sim R_z(\phi) R_x(\pi\mathbb{Z}_{\text{odd}}) R_z(\gamma) \sim R_z(\phi) X R_z(\gamma) = X R_z(\gamma - \phi),$$

which is again of the form given by Eq. (11).

In Case (b),

$$\begin{aligned} U &\in e^{i\alpha} R_z(\pi\mathbb{Z}/2) R_x(\pi\mathbb{Z}/2) R_z(\gamma) \\ &= e^{i\alpha} R_z(\pi\mathbb{Z}/2) H R_z(\pi\mathbb{Z}/2) H R_z(\gamma). \end{aligned} \quad (12)$$

But the elements of  $R_z(\pi\mathbb{Z}/2)$  are of the form  $S^j$ , for  $j \in \mathbb{Z}$ , up to a global phase. Therefore,  $R_z(\pi\mathbb{Z}/2) H R_z(\pi\mathbb{Z}/2) H$  is Clifford, and  $U$  is of the form Eq. (11). ◀

Hence, Theorem 7 tells us that under the assumption that the polynomial hierarchy is infinite,  $U$ -CCCs can be simulated efficiently (in the weak sense) if and only if  $U \sim \Gamma R_z(\lambda)$  for some single qubit Clifford operator  $\Gamma$ , i.e. if  $U$  is a Clifford operation times a  $Z$ -rotation.

### 3.2 Proofs of efficient classical simulation

In this section, we prove Cases (i) and (ii) of Lemma 6.

#### 3.2.1 Proof of Case (i): $\phi \in [0, 2\pi)$ and $\theta \in \pi\mathbb{Z}$

► **Theorem 8.** *Let  $U = R_z(\phi) R_x(\theta)$ . If  $\phi \in [0, 2\pi)$  and  $\theta \in \pi\mathbb{Z}$ , then  $U$ -CCCs are in PWEAK.*

**Proof.** First, we consider the case where  $\theta \in 2\pi\mathbb{Z}$ . In this case,  $U = R_z(\phi)$ , and the amplitudes of the  $U$ -CCC can be written as

$$\langle y | R_z(-\phi)^{\otimes n} \Gamma R_z(\phi)^{\otimes n} | x \rangle \sim \langle y | \Gamma | x \rangle \quad (13)$$

for some Clifford operation  $\Gamma$  and computational basis states  $|x\rangle$  and  $|y\rangle$ . By the Gottesman-Knill Theorem, these  $U$ -CCCs can be efficiently weakly simulated.

Next, we consider the case where  $\theta \in \pi\mathbb{Z}_{\text{odd}}$ . In this case,  $U = R_z(\phi) R_x(\pi) \sim R_z(\phi) X$ , and the amplitudes of the  $U$ -CCC can be written as

$$\langle y | X^{\otimes n} R_z(-\phi)^{\otimes n} \Gamma R_z(\phi)^{\otimes n} X^{\otimes n} | x \rangle \sim \langle \bar{y} | \Gamma | \bar{x} \rangle \quad (14)$$

for some Clifford operation  $\Gamma$  and computational basis states  $|x\rangle$  and  $|y\rangle$ , where  $\bar{z}$  is the bitwise negation of  $z$ . By the Gottesman-Knill Theorem, these  $U$ -CCCs can be efficiently weakly simulated.

Putting the above results together, we get that  $U$ -CCCs are in PWEAK. ◀

### 3.2.2 Proof of Case (ii): $\phi \in \frac{\pi}{2}\mathbb{Z}$ and $\theta \in \frac{\pi}{2}\mathbb{Z}$

► **Theorem 9.** *Let  $U = R_z(\phi)R_x(\theta)$ . If  $\phi \in \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}$ , then  $U$ -CCCs are in PWEAK.*

**Proof.** The elements of  $R_z(\frac{\pi}{2}\mathbb{Z})$  are of the form  $S^j$ , where  $j \in \mathbb{Z}$ , up to a global phase. Therefore,  $U = R_z(\phi)R_x(\theta) = R_z(\phi)HR_z(\theta)H$  is a Clifford operation, and so, the  $U$ -CCCs consist of only Clifford gates. By the Gottesman-Knill Theorem, these  $U$ -CCCs can be efficiently (weakly) simulated. ◀

## 3.3 Proofs of hardness

In this section, we prove Cases (iii) and (iv) of Lemma 6. Our proof uses postselection gadgets, similar to the techniques used in [14, 10]. One can also prove hardness using techniques from measurement-based-quantum computing, at least for certain  $U$ . We give such a proof in the appendix of the full version of our paper for the interested reader; we believe this proof may be more intuitive for those who are familiar with measurement-based quantum computing.

We start by proving a lemma that will be useful for the proofs of hardness.

► **Lemma 10.** *(Sufficient condition for PH-supremacy) Let  $U$  be a single-qubit gate. If there exists a unitary non-Clifford  $U$ -CCC postselection gadget  $G$ , then  $U$ -CCCs are PH-supreme.*

**Proof.** Suppose such a gadget  $G$  exists. Then, since the Clifford group plus any non-Clifford gate is universal [44, 45, 1], the Clifford group plus  $G$  must be universal on a single qubit. Then, by the inverse-free Solovay-Kitaev Theorem of Sardharwalla *et al.* [51], using polynomially many gates from the set  $G, H, S$  one can compile any desired one-qubit unitary  $V$  to inverse exponential accuracy (since in particular  $\langle H, S \rangle$  contains the Paulis). In particular, since any three-qubit unitary can be expressed as a product of a constant number of CNOTs and one-qubit unitaries, one can compile any gate in the set  $\{\text{CCZ, Controlled-H, all one-qubit gates}\}$  to inverse exponential accuracy with polynomial overhead.

In his proof that  $\text{PostBQP} = \text{PP}$ , Aaronson showed that postselected poly-sized circuits of the above gates can compute any language in PP [2]. Furthermore, as his postselection succeeds with inverse exponential probability, compiling these gates to inverse exponential accuracy is sufficient for performing arbitrary PP computations.

Hence, by using polynomially many gadgets for  $G$ , CNOT,  $H$  and  $S$ , one can compile Aaronson's circuits<sup>10</sup> for computing PP to inverse exponential accuracy, and hence these circuits can compute PP-hard problems. PH-supremacy then follows from the techniques of [14, 3]. Namely, a weak simulation of such circuits with constant multiplicative error would place  $\text{PP} \subseteq \text{BPP}^{\text{NP}} \subseteq \Delta_3$  by Stockmeyer counting, and hence by Toda's theorem this would result in the collapse of PH to the third level. In fact, by the arguments of Fujii *et al.* [23], one can collapse PH to the second level as well, by placing  $\text{coC=P}$  in SBP, and we refer the interested reader to their work for the complete argument. ◀

<sup>10</sup>More specifically, we compile the circuit given by  $(U^\dagger)^{\otimes n}$ , then Aaronson's circuit, then  $U^{\otimes n}$ , as we need to cancel the  $U$ 's at the beginning and the  $U^\dagger$ 's at the end in order to perform Aaronson's circuit which starts and measures in the computational basis. However as the  $U, U^\dagger$  are one-qubit gates, one can cancel them to inverse exponential accuracy using our gates, and hence this construction suffices.

### 3.3.1 Proof of Case (iii): $\phi \notin \frac{\pi}{2}\mathbb{Z}$ and $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$

Let  $U = R_z(\phi)R_x(\theta)$ . Consider the following  $U$ -CCC postselection gadget:

$$I(\phi, \theta) = \begin{array}{c} \text{---} \bullet \text{---} \boxed{U^\dagger} \text{---} \langle 0| \\ | \\ \boxed{U} \text{---} \bullet \text{---} \text{---} \\ | \\ |0\rangle \end{array} \quad (15)$$

We now prove some properties about  $I(\phi, \theta)$ .

► **Theorem 11.**

1. The action of  $I(\phi, \theta)$  is

$$A_{I(\phi, \theta)} = \begin{pmatrix} \cos^2 \frac{\theta}{2} & \frac{i}{2} \sin \theta e^{-i\phi} \\ -\frac{i}{2} \sin \theta e^{i\phi} & -\sin^2 \frac{\theta}{2} \end{pmatrix}. \quad (16)$$

2.  $I(\phi, \theta)$  is a unitary gadget if and only if  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ . When  $I(\phi, \theta)$  is unitary,

$$\tilde{A}_{I(\phi, \theta)} = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & i(-1)^k e^{-i\phi} \\ -i(-1)^k e^{i\phi} & -1 \end{pmatrix}, \quad (17)$$

where  $k = \frac{\theta}{\pi} - \frac{1}{2}$ .

3.  $I(\phi, \theta)$  is a Clifford gadget if and only if  $\phi \in \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ .
4.  $I(\phi, \theta)$  is a unitary non-Clifford gadget if and only if  $\phi \notin \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ .

**Proof.**

1. By direct calculation.
2. By Eq. (16),

$$A_{I(\phi, \theta)}^\dagger A_{I(\phi, \theta)} = \begin{pmatrix} \cos^2 \frac{\theta}{2} & \frac{i}{4} \sin(2\theta) e^{-i\phi} \\ -\frac{i}{4} \sin(2\theta) e^{i\phi} & \sin^2 \frac{\theta}{2} \end{pmatrix}. \quad (18)$$

If  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ , then  $A_{I(\phi, \theta)}^\dagger A_{I(\phi, \theta)} = \frac{1}{2}I$ , which implies that  $I(\phi, \theta)$  is a unitary gadget, by Lemma 4. Conversely, assume that  $I(\phi, \theta)$  is a unitary gadget. Suppose that  $\theta \notin \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ . Then  $\sin(2\theta) \neq 0$ , which implies that  $A_{I(\phi, \theta)}^\dagger A_{I(\phi, \theta)} \not\propto I$ , which is a contradiction. Hence,  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ .

Next,  $k = \frac{\theta}{\pi} - \frac{1}{2}$  implies that  $\theta = \frac{\pi}{2}(2k+1)$ . Since  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ , it follows that  $k \in \mathbb{Z}$ . Then  $\sin \theta = (-1)^k$ ,  $\cos^2 \frac{\theta}{2} = \frac{1}{2}$  and  $\sin^2 \frac{\theta}{2} = \frac{1}{2}$ . Hence,

$$A_{I(\phi, \theta)} = \begin{pmatrix} \frac{1}{2} & \frac{i}{2}(-1)^k e^{-i\phi} \\ -\frac{i}{2}(-1)^k e^{i\phi} & -\frac{1}{2} \end{pmatrix}. \quad (19)$$

Hence,  $\det A_{I(\phi, \theta)} = -\frac{1}{2}$ . Plugging this and Eq. (19) into Eq. (6) gives Eq. (17).

3. ( $\Leftarrow$ ) Let  $\phi \in \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ . Write  $\phi = \frac{\pi}{2}l$  and  $\theta = \frac{\pi}{2}(2k+1)$ . Then, by Eq. (17),

$$\tilde{A}_{I(\phi, \theta)} = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & i^{1+2k+3l} \\ i^{3+2k+l} & -1 \end{pmatrix}. \quad (20)$$

Now, it is straightforward to check that for all  $k, l \in \mathbb{Z}$ ,  $\tilde{A}_{I(\phi, \theta)} X \tilde{A}_{I(\phi, \theta)}^\dagger \in \{-X, Z, -Z\}$  and  $\tilde{A}_{I(\phi, \theta)} Z \tilde{A}_{I(\phi, \theta)}^\dagger \in \{-Y, X, Y, -X\}$ . This shows that  $\tilde{A}_{I(\phi, \theta)}$  maps the Pauli group to itself, under conjugation, which implies that  $\tilde{A}_{I(\phi, \theta)}$  is Clifford.

( $\Rightarrow$ ) Assume that  $I(\phi, \theta)$  is a Clifford gadget. Suppose that  $\phi \notin \frac{\pi}{2}\mathbb{Z}$  or  $\theta \notin \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ . But  $I(\phi, \theta)$  is unitary, and hence,  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ . So  $\phi \notin \frac{\pi}{2}\mathbb{Z}$ . By Lemma 5,  $\tilde{A}_{I(\phi, \theta)} X \tilde{A}_{I(\phi, \theta)}^\dagger \sim X$  or  $Y$  or  $Z$ . But, as we compute,

$$\tilde{A}_{I(\phi, \theta)} X \tilde{A}_{I(\phi, \theta)}^\dagger = \begin{pmatrix} (-1)^k \sin \phi & -e^{-i\phi} \cos \phi \\ -e^{i\phi} \cos \phi & -(-1)^k \sin \phi \end{pmatrix}. \quad (21)$$

If  $\tilde{A}_{I(\phi, \theta)} X \tilde{A}_{I(\phi, \theta)}^\dagger \sim X$  or  $Y$ , then  $\sin \phi = 0$ , which is a contradiction, since  $\phi \notin \frac{\pi}{2}\mathbb{Z}$ . Hence,  $\tilde{A}_{I(\phi, \theta)} X \tilde{A}_{I(\phi, \theta)}^\dagger \sim Z$ , which implies that  $\cos \phi = 0$ . But this also contradicts  $\phi \notin \frac{\pi}{2}\mathbb{Z}$ . Hence,  $\phi \in \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ .

4. Follows from Parts 2 and 3 of Theorem 11.  $\blacktriangleleft$

**► Theorem 12.** *Let  $U = R_z(\phi)R_x(\theta)$ . If  $\phi \notin \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ , then  $U$ -CCCs are PH-supreme.*

**Proof.** By Theorem 11, when  $\phi \notin \frac{\pi}{2}\mathbb{Z}$  and  $\theta \in \frac{\pi}{2}\mathbb{Z}_{\text{odd}}$ , then  $I(\phi, \theta)$  is a unitary non-Clifford  $U$ -CCC postselection gadget. Hence, by Lemma 10,  $U$ -CCCs are PH-supreme.  $\blacktriangleleft$

### 3.3.2 Proof of Case (iv): $\theta \notin \frac{\pi}{2}\mathbb{Z}$

Let  $U = R_z(\phi)R_x(\theta)$ . Consider the following  $U$ -CCC postselection gadget:

$$J(\phi, \theta) = \begin{array}{c} \text{-----} \bullet \text{-----} \\ | \\ |0\rangle \text{---} [U] \text{---} [S] \text{---} \bullet \text{---} [U^\dagger] \text{---} \langle 0| \end{array} \quad (22)$$

We now prove some properties about  $J(\phi, \theta)$ .

**► Theorem 13.**

1. *The action of  $J(\phi, \theta)$  is*

$$\begin{aligned} A_{J(\phi, \theta)} &= \frac{1}{\sqrt{2}} e^{-i\frac{\pi}{4}} \begin{pmatrix} i + \cos \theta & 0 \\ 0 & 1 + i \cos \theta \end{pmatrix} \\ &= \frac{i}{\sqrt{2}} e^{-i\frac{\pi}{4}} \sqrt{1 + \cos^2 \theta} S^\dagger R_z(2 \tan^{-1}(\cos \theta)). \end{aligned} \quad (23)$$

2.  *$J(\phi, \theta)$  is a unitary gadget for all  $\theta, \phi \in [0, 2\pi)$ . The normalized action is*

$$\tilde{A}_{J(\phi, \theta)} \sim S^\dagger R_z(2 \tan^{-1}(\cos \theta)). \quad (24)$$

3.  *$J(\phi, \theta)$  is a Clifford gadget if and only if  $\theta \in \frac{\pi}{2}\mathbb{Z}$ .*

4.  *$J(\phi, \theta)$  is a unitary non-Clifford gadget if and only if  $\theta \notin \frac{\pi}{2}\mathbb{Z}$ .*

**Proof.**

1. By direct calculation.

2. The determinant of  $A_{J(\phi, \theta)}$  is

$$\det A_{J(\phi, \theta)} = \frac{1}{2}(1 + \cos^2 \theta) \neq 0 \quad (25)$$

for all  $\theta$  and  $\phi$ . Hence,  $A_{J(\phi, \theta)} \propto S^\dagger R_z(2 \tan^{-1}(\cos \theta))$  for all  $\theta$  and  $\phi$ , which implies that  $J(\phi, \theta)$  is a unitary gadget for all  $\theta$  and  $\phi$ .

Hence,

$$\tilde{A}_{J(\phi, \theta)} = \frac{A_{J(\phi, \theta)}}{\sqrt{\det A_{J(\phi, \theta)}}} = i e^{-i\frac{\pi}{4}} S^\dagger R_z(2 \tan^{-1}(\cos \theta)).$$

## 21:16 Complexity Classification of Conjugated Clifford Circuits

3.

$$\begin{aligned}
 J(\phi, \theta) \text{ is a Clifford gadget} &\Leftrightarrow S^\dagger R_z(2 \tan^{-1}(\cos \theta)) \text{ is Clifford} \\
 &\Leftrightarrow R_z(2 \tan^{-1}(\cos \theta)) \text{ is Clifford} \\
 &\Leftrightarrow 2 \tan^{-1}(\cos \theta) \in \frac{\pi}{2}\mathbb{Z} \quad \text{by Fact 1} \\
 &\Leftrightarrow \cos \theta \in \{0, 1, -1\} \\
 &\Leftrightarrow \theta \in \frac{\pi}{2}\mathbb{Z}.
 \end{aligned} \tag{26}$$

4. Follows from Parts 2 and 3 of Theorem 13. ◀

► **Theorem 14.** *Let  $U = R_z(\phi)R_x(\theta)$ . If  $\theta \notin \frac{\pi}{2}\mathbb{Z}$ , then  $U$ -CCCs are PH-supreme.*

**Proof.** By Theorem 13, when  $\theta \notin \frac{\pi}{2}\mathbb{Z}$ , then  $I(\phi, \theta)$  is a unitary non-Clifford  $U$ -CCC postselection gadget. Hence, by Lemma 10,  $U$ -CCCs are PH-supreme. ◀

### 4 Weak simulation of CCCs with additive error

Here we show how to achieve additive hardness of simulating conjugated Clifford circuits, under additional hardness assumptions. Specifically, we will show that under these assumptions, there is no classical randomized algorithm which given a one-qubit unitary  $U$  and a Clifford circuit  $V$ , samples the output distribution of  $V$  conjugated by  $U$ 's up to constant  $\ell_1$  error.

In the following, let  $V$  be a Clifford circuit on  $n$  qubits,  $U$  be a one-qubit unitary which is not a  $Z$ -rotation times a Clifford, and  $y \in \{0, 1\}^n$  be an  $n$ -bit string. Define

$$p_{y,U,V} = |\langle y | (U^\dagger)^{\otimes n} V U^{\otimes n} |0^n\rangle|^2.$$

In other words  $p_{y,U,V}$  is the probability of outputting the string  $y$  when applying the circuit  $V$  conjugated by  $U$ 's to the all 0's state, and then measuring in the computational basis. Let the corresponding probability distribution on  $y$ 's given  $U$  and  $V$  be denoted  $D(U, V)$ .

► **Theorem 15.** *Assuming that PH is infinite and Conjecture 16, then there is no classical algorithm which given a one-qubit unitary  $U$  and an  $n$ -qubit Clifford circuit  $V$ , outputs a probability distribution which is  $1/100$  close to  $D(U, V)$  in total variation distance.*

► **Conjecture 16.** *For any  $U$  which is not equal to a  $Z$ -rotation times a Clifford, it is #P-hard to approximate a  $6/50$  fraction of the  $p_{y,U,V}$  over the choice of  $y, V$  to within multiplicative error  $1/2 + o(1)$ .*

In order to prove this we'll actually prove a more general theorem described below; the result will then follow from simply setting  $a = c = 1/5$ ,  $\epsilon = 1/100$ . One can in general plug in any values they like subject to the constraints; for instance one can strengthen the hardness assumption by assuming computing a smaller fraction of the  $p_{y,U,V}$  is still #P-hard to obtain larger allowable error in the simulation. These parameters are similar to those appearing in other hardness conjectures, for example those used for IQP [15].

► **Theorem 17.** *Pick constants  $0 < \epsilon, a, c < 1$  such that  $(1-a)^2/2 - c > 0$  and  $\frac{2\epsilon}{ac} < 1$ . Then assuming Conjecture 18, given a one-qubit unitary  $U$  and an  $n$ -qubit Clifford circuit  $V$ , one cannot weakly simulate the distribution  $D(U, V)$  with a randomized classical algorithm with total variation distance error  $\epsilon$ , unless the polynomial hierarchy collapses to the third level.*

► **Conjecture 18.** *For any  $U$  which is not equal to a  $Z$ -rotation times a Clifford, it is #P-hard to multiplicatively approximate  $(1 - a)^2/2 - c$  fraction of the  $p_{y,U,V}$  over the choice of  $(y, V)$ , up to multiplicative error  $\frac{2\epsilon}{ac} + o(1)$ .*

**Proof of Theorem 17.** Suppose by way of contradiction that there exists a classical poly-time randomized algorithm which given inputs  $U, V$  outputs samples from a distribution  $D'(U, V)$  such that  $\frac{1}{2}|D(U, V) - D'(U, V)|_1 < \epsilon$ . In particular, let  $q_{y,U,V}$  be the probability that  $D'(U, V)$  outputs  $y$  – i.e. the probability that the simulation outputs  $y$  under inputs  $U, V$ .

By our simulation assumption, for all  $U, V$  we have that  $\sum_y |q_{y,U,V} - p_{y,U,V}| \leq 2\epsilon$ . Therefore by Markov's inequality, given our constant  $0 < c < 1$ , we have that for all  $U$  and  $V$  there exists a set  $S' \subseteq \{0, 1\}^n$  of output strings  $y$  of size  $|S'|/2^n > 1 - c$ , such that for all  $y \in S'$ ,

$$|q_{y,U,V} - p_{y,U,V}| \leq \frac{2\epsilon}{c2^n}.$$

In particular, by averaging over  $V$ 's, we see that for any  $U$  as above, there exists a set  $S \subset \{0, 1\}^n \times \mathcal{C}$  of pairs  $(y, V)$  such that for all  $(y, V) \in S$ ,  $|q_{y,U,V} - p_{y,U,V}| \leq \frac{2\epsilon}{c2^n}$ . Furthermore  $S$  has measure at least  $(1 - c)$  over a uniformly random choice of  $(y, V)$ .

We now show the following anticoncentration lemma (similar theorems were shown independently in [29, 39, 31]):

► **Lemma 19.** *For any fixed  $U$  and  $y$  as above, and for any constant  $0 < a < 1$ , we have that at least  $\frac{(1-a)^2}{2}$  fraction of the Clifford circuits  $V$  have the property that*

$$p_{y,U,V} \geq \frac{a}{2^n}.$$

We will prove Lemma 19 shortly. First, we will show why this implies Theorem 17. In particular, by averaging Lemma 19 over  $y$ 's, we see that for any  $U$  as above, there exists a set  $T \subset \{0, 1\}^n \times \mathcal{C}$  of pairs  $(y, V)$  such that for all  $(y, V) \in T$ ,  $p_{y,U,V} \geq \frac{a}{2^n}$ . Furthermore  $T$  has measure at least  $\frac{(1-a)^2}{2}$  over a uniformly random choice of  $(y, V)$ . Since we assumed that  $(1 - a)^2/2 + (1 - c) > 1$ , then  $S \cap T$  must be nonempty, and in particular must contain  $(1 - a)^2/2 - c$  fraction of the pairs  $(y, V)$ . On this set  $S \cap T$ , we have that

$$q_{y,U,V} \leq p_{y,U,V} + \frac{2\epsilon}{c2^n} = p_{y,U,V} + \frac{2\epsilon}{ac} \frac{a}{2^n} \leq \left(1 + \frac{2\epsilon}{ac}\right) p_{y,U,V},$$

and likewise

$$q_{y,U,V} \geq p_{y,U,V} - \frac{2\epsilon}{c2^n} = p_{y,U,V} - \frac{2\epsilon}{ac} \frac{a}{2^n} \geq \left(1 - \frac{2\epsilon}{ac}\right) p_{y,U,V}.$$

Since  $1 - \frac{2\epsilon}{ac} > 0$  (which we guaranteed by assumption),  $q_{y,U,V}$  is a multiplicative approximation to  $p_{y,U,V}$  with multiplicative error  $\frac{2\epsilon}{ac}$  for  $(y, V)$  in the set  $S \cap T$ . The set  $S \cap T$  contains at least  $(1 - a)^2/2 - c$  fraction of the total pairs  $(y, V)$ .

On the other hand, by Conjecture 18 we have that computing a  $(1 - a)^2/2 - c$  fraction of the  $p_{y,U,V}$  to this level of multiplicative error is a #P-hard task. So approximating  $p_{y,U,V}$  to this level of multiplicative error for this fraction of outputs is both #P-hard, and achievable by our simulation algorithm. This collapses PH to the third level by known arguments [3, 14]. In particular, by applying Stockmeyer's approximate counting algorithm [55] to  $p_{y,U,V}$ , one can multiplicatively approximate  $q_{y,U,V}$  to multiplicative error  $\frac{1}{\text{poly}}$  in  $\text{FBPP}^{\text{NP}}$  for those

## 21:18 Complexity Classification of Conjugated Clifford Circuits

elements in  $S \cap T$ . But since  $q_{y,U,V}$  is a  $\frac{2\epsilon}{ac}$ -approx to  $p_{y,U,V}$ , this is a  $\frac{2\epsilon}{ac} + o(1)$  multiplicative approximation to  $p_{y,U,V}$  in  $S \cap T$ . Hence a #P-hard quantity is in  $\text{FBPP}^{\text{NP}}$ . This collapses PH to the third level by Toda's theorem [57].

To complete our proof of Theorem 17, we will prove Lemma 19.

**Proof of Lemma 19.** To prove this, we will make use of the fact that the Clifford group is an exact 2-design<sup>11</sup> [58, 59]. The fact that the Clifford group is a 2-design means that for any polynomial  $p$  over the variables  $\{V_{ij}\}$  and their complex conjugates, which is of degree at most 2 in the  $V_{ij}$ 's and degree at most 2 in the  $V_{ij}^*$ 's, we have that

$$\frac{1}{|C|} \sum_{V \in C} p(V, V^*) = \int p(V, V^*) dV,$$

where  $C$  denotes the Clifford group and the integral  $dV$  is taken over the Haar measure. In other words, the expectation values of low-degree polynomials in the entries of the matrices are exactly identical to the expectation values over the Haar measure.

In particular, note that  $p_{y,U,V}$  is a degree-1 polynomial in the entries of  $V$  and their complex conjugates, and  $p_{y,U,V}^2$  is a degree-2 polynomial in these variables. Therefore, since the Clifford group is an exact 2-design, we have that for any  $y$  and  $U$ ,

$$\frac{1}{|C|} \sum_{V \in C} p_{y,U,V} = \int p_{y,U,V} dV = \frac{1}{2^n}$$

and

$$\frac{1}{|C|} \sum_{V \in C} p_{y,U,V}^2 = \int p_{y,U,V}^2 dV = \frac{2}{2^{2n} - 1} \left(1 - \frac{1}{2^n}\right),$$

where the values of these integrals over the Haar measure are well known – see for instance Appendix D of [30].

Following [15], we now invoke the Paley-Zygmund inequality, which states that:

► **Fact 20.** *Given a parameter  $0 < a < 1$ , and a non-negative random variable  $p$  of finite variance, we have*

$$\Pr[p \geq a\mathbb{E}[p]] \geq (1 - a)^2 \mathbb{E}[p]^2 / \mathbb{E}[p^2].$$

Applying this inequality to the random variable  $p_{y,U,V}$  over the choice of the Clifford circuit  $V$ , we have that

$$\Pr_V \left[ p_{y,U,V} \geq \frac{a}{2^n} \right] \geq (1 - a)^2 \frac{2^{-2n}}{\frac{2 - 2^{-n+1}}{2^{2n} - 1}} = (1 - a)^2 \frac{1 - 2^{-2n}}{2 - 2^{-n+1}} \geq \frac{(1 - a)^2}{2}$$

which implies the claim. ◀

This completes the proof of Theorem 17. ◀

---

<sup>11</sup>The Clifford group is also a 3-design, but we will only need the fact it is a 2-design for our proof.



## 5 Evidence in favor of hardness conjecture

In Section 4, we saw that by assuming an average case hardness conjecture (namely Conjecture 18), we could show that a weak simulation of CCCs to additive error would collapse the polynomial hierarchy. A natural question is: what evidence do we have that Conjecture 18 is true?

In this section, we show that the worst-case version of Conjecture 18 is true. In fact, we show that for any  $U \neq CR_Z(\theta)$  for a Clifford  $C$ , there exists a Clifford circuit  $V$  and an output  $y$  such that computing  $p_{y,U,V}$  is  $\#P$ -hard to constant multiplicative error. Therefore certainly *some* output probabilities of CCCs are  $\#P$ -hard to compute. Conjecture 18 is merely conjecturing further that computing a large fraction of such output probabilities is just as hard.

► **Theorem 21** (Worst-case version of Conjecture 18). *For any  $U$  which is not equal to a  $Z$ -rotation times a Clifford, there exists a Clifford circuit  $V$  and string  $y \in \{0,1\}^n$  such that it is  $\#P$ -hard to multiplicatively approximate a  $p_{y,U,V}$  to multiplicative error  $1/2 - o(1)$ .*

**Proof.** This follows from combining the ideas from the proof of Lemma 6 with previously known facts about BQP. In particular, we will use the following facts:

1. There exists a uniform family of poly-size BQP<sup>12</sup> circuits  $C_x$  where  $x \in \{0,1\}^n$  using a gate set with algebraic entries such that computing  $|\langle 0^n | C_x | 0^n \rangle|^2$  to multiplicative error  $1/2$  is  $\#P$ -hard [15].
2. For any poly-sized quantum circuit  $C$  over a gate set with algebraic entries, any non-zero output probability has magnitude at least inverse exponential [35].
3. As shown in the proof of Theorem 7, for any  $U$  which is not a Clifford gate times a  $Z$  rotation, there is a postselection gadget  $G$  which performs a unitary but non-Clifford one-qubit operation. Furthermore all ancilla qubits in  $G$  begin in the state  $|0\rangle$ .

From these facts, we can now prove the theorem. Let  $p = |\langle 0^n | C_x | 0^n \rangle|^2$ . By Fact 2, the circuit  $C_x$  from Fact 1 either has  $p = 0$  or  $p \geq 2^{-O(n^c)}$  for some constant  $c$ . Now suppose we compile the circuit  $C_x$  from Fact 1 using Clifford gates plus the postselection gadget  $G$  – call this new circuit with postselection  $C'_x$ . By Sardharwalla *et al.* [51] we can compile this circuit with accuracy  $\epsilon = 2^{-O(n^c)-100}$  with only polynomial overhead.

Let  $\ell \in \{0,1\}^k$  be the string of postselection bits of the circuit  $C'_x$  (which without loss of generality are the last bits of the circuit), and let  $\alpha$  is the probability that all postselections succeed. Note  $\alpha$  is a known and easily calculated quantity, since each postselection gadget is unitary so succeeds with a known constant probability.

Let  $p' = |\langle 0^n \ell | C'_x | 0^{n+k} \rangle|^2 / \alpha$ . Then we have that:

- If  $p = 0$  then  $p' \leq 2^{-O(n^c)-100}$ .
- If  $p \neq 0$  then  $p - 2^{-O(n^c)-100} \leq p' \leq p + 2^{-O(n^c)-100}$ . Since  $p \geq 2^{-O(n^c)}$ , this is a multiplicative approximation to  $p$  with error  $2^{-100}$ .

Now suppose that one can compute  $|\langle 0^n \ell | C'_x | 0^{n+k} \rangle|^2$  to multiplicative error  $\gamma$  to be chosen shortly. Then immediately one can compute  $p' = |\langle 0^n \ell | C'_x | 0^{n+k} \rangle|^2 / \alpha$  to the same amount of multiplicative error – call this estimate  $p''$ . By the above argument, if  $p = 0$  then  $p'' < 2^{-O(n^c)-100}(1+\gamma)$ . On the other hand if  $p > 0$  then  $p' > 2^{-O(n^c)}$ , so  $p'' > 2^{-O(n^c)}(1-\gamma)$ . So long as  $\gamma$  is chosen such that  $2^{-100}(1+\gamma) < (1-\gamma)$  these two cases can be distinguished – which holds in particular if  $\gamma \approx 1/2$ .

<sup>12</sup>Even IQP suffices here [15].

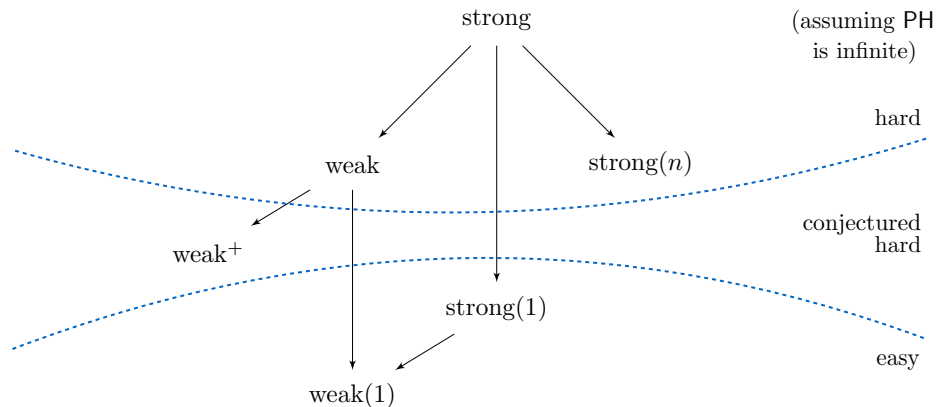
Therefore, if  $p'' < 2^{-O(n^c)}$  then we can infer that  $p = 0$ . If  $p'' > 2^{-O(n^c)}(1 - \gamma)$ , then  $p > 0$  so  $p''$  is a  $\gamma$  approximation to  $p'$  and hence a  $\gamma + 2^{-100} + \gamma 2^{-100}$  approximation to  $p$ . In either case we have computed a  $\gamma + 2^{-100} + \gamma 2^{-100}$  approximation to  $p$ . Therefore, if  $\gamma = 1/2 - 2^{-99}$ , then we have computed a  $1/2$ -multiplicative approximation to  $p$ , which is  $\#P$ -hard by Fact 1. Therefore, computing some the probability that the CCC corresponding to  $C'_x$  outputs  $|0^n \ell\rangle$  to multiplicative error  $1/2 - 2^{-99}$  is  $\#P$ -hard. One can similarly improve this hardness to  $1/2 - o(1)$ .  $\blacktriangleleft$

Given that the worst-case version of Conjecture 18 is true, a natural question to ask is how difficult it would be to prove the average-case conjecture. To do so would in particular prove quantum advantage over classical computation with realistic error, and merely assuming the polynomial hierarchy is infinite. In some ways this would be stronger evidence for quantum advantage over classical computation than Shor's factoring algorithm, as there are no known negative complexity-theoretic consequences if factoring is contained in P.

Unfortunately, recent work has shown that proving Conjecture 18 would be a difficult task. Specifically, Aaronson and Chen [4] demonstrated an oracle relative to which PH is infinite, but classical computers can efficiently weakly simulate quantum devices to constant additive error. Therefore, any proof which establishes quantum advantage with additive error under the assumption that PH is infinite must be non-relativizing. In particular this implies any proof of Conjecture 18 would require non-relativizing techniques – in other words it could not remain true if one allows for classical oracle class in the circuit. This same barrier holds for proving the similar average-case hardness conjectures to show advantage for Boson Sampling, IQP, DQC1, or Fourier sampling. Therefore any proof of Conjecture 18 would require facts specific to the Clifford group. We leave this as an open problem. We also note that it remains open to prove the average-case *exact* version of Conjecture 18 - i.e. whether it is hard to exactly compute a large fraction of  $p_{y,U,C}$ . We believe this may be a more tractable problem to approach than Conjecture 18. However this remains open, as is the analogous average-case exact conjecture corresponding to IQP. We note the corresponding average-case exact conjecture for Boson Sampling and Fourier sampling are known to be true [3, 22], though these models are not known to anticoncentrate.

## 6 Summary of simulability of CCCs

For completeness, in this section we summarize the simulability of  $U$ -CCCs when  $U$  is not a Clifford rotation times a  $Z$  rotation. There are various notions of classical simulation at play here. The results of this paper so far have focused on notions of approximate *weak* simulation. A *weak* simulation of a family of quantum circuits is a classical randomized algorithm that samples from the same distribution as the output distribution of the circuit. On the other hand, a *strong* simulation of a family of quantum circuits is a classical algorithm that computes not only the joint probabilities, but also any marginal probabilities of the outcomes of the measurements in the circuit. Following [34], we can further refine these definitions according to the number of qubits being measured: a strong(1) simulation computes the marginal output probabilities on individual qubits, and a strong( $n$ ) simulation computes the probability of output strings  $y \in \{0, 1\}^n$ . Similarly, a weak(1) simulation samples from the marginal output probabilities on individual qubits, and a weak( $n$ ) simulation samples from  $p(y_1, \dots, y_n)$ . A weak<sup>+</sup> simulation samples from the same distribution on all  $n$  output qubits up to constant additive error. Our previous results have shown that efficient weak( $n$ ) simulations (Theorem 7), weak<sup>+</sup> simulations (Theorem 17), and strong( $n$ ) simulations (Theorem 21) of CCCs are implausible. However it is natural to ask if it is possible to simulate single output probabilities



■ **Figure 1** Relationships between different notions of classical simulation and summary of the hardness of simulating CCCs. An arrow from  $A$  to  $B$  ( $A \rightarrow B$ ) means that an efficient  $A$ -simulation of a computational task implies that there is an efficient  $B$ -simulation for the same task. Note also that a weak( $n$ ) simulation exists if and only if a weak simulation exists. For a proof of these relationships, see [34]. The two curves indicate the boundary between efficiencies of simulation of  $U$ -CCCs, where  $U$  is not a Clifford operation times a  $Z$  rotation. “Hard” means that an efficient simulation of  $U$ -CCCs is not possible, unless PH collapses. “Conjectured hard” means that an efficient simulation of  $U$ -CCCs is not possible, if we assume Conjecture 18. “Easy” means that an efficient simulation of  $U$ -CCCs exists. Note that when  $U$  is a Clifford operation times a  $Z$  rotation, all the above notions become easy.

of CCCs. It turns out the answer to this question is yes. This follows immediately from Theorem 5 of [34], which showed more generally that Clifford circuits with product inputs or measurements have an efficient strong(1) and weak(1) simulation. Therefore this completes the complexity classification of the simulability of such circuits. We note that IQP has identical properties in this regard. This emphasizes that the difficulty in simulating CCCs (or IQP circuits) comes from the difficulty of simulating all of the *marginal* probability distributions contained in the output distribution, where the marginal is taken over a large number of output bits. The probabilities of computing individual output bits of either model are easy for classical computation. This is summarized in Figure 1.

## 7 Open Problems

Our work leaves open a number of problems.

- What is the computational complexity of commuting CCCs? In other words, can the gate set  $CZ, S$  conjugated by a one-qubit gate  $U$  ever give rise to quantum advantage? Note that this does not follow from Bremner, Jozsa and Shepherd’s results [14], as their hardness proof uses the gate set  $CZ, T$  or  $CCZ, CZ, Z$  conjugated by one-qubit gates. If this is true, it would say that the “intersection” of CCCs and IQP remains computationally hard. One can also consider the computational power of arbitrary fragments of the Clifford group, which were classified in [27]. Perhaps by studying such fragments of the Clifford group one could achieve hardness with lower depth circuits (see additional question below).

- We showed that Clifford circuits conjugated by tensor-product unitaries are difficult to simulate classically. A natural extension of this question is: suppose your gate set consists of all two-qubit Clifford gates, conjugated by a unitary  $U$  which is *not* a tensor product of the same one-qubit gate. Can one show that all such circuits are difficult to simulate classically (say exactly)? Such a theorem could be a useful step towards classifying the power of all two-qubit gate sets.
- Generic Clifford circuits have a depth which is linear in the number of qubits [5]. In particular the lowest-depth decomposition for a generic Clifford circuit over  $n$  qubits to date has depth  $14n - 4$  [41]. Such depth will be difficult to achieve in near-term quantum devices without error-correction. As a result, others have considered quantum supremacy experiments with lower-depth circuits. For instance, Bremner, Shepherd and Montanaro showed advantage for a restricted version of IQP circuits with depth  $O(\log n)$  [16] with long-range gates (which becomes depth  $O(n^{1/2} \log n)$  if one uses SWAP gates to simulate long-range gates using local operations on a square lattice). We leave open the problem of determining if quantum advantage can be achieved with CCCs of lower depth (say  $O(n^{1/2})$  or  $O(n^{1/3})$ ) with local gates only.
- In order to establish quantum supremacy for CCCs, we conjectured that it is #P-hard to approximate a large fraction of the output probabilities of randomly chosen CCCs (Conjecture 18). Is it also #P-hard to exactly compute that large of a fraction of the output probabilities? This is a necessary but not sufficient condition for Conjecture 18 to be true, and we believe it may be a more approachable problem.

---

## References

- 1 Universal sets of gates for  $SU(3)$ ?, 2012. Accessed: 2017-08-01. URL: <https://csttheory.stackexchange.com/questions/11308/universal-sets-of-gates-for-su3>.
- 2 Scott Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 461, pages 3473–3482. The Royal Society, 2005.
- 3 Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM Symposium on Theory of Computing*, pages 333–342. ACM, 2011.
- 4 Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. *Proc. CCC*, 2017.
- 5 Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, 2004.
- 6 Scott Aaronson, Daniel Grier, and Luke Schaeffer. The classification of reversible bit operations. In *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, 2017.
- 7 Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the twenty-ninth annual ACM Symposium on Theory of Computing*, pages 176–188. ACM, 1997.
- 8 Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *arXiv:1608.00263*, 2016.
- 9 Adam Bouland and Scott Aaronson. Generation of universal linear optics by any beam splitter. *Physical Review A*, 89(6):062316, 2014.
- 10 Adam Bouland, Laura Mancinska, and Xue Zhang. Complexity classification of two-qubit commuting hamiltonians. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPICs*, pages 28:1–

- 28:33. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.28.
- 11 Sergey Bravyi, David Gosset, and Robert Koenig. Quantum advantage with shallow circuits. *arXiv:1704.00690*, 2017.
  - 12 Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Physical Review A*, 86(5):052329, 2012.
  - 13 Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, 2005.
  - 14 Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20100301. The Royal Society, 2010.
  - 15 Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Physical Review Letters*, 117(8):080501, 2016.
  - 16 Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Achieving quantum supremacy with sparse and noisy commuting quantum computations. *Quantum*, 1, 2017.
  - 17 Hans J Briegel, David E Browne, W Dür, Robert Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009.
  - 18 Christopher M Dawson and Michael A Nielsen. The Solovay-Kitaev algorithm. *Quantum Information & Computation*, 6(1):81–95, 2006.
  - 19 David P. DiVincenzo and Peter W. Shor. Fault-tolerant error correction with efficient quantum codes. *Phys. Rev. Lett.*, 77:3260–3263, Oct 1996. doi:10.1103/PhysRevLett.77.3260.
  - 20 Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Physical Review Letters*, 102(11):110502, 2009.
  - 21 Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv:1602.07674*, 2016.
  - 22 Bill Fefferman and Christopher Umans. On the power of quantum fourier sampling. In Anne Broadbent, editor, *11th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2016, September 27-29, 2016, Berlin, Germany*, volume 61 of *LIPIcs*, pages 1:1–1:19. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.TQC.2016.1.
  - 23 Keisuke Fujii, Hirotada Kobayashi, Tomoyuki Morimae, Harumichi Nishimura, Shuhei Tamate, and Seiichiro Tani. Impossibility of classically simulating one-clean-qubit computation. *arXiv:1409.6777*, 2014.
  - 24 Daniel Gottesman. Stabilizer codes and quantum error correction. *Ph.D. Thesis, California Institute of Technology, arXiv:quant-ph/9705052*, 1997.
  - 25 Daniel Gottesman. The Heisenberg representation of quantum computers. *Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, pages 32–43, 1999.
  - 26 Chris Granade and Ben Criger. QuaEC: Quantum error correction analysis in Python. <http://www.cgranade.com/python-quaec/groups.html#>, 2012. Accessed: 2017-06-01.
  - 27 Daniel Grier and Luke Schaeffer. The classification of stabilizer operations over qubits. *arXiv:1603.03999*, 2016.
  - 28 Amihay Hanany and Yang-Hui He. A monograph on the classification of the discrete subgroups of  $SU(4)$ . *Journal of High Energy Physics*, 2001(02):027, 2001.
  - 29 Dominik Hangleiter, Juan Bermejo-Vega, Martin Schwarz, and Jens Eisert. Anti-concentration theorems for schemes showing a quantum computational supremacy. *arXiv:1706.03786*, 2017.

- 30 Daniel Harlow. Jerusalem lectures on black holes and quantum information. *Reviews of Modern Physics*, 88(1):015002, 2016.
- 31 Aram Harrow and Saeed Mehraban. Personal communication. 2018.
- 32 Richard Jozsa and Akimasa Miyake. Matchgates and classical simulation of quantum circuits. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 464, pages 3089–3106. The Royal Society, 2008.
- 33 Richard Jozsa and Maarten Van den Nest. Classical simulation complexity of extended Clifford circuits. *Quantum Information and Computation*, 14(7/8):633–648, 2014.
- 34 Dax Enshan Koh. Further extensions of Clifford circuits and their classical simulation complexities. *Quantum Information & Computation*, 17(3&4):0262–0282, 2017.
- 35 Greg Kuperberg. How hard is it to approximate the Jones polynomial? *Theory of Computing*, 11(6):183–219, 2015.
- 36 Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975.
- 37 Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect quantum error correcting code. *Physical Review Letters*, 77(1):198, 1996.
- 38 Easwar Magesan, Jay M Gambetta, and Joseph Emerson. Scalable and robust randomized benchmarking of quantum processes. *Physical Review Letters*, 106(18):180504, 2011.
- 39 Ryan L. Mann and Michael J. Bremner. On the complexity of random quantum computations and the Jones polynomial. *arXiv:1711.00686*, 2017.
- 40 Enrique Martin-Lopez, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Qi Zhou, and Jeremy L O’Brien. Experimental realization of shor’s quantum factoring algorithm using qubit recycling. *Nature Photonics*, 6(11):773–776, 2012.
- 41 Dmitri Maslov and Martin Roetteler. Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations. *arXiv:1705.09176*, 2017.
- 42 Tomoyuki Morimae. Hardness of classically sampling one clean qubit model with constant total variation distance error. *arXiv:1704.03640*, 2017.
- 43 Tomoyuki Morimae, Keisuke Fujii, and Joseph F Fitzsimons. Hardness of classically simulating the one-clean-qubit model. *Physical Review Letters*, 112(13):130502, 2014.
- 44 Gabriele Nebe, Eric M Rains, and Neil JA Sloane. The invariants of the clifford groups. *Designs, Codes and Cryptography*, 24(1):99–122, 2001.
- 45 Gabriele Nebe, Eric M Rains, and Neil James Alexander Sloane. *Self-dual codes and invariant theory*, volume 17. Springer, 2006.
- 46 Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- 47 Michał Oszmaniec and Zoltán Zimborás. Universal extensions of restricted classes of quantum operations. *arXiv:1705.11188*, 2017.
- 48 John Preskill. Quantum computing and the entanglement frontier. *arXiv:1203.5813*, 2012.
- 49 Robert Raussendorf and Hans J Briegel. A one-way quantum computer. *Physical Review Letters*, 86(22):5188, 2001.
- 50 Robert Raussendorf, Daniel E Browne, and Hans J Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2):022312, 2003.
- 51 Imdad SB Sardharwalla, Toby S Cubitt, Aram W Harrow, and Noah Linden. Universal refocusing of systematic quantum noise. *arXiv:1602.07963*, 2016.
- 52 Adam Sawicki and Katarzyna Karnas. Criteria for universality of quantum gates. *Phys. Rev. A*, 95:062303, Jun 2017.
- 53 Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

- 54 Andrew Steane. Multiple-particle interference and quantum error correction. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 452, pages 2551–2577. The Royal Society, 1996.
- 55 Larry Stockmeyer. The complexity of approximate counting. In *Proceedings of the fifteenth annual ACM Symposium on Theory of Computing*, pages 118–126. ACM, 1983.
- 56 Barbara M Terhal and David P DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and Arthur-Merlin games. *Quantum Information & Computation*, 4(2):134–145, 2004.
- 57 Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. doi:10.1137/0220053.
- 58 Zak Webb. The Clifford group forms a unitary 3-design. *Quantum Information and Computation*, 16:1379–1400, 2016.
- 59 Huangjun Zhu. Multiqubit clifford groups are unitary 3-designs. *Physical Review A*, 96(6):062336, 2017.





# Efficient Batch Verification for UP

Omer Reingold<sup>1</sup>

Stanford University  
Palo Alto CA, USA  
reingold@stanford.edu

Guy N. Rothblum

Weizmann Institute  
Rehovot, Israel  
rothblum@alum.mit.edu

Ron D. Rothblum<sup>2</sup>

MIT and Northeastern University  
Cambridge and Boston MA, USA  
ronr@csail.mit.edu

---

## Abstract

Consider a setting in which a prover wants to convince a verifier of the correctness of  $k$  NP statements. For example, the prover wants to convince the verifier that  $k$  given integers  $N_1, \dots, N_k$  are all RSA moduli (i.e., products of equal length primes). Clearly this problem can be solved by simply having the prover send the  $k$  NP witnesses, but this involves a lot of communication. Can interaction help? In particular, is it possible to construct *interactive* proofs for this task whose communication grows sub-linearly with  $k$ ?

Our main result is such an interactive proof for verifying the correctness of any  $k$  UP statements (i.e., NP statements that have a unique witness). The proof-system uses only a constant number of rounds and the communication complexity is  $k^\delta \cdot \text{poly}(m)$ , where  $\delta > 0$  is an arbitrarily small constant,  $m$  is the length of a single witness, and the **poly** term refers to a fixed polynomial that only depends on the language and not on  $\delta$ . The (honest) prover strategy can be implemented in polynomial-time given access to the  $k$  (unique) witnesses.

Our proof leverages “interactive witness verification” (IWV), a new type of proof-system that may be of independent interest. An IWV is a proof-system in which the verifier needs to verify the correctness of an NP statement using: (i) a sublinear number of queries to an alleged NP witness, and (ii) a short interaction with a powerful but untrusted prover. In contrast to the setting of PCPs and Interactive PCPs, here the verifier only has access to the *raw* NP witness, rather than some encoding thereof.

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

**Keywords and phrases** Interactive Proof, Batch Verification, Unique Solution

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.22

**Related Version** A full version of the paper is available at [30], <https://eccc.weizmann.ac.il/report/2018/022>.

---

<sup>1</sup> Supported by NSF grant CCF-1749750.

<sup>2</sup> Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship and in part by the Defense Advanced Research Projects Agency (DARPA), the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236 and by the Cybersecurity and Privacy Institute at Northeastern University.



© Omer Reingold, Guy N. Rothblum,  
and Ron D. Rothblum;  
licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 22; pp. 22:1–22:23



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**Acknowledgements** We thank Oded Goldreich for illuminating conversations and particularly for his insights that helped us crystallize the notion of Interactive Witness Verification.

## 1 Introduction

The power of efficiently verifiable proof-systems is a central question in the study of computation. Interactive proofs, introduced in the seminal work of Goldwasser, Micali and Rackoff [24], are interactive protocols between a randomized verifier and an untrusted prover. The prover convinces the verifier of the validity of a computational statement, usually framed as membership of an input  $x$  in a language  $\mathcal{L}$ . Soundness is unconditional. Namely, if the input is not in the language, then no matter what (unbounded and adaptive) strategy a cheating prover might employ, the verifier should reject with high probability over its own coin tosses. Interactive proofs have had a dramatic impact on complexity theory and on cryptography. Opening the door to randomized and interactive verification led to revolutionary notions of proof verification, such as zero knowledge interactive proofs [24, 21] and probabilistically checkable proofs (PCPs) [7, 15, 5, 4, 13, 3, 1]. Interactive proof-systems also allow for more efficient verification of larger classes of computations (compared with NP proof systems), as demonstrated in the celebrated  $\text{IP} = \text{PSPACE}$  Theorem [29, 33].

This work studies whether interactive proofs can allow for more efficient *batch verification* of NP statements. Namely:

*Can an untrusted prover convince a verifier of the correctness of  $k$  NP statements with communication complexity that is sublinear in  $k$ ?*

Note that the naive solution is sending the  $k$  witnesses in their entirety. Looking ahead, our main result answers this question in the affirmative for a rich subclass of NP (the class UP of NP statements that have at most one witness). Along the way, we also introduce and study a new notion of proof-system: Interactive Witness Verification (IWW), which allow for the verification of NP statements using a sublinear number of queries to a “raw” (unencoded) NP witness and a short interaction with an untrusted prover. We construct IWWs for a rich subclass of NP, and these are a primary ingredient in our efficient batch verification protocol for UP.

Before proceeding to detail these contributions, we observe that the membership of  $k$  inputs in an NP language can be solved in space  $O(\log k + m \cdot \text{poly}(n))$ , where  $n$  is the length of a single input and  $m$  is the length of a single NP witness. Thus, by the  $\text{IP} = \text{PSPACE}$  Theorem, there is an interactive proof for batch verification with communication complexity  $\text{poly}(\log k, n, m)$ . A major caveat, however, is that the complexity of proving correctness (the running time of the *honest* prover) is *exponential in  $\text{poly}(n, m)$* . We, on the other hand, focus on batch verification where the honest prover runs in *polynomial time* given the  $k$  NP witnesses. We refer to such an interactive proof as having an *efficient prover*.<sup>3</sup>

---

<sup>3</sup> Efficiency of the honest prover (given an NP witness) has been central in the study of zero-knowledge interactive proofs [24, 21]. Our emphasis on an efficient honest prover is also inspired by the recent line of work on *doubly-efficient interactive proofs* [23]. That line of work focuses on proofs for deterministic polynomial-time computations and the prover is required to run in polynomial-time without any auxiliary input. We remark that doubly efficient interactive proofs for deterministic computations do not seem to imply protocols for non-deterministic computations such as those we consider here.

## 1.1 Our Results

Our main result is an interactive proof-system, with an efficient prover, for verifying the correctness of  $k$  UP statements. Recall that UP refers to the subclass of NP statements for which correct statements have a *unique* witness. The canonical example (of a promise problem) in this class is **unique-SAT** in which one needs to distinguish unsatisfiable formulas from those having a unique satisfying assignment. Multiple other examples arise from cryptography, where problems related to factoring, discrete-log or lattices all have unique solutions.

Our protocol for UP batch verification uses a constant number of rounds and has communication complexity  $(k^\delta \cdot \text{poly}(m))$ , for any arbitrarily small constant  $\delta > 0$ . For UP relations that are checkable in polynomial-time and bounded polynomial space, we can reduce the communication complexity to  $(k^\delta \cdot m^{1+\delta})$ . When the number of instances  $k$  is large, this is a significant improvement over the trivial solution in which the prover sends over all  $k$  witnesses.

► **Theorem 1** (Informally Stated, see Theorem 13). *Let  $\mathcal{L}$  be a language in UP with witnesses of length  $m$ . For every  $\delta > 0$ , there exists a constant-round interactive proof for verifying that  $k$  instances  $x_1, \dots, x_k$ , each of length  $n$ , all belong to  $\mathcal{L}$ . The communication complexity is  $k^\delta \cdot \text{poly}(m)$ . The verifier runs in time  $\tilde{O}(k \cdot n) + k^\delta \cdot \text{poly}(m)$ , where  $n$  is the length of each of the instances. The honest prover runs in time  $\text{poly}(k, n, m)$  given the  $k$  unique witnesses.*

**Comparison to prior work.** Theorem 1 improves over the aforementioned protocol derived from IP = PSPACE theorem in two ways: (1) it has an efficient prover strategy (given the witnesses), and (2) it uses only a constant number of rounds of interaction (whereas the IP = PSPACE theorem uses  $\text{poly}(\log k, n, m)$  rounds).

Theorem 1 also improves over a prior result for batch verification of UP statements [31]. The communication complexity of that protocol has an additional additive  $k \cdot \text{polylog}(m)$  term. In particular, even when  $k$  is larger than  $m$ , the [31] protocol gives at most a quadratic saving over just naively sending all  $k$  witnesses. In contrast, our protocol yields an arbitrarily large polynomial saving in the parameter  $k$ .<sup>4</sup> A comparison of our techniques with those of [31] is provided in Section 1.2.1.

**A new type of proof-system.** The protocol of Theorem 1 makes extensive use of a new type of proof-system that we introduce and construct. In a nutshell, these are proof-systems in which the verifier needs to check the correctness of an NP statement given oracle access to the NP witness. In contrast to PCPs, the verifier is only given access to the *raw* (i.e., original) NP witness, rather than to an encoding thereof. We allow the verifier to have a short interaction with an all powerful, but untrusted prover (this part of the interaction is similar to the interactive PCPs of Kalai and Raz [27]). We use the name “interactive witness verification” (IWW) for these proof-systems.

Jumping ahead, we remark that IWWs are closely related to *interactive proofs of proximity* (IPPs) [12, 32]. Indeed, we show that IPPs for a class of deterministic polynomial-time computations directly imply IWWs for a related class of NP relations. The IWW protocol used in the proof of Theorem 1 is derived from known results on IPPs [32]. We proceed to elaborate on the notion of IWWs.

<sup>4</sup> We remark that a linear dependence of the communication complexity on  $m$  is inherent. Indeed, by results of Goldreich *et al.* [20, 22], under complexity theoretic assumptions, even verification of a single instance (i.e.,  $k = 1$ ) requires  $\Omega(m)$  communication.

### 1.1.1 Interactive Witness Verification

**Motivation: sublinear witness verification.** Suppose Alice wants to test whether a given graph  $G$  is 3-colorable. The validity of an alleged 3-coloring  $\chi$  can be verified in polynomial time, but this requires reading every bit of the coloring. Can Alice verify  $\chi$ 's validity while reading a *sublinear* number of bits from  $\chi$ ?

At first glance, it may seem that PCPs give a direct solution to this problem. Recall that a PCP is an encoding of an NP witness that can be verified by reading only a very small number of bits. The celebrated PCP theorem [2] shows that every NP language has a PCP proof-system. However, the reason that PCPs do *not* solve Alice's problem is that she only has oracle access to the *original* 3-coloring  $\chi$  of the graph. In contrast, in the PCP setting, the verifier is given oracle access to an *encoding* of the witness (e.g., via an error correcting code).

Indeed, sublinear witness verification for general NP relations is not possible (assuming that  $P \neq NP$ ). Consider an NP relation for satisfiability where the witness is an encoding of the  $m$ -bit satisfying assignment  $w$  on a polynomial of high degree (say degree  $10m$ ). The polynomial is given by a list of valuations on field elements and the satisfying assignment can be recovered by interpolating the polynomial. There are many possible polynomials that encode a given satisfying assignment. In particular, if Alice is given a *random* polynomial encoding the assignment, she must read  $\Omega(m)$  valuations before she learns anything about the alleged satisfying assignment.

**Adding interaction.** While sublinear witness verification for general NP relations is not possible, this situation changes when we also allow *interaction*. Namely, we allow Alice to interact with an untrusted prover Bob who knows the entire 3-coloring of the graph. The communication should also be sublinear in the witness size (in particular, Bob cannot simply send  $\chi$  to Alice). Given the 3-coloring, an honest Bob should run in polynomial time.

More generally, an *interactive witness verification* for a given NP relation  $R$  for a language  $\mathcal{L}$  is a protocol between a prover  $\mathcal{P}$  and verifier  $\mathcal{V}$ , who both get as input an instance  $x$ . In addition, the verifier has query access to an alleged witness  $w$  for  $x$  and the prover is given full explicit access to  $w$ . The prover wants to convince the verifier that indeed  $(x, w) \in R$ . Towards this end, the prover and verifier run an interactive protocol. The verifier  $\mathcal{V}$ , on examination of the communication transcript with  $\mathcal{P}$  and the queried points in  $w$ , accepts or rejects the prover's claim. For completeness, if  $(x, w) \in R$  then the verifier  $\mathcal{V}$ , when interacting with the honest prover  $\mathcal{P}$ , should accept. For soundness, we emphasize that *the witness  $w$  is fixed before the protocol begins*. I.e., the soundness property is that if  $x \notin \mathcal{L}$ , then for any *fixed* false witness  $w^*$ , and for any (unbounded) cheating prover strategy  $\mathcal{P}^*$ , if we run the interactive protocol between  $\mathcal{V}$  and  $\mathcal{P}^*$ , where  $\mathcal{V}$ 's witness-queries are answered using the fixed (false) witness  $w^*$ , then  $\mathcal{V}$  will reject w.h.p. over its coins tosses. In terms of complexity, our goal is to have both the number of witness-queries and the communication be significantly smaller than  $m = |w|$ .

Note that an IWV refers to a specific NP *relation*, and not only to the underlying NP language (which can have many possible NP relations). Indeed, every NP language has *some* witness relation with an extremely efficient IWV.<sup>5</sup> The point, however, is that we would like to construct IWVs for arbitrary NP relations, not just highly structured ones.

---

<sup>5</sup> Consider the relation in which witnesses are PCP proof strings. For such relations even the interaction with the prover is unnecessary.

In particular, an IWV can be particularly useful in situations where the NP witness is outside the prover's control. For example, the witness could arise from nature in the form of a physical or biological observation, in which case it is not reasonable to assume that it is given in an encoded format. Thus, we find IWVs to be natural objects worthy of further study (beyond their importance as a technical tool in our protocol for UP batch verification).

**Constructions.** We construct IWVs for a large class of NP relations. Namely, any NP relation  $R$  that is checkable by bounded-space Turing machines or bounded-depth (logspace uniform) circuits. This includes, for example, the natural NP relations for 3-coloring, satisfiability,  $k$ -clique, etc. In the protocol the verifier only reads  $\sqrt{m}$  bits of the witness and the communication complexity is also roughly  $\sqrt{m}$  (recall that  $m$  denotes the witness length). Thus, the verifier only observes  $\sqrt{m}$  bits of information about the witness. More precisely, we obtain the following result, which allows for a tradeoff between the query and communication complexities:

► **Theorem 2** (Informally Stated, see Theorem 9). *Let  $q$  be a parameter. Let  $R$  be an NP relation with witness length  $m$  that is checkable either by a bounded space Turing Machine or by a bounded depth (logspace uniform) circuit. For every constant  $\delta > 0$ , and  $q \in [m]$ , there exists an IWV for  $R$  in which the verifier reads  $q$  bits of the witness and the communication complexity is  $O(m^{1+\delta}/q)$ . Furthermore, the prover, given the NP witness, can be implemented efficiently (i.e., in polynomial time). The verifier runs in time  $\tilde{O}(n + q + (m^{1+\delta}/q))$ . For bounded space relations the number of rounds is constant, and for bounded depth relations it is  $\text{polylog}(n)$ .*

The proof of Theorem 2 relies on known constructions of *interactive proofs of proximity* (IPPs). Loosely speaking, IPPs are interactive proofs in which the verifier runs in time that is sub-linear in the input length and is convinced that the input is *close* to the language (see Section 1.3 for additional details and related works on IPPs). Specifically, Theorem 2 follows from an IPP construction of Rothblum, Vadhan and Wigderson [32] (together with an extension due to [31]).

In particular, IWVs are closely related to IPPs. To see this, observe that an IWV for a relation  $R$  may be thought of as an IPP, where the IPP input is the IWV witness  $w$ , and the goal is to check whether the witness is “close” to the language  $\mathcal{L}_x = \{w' : (x, w') \in R\}$ . Even though IWVs do not refer to the proximity of the witness to  $\mathcal{L}_x$ , observe that if  $x$  is not in the language, then  $\mathcal{L}_x = \emptyset$ , and so any fixed  $w$  will have “infinite” distance from  $\mathcal{L}_x$ . In general, IWVs seem to be a more relaxed object than IPPs (in particular, the above reduction gives instances with infinite distance).

**Lower Bound for IWVs.** We also give a lower bound for IWVs that shows that Theorem 2 is quantitatively almost tight. Moreover, the specific NP relation for which we demonstrate this lower bound is in the class of relations for which we assume an IWV in the proof of Theorem 1. For our lower bound to hold we assume the existence of an exponentially strong cryptographic pseudorandom generator. The proof of the lower bound follows from a similar lower bound of Kalai and Rothblum [28] for IPPs, but is somewhat simpler.

## 1.2 Technical Overview

Let  $\mathcal{L}$  be a UP language and let  $R$  be the corresponding UP relation. By the proof of the Cook-Levin theorem we may assume without loss of generality that  $R$  is computable in  $\text{NC}^1$

(more specifically, by a CNF formula).<sup>6</sup> Recall that our goal is to design a protocol, between a prover  $\mathcal{P}_{\text{batch}}$  and verifier  $\mathcal{V}_{\text{batch}}$  that both get as input  $k$  instances  $x_1, \dots, x_k$ . The prover, in addition, also gets witnesses  $w_1, \dots, w_k$ , each of length  $m$ , and needs to convince  $\mathcal{V}$  that  $x_1, \dots, x_k \in \mathcal{L}$ .

For sake of simplicity, we will focus on constructing a protocol with small communication  $O(k^\delta \cdot m^{1+\delta})$  for some small constant  $\delta > 0$ , and ignore the running time of the verifier. Obtaining a verifier that runs in time that is sub-linear in  $k$  amounts to maintaining concise descriptions of all the objects involved in the interaction. We ignore the verification time for this overview.<sup>7</sup>

**A Warmup: Batch Verification with  $\sqrt{k} \cdot m^{1+\delta}$  Communication.** As a warmup, we first consider a quantitatively easier task: batch verification with communication complexity  $\sqrt{k} \cdot m^{1+\delta}$  (rather than our eventual goal of  $k^\delta \cdot m^{1+\delta}$ ). This task is already non-trivial and demonstrates most of our key ideas.

Consider an augmented UP relation  $R^{\otimes k}$  defined as:

$$R^{\otimes k} = \left\{ ((x_1, \dots, x_k), (w_1, \dots, w_k)) : \forall j \in [k], (x_j, w_j) \in R \right\}.$$

Note that  $R^{\otimes k}$  is computable in  $\text{NC}^1$ . By Theorem 2,  $R^{\otimes k}$  has an efficient IWV protocol. Setting the parameter  $q$  of Theorem 2 to  $\sqrt{k}$ , we obtain an IWV protocol for  $R^{\otimes k}$  in which the verifier reads  $\sqrt{k}$  bits of the witness  $\mathbf{w} = (w_1, \dots, w_k)$  and with communication roughly  $\sqrt{k} \cdot m^{1+\delta}$  (where recall that  $m = |w_1| = \dots = |w_k|$  is the length of a single witness).

We would like for  $\mathcal{P}_{\text{batch}}$  and  $\mathcal{V}_{\text{batch}}$  to run this IWV. An immediate objection that should arise is that in contrast to the IWV setting, we are now trying to construct a standard interactive proof and so the verifier does not have access to the witness string  $\mathbf{w} = (w_1, \dots, w_k)$ . To get around this, we will leverage a property of the IWV of Theorem 2. Specifically, that IWV operates in two phases: an online phase followed by an offline phase. First, in the online phase, the verifier does *not* have oracle access to the witness  $w$ , but is allowed to communicate with the prover. The result of this interaction is a set of coordinates  $Q$  of  $w$  that the verifier would like to read and a predicate  $\phi$  to be applied to  $w_Q$ . In the second (offline) phase, the verifier is given oracle access to  $w$  but is no longer allowed to interact with the prover. Rather, the verifier just reads  $w_Q$  and accepts if and only if  $\phi(w_Q) = 1$ . The soundness condition remains unchanged. Namely, for any alleged witness  $w$  for a false statement, which is fixed prior to the interaction, with high probability over the coins tossed in the online phase, either the verifier rejects or it generates  $Q$  and  $\phi$  such that  $\phi(w_Q) = 0$ .

The batch verification prover  $\mathcal{P}_{\text{batch}}$  and verifier  $\mathcal{V}_{\text{batch}}$  start by running the online phase of the IWV for  $R^{\otimes k}$  on input  $(x_1, \dots, x_k)$ . The prover  $\mathcal{P}_{\text{batch}}$  also uses  $\mathbf{w} = (w_1, \dots, w_k)$  as its witness string. Observe that for this online phase the verifier  $\mathcal{V}_{\text{batch}}$  does not need oracle access to  $\mathbf{w}$  (indeed, as discussed above,  $\mathcal{V}_{\text{batch}}$  has no such oracle access).

<sup>6</sup> This step incurs a polynomial blowup in the witness size, which is the source of the  $\text{poly}(m)$  dependence in Theorem 1. This blowup can be avoided for many natural UP relations which are natively checkable in  $\text{NC}^1$  and more generally, for relations that are checkable by bounded space Turing machines or bounded depth (logspace uniform) circuits.

<sup>7</sup> One way getting an efficient verifier is to first construct a protocol with small communication but large verification time (as will be described in this overview), and then further delegate the verification task (which is a deterministic computation applied to the transcript of the interaction and the input) to the prover using a doubly-efficient interactive proof such as those of [23] or [31]. However, in our actual construction we show that the verifier can be implemented efficiently directly, without this additional trick.



At the end of this phase,  $\mathcal{V}_{\text{batch}}$  obtains a set  $Q \subseteq [k] \times [m]$ , of size  $|Q| \leq \sqrt{k}$ , of points that it would like to read from  $\mathbf{w} = (w_1, \dots, w_k)$  together with a predicate  $\phi : \{0, 1\}^{|Q|} \rightarrow \{0, 1\}$  to be evaluated on  $\mathbf{w}_Q$ . However, since  $\mathcal{V}_{\text{batch}}$  does not have oracle access to  $\mathbf{w}$ , it is not immediately clear how we can leverage the soundness condition of IWWs in our current setting. Jumping ahead, we shall do so by using the uniqueness of the witnesses in a crucial way.

Specifically, consider the fixed witness string  $\hat{\mathbf{w}} = (\hat{w}_1, \dots, \hat{w}_1)$ , where for every  $j \in [k]$ : if  $x_j \in \mathcal{L}$ , then we set  $\hat{w}_j$  to be the corresponding *unique* witness, and otherwise (in case  $x_j \notin \mathcal{L}$ ) we set  $\hat{w}_j$  as an arbitrary fixed string (e.g.,  $0^m$ ). Since  $\hat{\mathbf{w}}$  is an a priori fixed string, by the soundness of the IWW, if there exists some  $j^*$  such that  $x_{j^*} \notin \mathcal{L}$ , then, no matter what a cheating prover does, with high probability the IWW verifier generates  $\phi$  and  $Q$  such that  $\phi(\hat{\mathbf{w}}_Q) = 0$  (or it rejects, which case  $\mathcal{V}_{\text{batch}}$  also rejects).

Suppose that indeed  $\phi(\hat{\mathbf{w}}_Q) = 0$ . While our verifier  $\mathcal{V}_{\text{batch}}$  does not have oracle access to  $\hat{\mathbf{w}}$ , since  $|Q| \leq \sqrt{k}$ , we can ask the prover  $\mathcal{P}_{\text{batch}}$  to send all of the witnesses belonging to the subset  $S \subseteq [k]$  of witnesses that are “touched” by  $Q$  (i.e.,  $S$  is the projection of the set  $Q \subseteq [k] \times [m]$  to its first coordinate). Denote the set of alleged witnesses that the prover sends by  $\tilde{\mathbf{w}} = (\tilde{w}_j)_{j \in S}$ . Sending  $\tilde{\mathbf{w}}$  only costs us an additional  $|S| \cdot m = O(\sqrt{k} \cdot m)$  communication. The verifier  $\mathcal{V}_{\text{batch}}$  checks that  $(x_j, \tilde{w}_j) \in R$ , for all  $j \in S$ , and that  $\phi(\tilde{\mathbf{w}}) = 1$ . To argue that soundness holds, observe that if for some  $j \in S$  it holds that  $x_j \notin \mathcal{L}$ , then for any  $\tilde{w}_j$  that a potential cheating prover  $\mathcal{P}_{\text{batch}}^*$  might send, it holds that  $(x_j, \tilde{w}_j) \notin \mathcal{R}$  and so  $\mathcal{V}_{\text{batch}}$  rejects. On the other hand, since  $\mathcal{L} \in \text{UP}$ , if  $x_j \in \mathcal{L}$  for all  $j \in S$  (which can certainly happen), the prover  $\mathcal{P}_{\text{batch}}^*$  has to send the *unique* witnesses  $\tilde{\mathbf{w}} = \hat{\mathbf{w}}_S$  (in order for these witnesses to satisfy the relation  $R$ ) in which case  $\mathcal{V}_{\text{batch}}$  rejects when checking that  $\phi(\tilde{\mathbf{w}}) = \phi(\hat{\mathbf{w}}_S) = 1$ . Thus, in any case  $\mathcal{V}_{\text{batch}}$  rejects and soundness follows.

Taking a step back, our proof of soundness strongly leverages the uniqueness of the witnesses to emulate query access to an a priori fixed witness by having the prover send the relevant parts of the witness a posteriori (i.e., after the interactive phase of the IWW).

Observe that by our setting of parameters, the IWW part of the protocol uses  $\sqrt{k} \cdot m^{1+\delta}$  communication, and actually sending the witnesses  $(\hat{w}_j)_{j \in S}$  adds only an additional  $\sqrt{k} \cdot m$  communication. Overall we obtain communication complexity  $\sqrt{k} \cdot m^{1+\delta}$ .

**Batch Verification with  $k^\delta \cdot m$  Communication.** The main idea for obtaining smaller communication  $k^\delta \cdot m^{1+\delta}$ , where  $\delta > 0$  is an arbitrary small constant, is to recursively apply the solution for the warmup case (with slightly different parameters). We describe our approach in detail next.

First, in contrast to the warmup case, we use the IWW of Theorem 2 with respect to parameter  $q = k^{1-\delta}$  (rather than  $q = \sqrt{k}$ ). Thus, we have an IWW for the relation  $R_k$  with communication  $k^\delta \cdot m^{1+\delta}$  and query complexity  $k^{1-\delta}$ .

The prover  $\mathcal{P}_{\text{batch}}$  and  $\mathcal{V}_{\text{batch}}$  run the IWW as in the warmup. The main difference is that now the set  $S$  of instances that are queried in the IWW is of size  $k^{1-\delta}$  and we cannot afford for  $\mathcal{P}_{\text{batch}}$  to send  $(w_j)_{j \in S}$  explicitly to  $\mathcal{V}_{\text{batch}}$ . Rather, we observe that after running the (online part of the) IWW what remains to be checked is that for every  $j \in S$  it holds that  $x_j \in \mathcal{L}$  and that the corresponding (unique) witnesses  $(w_j)_{j \in S}$  satisfy  $\phi((w_j)_{j \in S})$ .

Our approach is to check that these conditions hold by a recursive application of our batch verification protocol on  $(x_j)_{j \in S}$ . Observe that the number of instances has shrunk from  $k$  to  $k^{1-\delta}$  so we have made significant progress. Actually, batch verification per se does not suffice since it only guarantees that  $x_j \in \mathcal{L}$  for every  $j \in S$  but does not guarantee that  $\phi((w_j)_{j \in S})$ . Still, we can obtain also the latter condition by using the fact that  $\phi$  is computable in  $\text{NC}_1$  and therefore can be incorporated into the augmented relation which is defined for the next round.

Thus, in each iteration of the recursion we shrink the number of instances by a  $k^\delta$  factor. After  $\ell = O(1/\delta)$  iterations we will be left with a constant number of instances and the prover  $\mathcal{P}^{\text{batch}}$  can send the corresponding witnesses explicitly. As in the warmup case, the verifier checks that all these witnesses satisfy the base relation  $R$  individually and that they jointly satisfy the predicate  $\phi_\ell$  generated by the last iteration of the recursion.

By our setting of parameter for the IWV, the communication complexity in each one of the iterations is  $k^\delta \cdot m^{1+\delta}$ . At the final step the verifier sends  $O(1)$  witnesses in the clear so that also adds at most  $O(m)$  communication. The prover can be implemented efficiently (given the UP witnesses) since the underlying IWV has an efficient prover strategy. As described above, the verifier’s running time might be as high as  $\Omega(k \cdot m)$  (naively, that will be complexity of the first iteration) but in the technical sections we provide a more refined analysis that shows how to implement the verifier more efficiently.

### 1.2.1 Technical Comparison with [31]

As noted above, a less efficient UP batch verification protocol was presented in [31]. We briefly review that approach and the differences from the current work. We refer the reader to [31] and to Goldreich’s recent survey [18] for more details on the [31] protocol.

The high level idea in the [31] batch verification protocol is for the prover to generate PCP proof strings for all the  $k$  instances and send a short “checksum” of these PCPs. The verifier in turn, generates PCP queries and asks the prover to reveal the answer to these queries for all  $k$  PCPs. The checksum is designed to guarantee that a cheating prover must either answer these queries in a way that is consistent with predetermined PCPs, or alternatively, it can send answers that are inconsistent with the correct PCPs of *many* of the statements. For UP the correct PCPs are unique, and so the latter situation can be checked directly by having the verifier specify a random subset of the PCP proofs for the prover to fully reveal (also here, similarly to our protocol, recursion can be used to obtain improved parameters).

The current work completely avoids the use of checksums and PCPs. Instead, we use IWVs to force a cheating prover to make false claim about a subset of the witnesses. IWVs, together with the uniqueness of the witnesses, let us accomplish this without requiring the verifier to make explicit queries to the witnesses. In terms of complexity, the key difference is that in the [31] protocol, the prover must reveal the PCP answers for all the  $k$  instances. This step adds  $\Omega(k)$  to the communication complexity, which we avoid.

## 1.3 Additional Related Works

**Doubly Efficient Interactive Proofs.** Goldwasser, Kalai and Rothblum [23] introduced a variant of interactive proofs, called *doubly-efficient* interactive proofs, in which both the prover and the verifier are highly efficient. Namely, the honest prover must run in time that is proportional to the complexity of the statement being proved, whereas the verifier should be much faster than the complexity of the computation. Soundness is required to hold against computationally unbounded provers.

Goldwasser *et al.* [23] construct such doubly efficient interactive proofs for every language in (logspace-uniform) NC. The aforementioned recent work of Reingold *et al.* [31] gives such proof-systems for languages computable in polynomial-time and some bounded polynomial space. This includes in particular the complexity class SC. Moreover the latter protocol only requires a constant number of rounds of interaction. We note that batch verification of certain types of interactive proofs (that generalize UP batch verification) was a key ingredient in the [31] result.



We remark that [23] and [31] doubly-efficient interactive proof-systems are for *deterministic* computations and do not seem to immediately imply a protocol for batch NP, or even UP, verification.

**Batch Verification with Computational Soundness.** A recent work of Brakerski, Holmgren and Kalai [10] shows an *argument-system* for batch verification of NP statements. We emphasize that they only obtain soundness against polynomial-time cheating provers whereas we achieve statistical soundness against computationally unbounded provers. In addition, the result of [10] relies on an unproven cryptographic assumption (specifically a computational private information retrieval scheme) whereas our result is unconditional. On the other hand, the protocol of [10] also offers significant advantages (some of which are likely to be infeasible in the context of interactive proofs with statistical soundness). First, their result holds for any NP language (rather than just UP). Second, their protocol requires only 2 messages whereas our protocol requires a larger (but still constant) number of rounds. Lastly, under strong enough assumptions, [10] achieve communication that depends only poly-logarithmically on  $k$ , whereas our dependence is any arbitrarily small polynomial.

**Interactive PCPs and PCIPs.** Interactive PCPs, introduced by Kalai and Raz [27] are a generalization of PCPs in which the PCP verifier can, in addition to reading bits of the PCP, also interact with a prover. Our notion of IWV can be thought of as a restriction of Interactive PCPs in which the PCP proof string is exactly the NP witness and cannot be further encoded.

A generalization of interactive PCPs, called probabilistically checkable interactive proofs (PCIPs) was recently introduced by Reingold *et al.* [31] and Ben Sasson *et al.* [8].<sup>8</sup> Loosely speaking, these are interactive proofs in which the verifier only reads few bits of each message. Again, and in contrast to IWVs, the prover is allowed to send long messages of its choice (e.g., a PCP encoding of the NP witness).

**Interactive Proofs of Proximity (IPPs).** As mentioned above, interactive proofs of proximity (IPPs) form an important component in our UP batch verification protocol. IPPs were introduced in [12, 32] and have seen a considerable amount of progress in recent years. The latter work gives a general purpose sub-linear IPP protocol for general bounded depth computations (which was extended to bounded space computations in [31]). A non-interactive variant of IPPs was studied in [25, 14]. Highly efficient protocols for restricted classes of computations (i.e., context free languages and small read-once branching programs) were given in [19]. A study of a computational variant of IPPs was initiated in [28]. The latter work also shows a lower bound for IPPs. As mentioned above, we use their proof technique to derive a similar lower bound for IWVs. Gur and Rothblum [26] show a round hierarchy for IPPs. Most recently, IPPs were considered in the context of zero-knowledge [9] and distribution testing [11].

## 1.4 Open Questions

We conclude the introduction by mentioning two open questions on batch verification:

1. Do there exist interactive proofs for batch verification of arbitrary NP statements (rather than just UP statements)? Ideally such a proof-system would be both constant-round and have an efficient prover strategy, but even getting a result satisfying only one of these

---

<sup>8</sup> Ben Sasson *et al.* refer to these as Interactive Oracle Proofs.

two requirements would be very interesting. We mention that we do not know a way to extend our UP batch verification to NP via the Valiant-Vazirani randomized reduction from NP to UP [34].

2. The communication complexity in our protocol is proportional to  $k^\delta$ . Is it possible to obtain a similar result with communication that grows only poly logarithmically with  $k$ ? (By [6], such a result would likely require a super constant number of rounds.) In particular, a quantitative improvement to the interactive proof of proximity of [32] could yield such a result.<sup>9</sup>

## 1.5 Organization

Section 2 contains definitions and notations. In Section 3 we formally define our notion of interactive witness verification IWV, show that they exist for bounded NP relations and demonstrate a lower bound on their complexity. In Section 4 we present our protocol for batch verification of UP statements.

## 2 Preliminaries

Throughout this work we use  $\text{NC}^1$  to refer to the class of logspace uniform Boolean circuits of logarithmic depth and constant fan-in. Namely,  $\mathcal{L} \in \text{NC}^1$  if there exists a logspace Turing machine  $M$  that on input  $1^n$  outputs a full description of a logarithmic depth circuit  $C : \{0, 1\}^n \rightarrow \{0, 1\}$  such that for every  $x \in \{0, 1\}^n$  it holds that  $C(x) = 1$  if and only if  $x \in \mathcal{L}$ . We recall that the class SC refers to languages decidable by Turing machines that run in polynomial-time and poly-logarithmic space.

We next define a notion of succinct representation of circuits. Loosely speaking, a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  has a succinct representation if there is a short string  $\langle f \rangle$ , of poly-logarithmic length, that describes  $f$ . That is,  $\langle f \rangle$  can be expanded to a full description of  $f$ . The actual technical definition is slightly more involved and in particular requires that the full description of  $f$  be an  $\text{NC}_1$  (i.e., logarithmic depth) circuit:

► **Definition 3** (Succinct Description of Functions). We say that a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  of size  $s$  has a *succinct description* if there exists a string  $\langle f \rangle$  of length  $\text{polylog}(n)$  and a logspace Turing machine  $M$  (of constant size, independent of  $n$ ) such that on input  $1^n$ , the machine  $M$  outputs a full description of an  $\text{NC}^1$  circuit  $C$  such that for every  $x \in \{0, 1\}^n$  it holds that  $C(\langle f \rangle, x) = f(x)$ . We refer to  $\langle f \rangle$  as the succinct description of  $f$ .

We also define succinct representation for sets  $S \subseteq [k]$ . Roughly speaking this means that the set can be described by a string of length  $\text{polylog}(k)$ . The formal definition is somewhat more involved:

► **Definition 4** (Succinct Description of Sets). We say that a set  $S \subseteq [k]$  of size  $s$  has a succinct description if there exists a string  $\langle S \rangle$  of length  $\text{polylog}(k)$  and a logspace Turing machine  $M$  such that on input  $1^k$ , the machine  $M$  outputs a full description of a depth  $\text{polylog}(k)$  and size  $\text{poly}(s, \log k)$  circuit (of constant fan-in) that on input  $\langle S \rangle$  outputs all the elements of  $S$  as a list (of length  $s \cdot \log(k)$ ).

We emphasize that the size of the circuit that  $M$  outputs is proportional to the actual size of the set  $S$ , rather than the universe size  $k$ .

<sup>9</sup> By slightly adjusting our parameters we could obtain a batch verification protocol with communication complexity  $k^{o(1)} \cdot m^{1+o(1)}$  (and a super constant number of rounds of interaction). Still, achieving communication  $\text{polylog}(k) \cdot m$  seems beyond our current techniques.

## 2.1 Interactive Proofs

An interactive proof-system, as defined by Goldwasser, Micali and Rackoff [24], is a protocol between a polynomial-time verifier  $\mathcal{V}$  and a computationally unbounded prover  $\mathcal{P}$ . The two parties interact and at the end of the interaction the verifier accepts if and only if the given computational statement is correct (with high probability).

We denote by  $(\mathcal{P}, \mathcal{V})(x)$  the output of  $\mathcal{V}$  after interacting with  $\mathcal{P}$  on common input  $x$ . If either  $\mathcal{V}$  or  $\mathcal{P}$  are given additional explicit inputs than we shall denote this by  $(\mathcal{P}(y), \mathcal{V}(z))(x)$  which refers to the output of  $\mathcal{V}$  after interacting with  $\mathcal{P}$  where  $\mathcal{V}$  gets as input  $(x, z)$  and  $\mathcal{P}$  gets as input  $(x, y)$ . We extend the foregoing notation to implicit access to the input by placing the implicit input as a superscript. Thus, by  $(\mathcal{P}, \mathcal{V}^z)(x)$  we refer to the output of  $\mathcal{V}$  after interacting with  $\mathcal{P}$ , where  $\mathcal{V}$  has oracle access to  $z$  and both parties have explicit access to  $x$ .

► **Definition 5.** An *interactive proof* for a language  $\mathcal{L}$  is an interactive protocol between a polynomial-time verifier  $\mathcal{V}$  and a computationally unbounded prover  $\mathcal{P}$ . Both parties are given as input a string  $x$  and must satisfy the following two properties:

- **Completeness:** If  $x \in \mathcal{L}$ , then

$$\Pr[(\mathcal{P}, \mathcal{V})(x) = 1] = 1.$$

- **Soundness:** If  $x \notin \mathcal{L}$ , then for every possible cheating strategy  $\mathcal{P}^*$ ,

$$\Pr[(\mathcal{P}^*, \mathcal{V})(x) = 1] \leq 1/2.$$

If  $\mathcal{L} \in \text{NP}$ , we say that an interactive proof for  $\mathcal{L}$  has an *efficient prover* if the honest prover strategy  $\mathcal{P}$  can be implemented in polynomial-time given access to an NP witness.

► **Remark (On Completeness and Soundness Errors).** We note that Theorem 5 can be generalized to allow for an error in the completeness condition. For simplicity however, and since our protocols achieve perfect completeness, we avoid doing so.<sup>10</sup> We also remark that the soundness error (and completeness error, if one allows for such) can be reduced at an exponential rate by either sequential or parallel repetition (see, e.g., [17, Lemma C.1]).

### 2.1.1 Doubly Efficient Interactive Proofs

Doubly-efficient interactive proofs, introduced by Goldwasser *et al.* [23] are interactive proofs in which the prover is relatively efficient (i.e., runs in time proportional to the complexity of the computation), whereas the verifier is extremely efficient (i.e., running in almost linear time). We shall use a recent result of Reingold *et al.* [31] giving such doubly efficient interactive proofs for bounded space computations. The reason that we use the more recent protocol of [31] rather than that of [23], is mainly because the former only requires a constant number of rounds.

► **Theorem 6 (Interactive Proofs for Bounded Space ([31])).** *Let  $T = T(n)$  and  $S = S(n)$  such that  $n \leq T \leq \exp(n)$  and  $\log(T) \leq S \leq \text{poly}(n)$ .*

*Let  $\mathcal{L} \in \text{DTISP}(T, S)$  and let  $\tau = \tau(n) \in (0, 1/2)$  such that  $\text{poly}(1/\tau) \leq \log(T)$ . Then,  $\mathcal{L}$  has a public-coin interactive proof with perfect completeness and soundness error  $\frac{1}{2}$ .*

<sup>10</sup>Fürer *et al.* [16] show how to transform any interactive proof to one having perfect completeness. Their transformation however does not preserve the efficiency of the prover.

The number of rounds is  $(1/\tau)^{O(1/\tau)}$ . The communication complexity is  $T^{O(\tau)} \cdot \text{poly}(S)$ . The (prescribed) prover runs in time  $T^{1+O(\tau)} \cdot \text{poly}(S)$  time, and the verifier runs in time  $(n \cdot \text{polylog}(T) + T^{O(\tau)} \cdot \text{poly}(S))$ .

### 3 Interactive Witness Verification

For an NP relation  $R$ , we denote by  $R(x)$  the set of witnesses for  $x$ , namely  $R(x) = \{w : R(x, w) = 1\}$ .

► **Definition 7** (Interactive Witness Verification IWV). An *Interactive Witness Verification Protocol* (IWV) for an NP relation  $R$ , is an interactive protocol between a computationally unbounded prover  $\mathcal{P}$  and a verifier  $\mathcal{V}$  on a given input  $x$ . The verifier also has *oracle* access to an alleged witness  $w$ . The protocol must satisfy the following two requirements:

- **Completeness:** If  $(x, w) \in R$  then

$$\Pr \left[ (\mathcal{P}, \mathcal{V}^w)(x, |w|) = 1 \right] = 1.$$

- **Soundness:** If  $R(x) = \emptyset$  (i.e.,  $x$  is not in the underlying NP language), then for every a priori fixed  $w^*$  and every prover strategy  $\mathcal{P}^*$ :

$$\Pr \left[ (\mathcal{P}^*, \mathcal{V}^{w^*})(x, |w^*|) = 1 \right] \leq 1/2.$$

The *query complexity*  $q = q(|x|, |w|)$  is the number of bits that  $\mathcal{V}$  reads from  $w$  and the *communication complexity*  $\text{cc} = \text{cc}(|x|, |w|)$  is the number of bits exchanged between  $\mathcal{V}$  and  $\mathcal{P}$  in the protocol.

We say that the IWV has an *efficient prover*, if the honest prover strategy  $\mathcal{P}$  can be implemented in polynomial time, if the prover is given explicit access to  $w$  (i.e., the same witness to which the verifier has oracle access).

Loosely speaking, we say that an IWV is *oblivious* if the verifier makes all its queries non-adaptively at the end of the interaction. Put differently, at the end of the interaction the verifier specifies some query set  $Q$  of bits from the witness, and a predicate  $\phi$  and accepts if and only if  $\phi(w_Q) = 1$ . For technical considerations in our proof, we actually require that the verifier generate succinct descriptions of  $Q$  and  $\phi$  (see Theorems 3 and 4 for the precise technical definition of succinct descriptions of functions and sets). This allows the verifier to run in time that is sublinear in the sizes of  $Q$  and  $\phi$ . We proceed to the formal definition:

► **Definition 8** (Oblivious IWV). An *Oblivious IWV* for an NP relation  $R$  with witness length  $m$ , is an interactive protocol between a computationally unbounded prover  $\mathcal{P}$  and a verifier  $\mathcal{V}$  on a given input  $x$ . At the end of the interaction either the verifier rejects or it outputs a succinct description  $\langle Q \rangle$  of a set  $Q \subseteq [m]$  of size  $q$  and succinct description  $\langle \phi \rangle$  of a predicate  $\phi : \{0, 1\}^q \rightarrow \{0, 1\}$  such that

- **Completeness:** If  $(x, w) \in R$  then

$$\Pr \left[ \mathcal{V} \text{ does not reject and } \phi(w_Q) = 1 \right] = 1.$$

- **Soundness:** If  $R(x) = \emptyset$  (i.e.,  $x$  is not in the underlying NP language), then for every a priori fixed  $w^*$  and every prover strategy  $\mathcal{P}^*$ :

$$\Pr \left[ \mathcal{V} \text{ does not reject and } \phi(w_Q^*) = 1 \right] \leq 1/2.$$

The *query complexity* of an oblivious IWV is  $q$ , the size of the set  $Q$ , and the *communication complexity*  $cc$  is the number of bits exchanged between  $\mathcal{V}$  and  $\mathcal{P}$  in the protocol.

We say that the IWV has an *efficient prover*, if the honest prover strategy  $\mathcal{P}$  can be implemented in polynomial time, if the prover is given explicit access to  $w$  (i.e., the same witness to which the verifier has oracle access to).

We will rely on the following theorem, which is established in Section 3.1. Loosely speaking, this result shows that any NP relation verifiable by small depth circuits has an IWV in which we can trade off the query and communication complexities, such that their product is roughly equal to the witness length. In particular, it yields IWV protocols for all such NP relations in which the complexity scales with the square root of the witness length.

► **Theorem 9.** [*IWVs for Bounded Space Relations*] *Let  $R$  be an NP relation with witness length  $m = m(n)$ , which can be verified in  $\text{poly}(n)$  time and space  $S = S(n)$ . Then, for every parameter  $q = q(n, m)$  and constant  $\delta > 0$ , there exists a constant-round oblivious IWV for  $R$ . The query complexity is  $q$  and the communication complexity is  $cc = cc(n, m) = ((m/q) \cdot m^\delta \cdot \text{poly}(S))$ . The verifier runs in time  $(n \cdot \text{polylog}(n, m) + \tilde{O}(cc))$ . The prover runs in time  $\text{poly}(n, m)$ , given as input the NP witness.*

We remark that a similar result holds for any language computable in (log-space uniform) NC, where in the case of NC the number of rounds is  $\text{polylog}(n, m)$  rather than constant, and the  $m^\delta$  terms in the communication complexity and the verifier runtime are replaced by  $m^{o(1)}$ .

### 3.1 Constructing IWVs for NP

The main technical tool that we use to prove Theorem 9 is the *interactive proofs of proximity* (IPPs) protocol of Rothblum, Vadhan and Wigderson [32]. First, in Section 3.1.1 we introduce the model of IPPs and state the [32] result. Then, in Section 3.1.2, we prove Theorem 9.

#### 3.1.1 Background on IPPs

Loosely speaking, IPPs are interactive proofs in which the verifier runs in sub-linear time in the input length and is assured that the input is *close* to the language. Actually, we will think of the input of the verifier as being composed of two parts: a short input  $x \in \{0, 1\}^n$  to which the verifier has direct access and a long input  $y \in \{0, 1\}^m$  to which the verifier has oracle access. The goal is for the verifier to run in time that is sub-linear in  $m$  and to verify that  $y$  is far from any  $y'$  such that the pair  $(x, y')$  are in the language. Since such languages are composed of input pairs, we refer to them as *pair languages*.

► **Definition 10** (Interactive Proof of Proximity (IPP) [12, 32]). *An interactive proof of proximity (IPP) for the pair language  $\mathcal{L}$  is an interactive protocol with two parties: a (computationally unbounded) prover  $\mathcal{P}$  and a computationally bounded verifier  $\mathcal{V}$ . Both parties get as input  $x \in \{0, 1\}^n$  and a proximity parameter  $\varepsilon > 0$ . The verifier also gets oracle access to  $y \in \{0, 1\}^m$  whereas the prover has full access to  $y$ . At the end of the interaction, the following two conditions are satisfied:*

1. **Completeness:** For every pair  $(x, y) \in \mathcal{L}$ , and proximity parameter  $\varepsilon > 0$  it holds that

$$\Pr \left[ (\mathcal{P}(y), \mathcal{V}^y)(x, |y|, \varepsilon) = 1 \right] = 1.$$

2. **Soundness:** For every  $\varepsilon > 0$ ,  $x \in \{0, 1\}^n$  and  $y$  that is  $\varepsilon$ -far from the set  $\{y' : (x, y') \in \mathcal{L}\}$ , and for every computationally unbounded (cheating) prover  $\mathcal{P}^*$  it holds that

$$\Pr \left[ (\mathcal{P}^*(y), \mathcal{V}^y)(x, |y|, \varepsilon) = 1 \right] \leq 1/2.$$

An IPP for  $\mathcal{L}$  is said to have **query complexity**  $q = q(n, m, \varepsilon)$  if, for every  $\varepsilon > 0$  and  $(x, y) \in \mathcal{L}$ , the verifier  $\mathcal{V}$  makes at most  $q(|x|, |y|, \varepsilon)$  queries to  $y$  when interacting with  $\mathcal{P}$ . The IPP is said to have **communication complexity**  $\text{cc} = \text{cc}(n, m, \varepsilon)$  if, for every  $\varepsilon > 0$  and pair  $(x, y) \in \mathcal{L}$ , the communication between  $\mathcal{V}$  and  $\mathcal{P}$  consists of at most  $\text{cc}(|x|, |y|, \varepsilon)$  bits.

We are now ready to state the main result of [32]. Actually we will use an extension, due to [31], of the [32] IPP.<sup>11</sup>

► **Theorem 11** (IPPs for Bounded Space Computations, [32, 31]). *Let  $\mathcal{L}$  be a pair language that is computable in  $\text{poly}(n, m)$  time and space  $S = S(n, m)$ . For every constant  $\delta > 0$ , there is an IPP for  $\mathcal{L}$  with the following parameters. For every input pair  $(x, y) \in \{0, 1\}^n \times \{0, 1\}^m$ , the query complexity is  $q = q(n, m, \varepsilon) = (1/\varepsilon) \cdot m^{O(\delta)}$ , the communication complexity is  $\text{cc} = \text{cc}(n, m, \varepsilon) = (\varepsilon \cdot m^{1+O(\delta)}) \cdot \text{poly}(S)$ , and the number of rounds is constant. The honest prover runs in time  $\text{poly}(n, m, 1/\varepsilon)$  and the verifier runs in time  $n \cdot \text{polylog}(n, m) + \tilde{O}(q + \text{cc})$ .*

*Furthermore, the verification can be implemented in two phases. In the communication phase the verifier interacts with the prover without querying  $y$ . The verifier's running time in this phase is  $n \cdot \text{polylog}(n, m) + \tilde{O}(\text{cc})$ . At the end of the communication phase, the verifier either rejects or it outputs a succinct description  $\langle Q \rangle$  of a set  $Q \subseteq [m]$  of size  $q$  and succinct description  $\langle \phi \rangle$  of a predicate  $\phi : \{0, 1\}^q \rightarrow \{0, 1\}$  which can be computed by a (logspace uniform)  $\text{NC}^1$  circuit of size  $\tilde{O}(q)$ . In the query phase, the verifier only queries  $y_Q$  and accepts if and only if  $\phi(y_Q) = 1$ .*

### 3.1.2 Proof of Theorem 9

Let  $R$  be an NP relation with witness length  $m = m(n)$ , which can be verified in  $\text{poly}(n)$  time and space  $S = S(n)$ . Let  $\delta > 0$  be a constant and let  $q = q(n, m) \in [m]$  be a parameter.

View  $R$  as a pair language. Namely each input-witness pair  $(x, w)$  is viewed as an input pair  $(x, w)$  for the pair language. Let  $(\mathcal{P}, \mathcal{V})$  be the IPP for  $R$  guaranteed by Theorem 11 with respect to proximity parameter  $\varepsilon = \frac{m^{O(\delta)}}{q}$ . We claim that  $(\mathcal{P}, \mathcal{V})$  is an IWV for  $R$ , where queries to the IPP input oracle  $y$  are emulated by querying the IWV witness oracle.

Completeness follows immediately from the completeness of the IPP. For soundness, let  $x \in \{0, 1\}^n$  such that  $R(x) = \emptyset$ . Fix an alleged witness string  $\tilde{y}$  and an IWV prover strategy  $\mathcal{P}^*$ . We view  $\mathcal{P}^*$  as a cheating prover strategy for the IPP  $(\mathcal{P}, \mathcal{V})$  with respect to the input pair  $(x, \tilde{y})$ . Since  $R(x) = \emptyset$ , it holds that  $(x, \tilde{y})$  is at infinite distance from the set  $\{y' : (x, y') \in R\}$  (in particular the distance is more than  $\varepsilon$ ). Thus, by the IPP soundness, the verifier rejects with probability at least  $1/2$ .

The fact that the foregoing IWV is *oblivious*, as well as its complexity, follows from the furthermore part of Theorem 11.

## 3.2 Lower Bound for IWVs

In this section we show a lower bound on the efficiency of IWVs that, loosely speaking, shows that Theorem 9 is tight. This lower bound relies on the existence of an *exponentially strong*

<sup>11</sup>The [31] IPP extends the [32] result from bounded depth computations to also hold for bounded space computations. Also, and more importantly for our purposes, the [31] IPP only requires a constant number of rounds (for languages computable in bounded space).



cryptographic pseudorandom generator (PRG). By exponentially strong, we mean that the output of the generators, when evaluated of a random string of length  $m$ , computationally indistinguishable from a uniformly random string even for adversaries running in time  $2^{m/100} \cdot \text{poly}(m)$ . We remark that by assuming only sub-exponential hardness (i.e., hardness against  $2^{m^\epsilon}$  time adversaries) we can still obtain a meaningful (albeit weaker) lower bound.

For an NP relation  $R$ , we denote by  $R^{\otimes k}$  the relation

$$R^{\otimes k} \stackrel{\text{def}}{=} \left\{ ((x_1, \dots, x_k), (w_1, \dots, w_k)) : \forall j \in [k], (x_j, w_j) \in R \text{ and } |x_1| = \dots = |x_k| \right\}.$$

► **Theorem 12.** *Assume the existence of an exponentially hard cryptographic PRG. Then, there exists an NP relation  $R$  with witnesses of size  $m$ , such that for every  $k \leq \text{poly}(m)$ , every IWV for  $R^{\otimes k}$  must have either query complexity  $q = \Omega(k)$  or communication complexity  $cc = \Omega(m)$ . Furthermore, if the PRG is injective, then  $R$  is a UP relation.*

The proof of Theorem 12 follows in a straightforward way from the IPP lower bound of Kalai and Rothblum [28]. We provide a proof sketch below.

**Proof Sketch.** Let  $G : \{0, 1\}^m \rightarrow \{0, 1\}^n$  be an exponentially strong PRG. Let  $R_G = \{(y, s) : y = G(s)\}$ . Clearly  $R_G \in \text{NP}$  and if  $G$  is injective, then  $R_G \in \text{UP}$ . Suppose toward a contradiction that there exists an IWV  $(\mathcal{P}, \mathcal{V})$  for  $\mathcal{R}_G^{\otimes k} = \left\{ ((y_1, \dots, y_k), (s_1, \dots, s_k)) : \forall j \in [k], y_j = G(s_j) \right\}$  with query complexity  $k/100$  and communication complexity  $m/100$ .

The proof is composed of two steps. First, we use  $(\mathcal{P}, \mathcal{V})$  to construct a relatively efficient interactive proof for  $R_G$  (i.e., with communication  $m/100$ ). The second step is to show that such an interactive proof violates the exponential hardness of  $G$ .

We start with the first step: constructing an interactive proof  $(\mathcal{P}', \mathcal{V}')$  for  $R_G$  - i.e., deciding whether a given string is in the image of  $G$ . Actually, we only achieve a relaxed notion of interactive proof. Specifically we have the following two relaxations:

- (Average-case Completeness:) Completeness holds for *most* inputs in the language but not necessarily for all inputs. Namely, for most  $s$ , the verifier  $\mathcal{V}'$  accepts with high probability after interacting with the prover  $\mathcal{P}'$  on common input  $G(s)$ .
- (Common Random String:) Both the prover and verifier have access to a (relatively long) common random string. We do not count this random string as part of the communication complexity of the protocol.

We proceed to describe the interactive proof  $(\mathcal{P}', \mathcal{V}')$  for  $R_G^{\otimes k}$ . The common random string consists of  $((s_1, \dots, s_k), i) \in_R (\{0, 1\}^m)^k \times [k]$ . We define  $y_j = G(s_j)$ , for all  $j \in [k]$ . In addition to the common random string, the verifier  $\mathcal{V}'$  and prover  $\mathcal{P}'$  are given as input  $y \in \{0, 1\}^n$ , and  $\mathcal{V}'$  needs to decide whether there exists  $s \in \{0, 1\}^m$  such that  $G(s) = y$  (i.e., whether  $R_G(y) \neq \emptyset$ ).

The interactive proof proceeds as follows. The prover  $\mathcal{P}'$  and verifier  $\mathcal{V}'$  run the IWV  $(\mathcal{P}, \mathcal{V})$  where the input is  $(y_1, \dots, y_{i-1}, y, y_{i+1}, \dots, y_k)$  and the witness, to which  $\mathcal{V}$  only gets oracle access, is  $(s_1, \dots, s_{i-1}, 0^m, s_{i+1}, \dots, s_k)$ .

To show that average-case completeness holds, let  $s \in_R \{0, 1\}^m$  and consider the execution of  $(\mathcal{P}', \mathcal{V}')$  on input  $y = G(s)$ . Consider a mental experiment in which we run the IWV with the witness  $(s_1, \dots, s_{i-1}, s, s_{i+1}, \dots, s_k)$  (rather than  $(s_1, \dots, s_{i-1}, 0^m, s_{i+1}, \dots, s_k)$  as in the real execution). By the completeness of the IWV, in this mental experiment,  $\mathcal{V}$  accepts.

Observe that, conditioned on not querying  $s$ , the view of  $\mathcal{V}$  is identical in the real execution and in the mental experiment, and so it will accept also in the real execution. Moreover, since  $i$  and  $y$  are random, and  $\mathcal{V}$  makes at most  $k/100$  queries, with constant probability,  $\mathcal{V}$  does not query  $s$  and average-case completeness follows.

Soundness of  $(\mathcal{P}', \mathcal{V})$  is easier to show. Specifically, if  $R_G(y) = \emptyset$  then  $R_G^{\otimes k}(y_1, \dots, y_{i-1}, y, y_{i+1}, \dots, y_\ell) = \emptyset$  and so by the soundness of the IWV, the verifier  $\mathcal{V}$  rejects with high probability given oracle access to any fixed witness (in particular,  $(s_1, \dots, s_{i-1}, 0^m, s_{i+1}, \dots, s_k)$ ) and no matter what the cheating prover does.

Thus,  $(\mathcal{P}', \mathcal{V})$  is an interactive proof for  $R_G$ . Observe that  $(\mathcal{P}', \mathcal{V})$  has the same communication as  $(\mathcal{P}, \mathcal{V})$  - namely,  $m/100$ .

The second step of the proof is to observe that the foregoing interactive proof can be emulated by an *algorithm*  $A$  running in time  $2^{m/100} \cdot \text{poly}(m, k)$ . This is similar to the proof that  $\text{IP} \subseteq \text{PSPACE}$  (i.e., interactive proofs can be emulated by bounded space machines).<sup>12</sup> Thus, using the fact that  $k = \text{poly}(m)$ , the PRG can be broken in time  $2^{m/100} \cdot \text{poly}(m)$  time, in contradiction to our assumption. ◀

## 4 Batch Verification for UP

For an NP relation  $R$ , we denote by  $R^{\otimes k}$  the relation

$$R^{\otimes k} \stackrel{\text{def}}{=} \left\{ ((x_1, \dots, x_k), (w_1, \dots, w_k)) : \forall j \in [k], (x_j, w_j) \in R \text{ and } |x_1| = \dots = |x_k| \right\}.$$

► **Theorem 13** (Batch Verification for UP). *Let  $R$  be a UP relation that is verifiable in  $\text{NC}^1$ , with witnesses of length  $m = m(n)$  such that  $m$  and  $n$  are polynomially related. Let  $k = k(n) \geq 1$  and let  $\delta > 0$  be a constant. There exists a constant-round interactive proof system for  $R^{\otimes k}$  such that the verifier runs in time  $(\tilde{O}(n \cdot k) + k^\delta \cdot m^{1+\delta})$ , the (honest) prover runs in time  $\text{poly}(n, m, k)$  and the communication complexity is  $(k^\delta \cdot m^{1+\delta})$ .*

By the proof of the Cook-Levin theorem every UP language has a UP relation that is verifiable in  $\text{NC}^1$  (albeit with a polynomial blowup in the witness size). Thus, Theorem 13 is applicable to *any* UP language.

As described in the technical overview (see Section 1.2), our batch verification protocol works in iterations, where the goal of each iteration is to significantly reduce the number of instances that are still “alive”. In Section 4.1 we describe the iterative step and then in Section 4.2 we describe the UP batch verification protocol.

### 4.1 The Iterative Step

We first describe the main step in our proof, corresponding to a single iteration of the protocol that was described in Section 1.2. Loosely speaking, this step shows an interactive protocol, where if we start with a false claim about a subset of the  $k$  UP statements, then at the end of the protocol, with high probability, we will have a false claim about a smaller subset of the statements.

This step, which appears next in Theorem 14, is where we rely on the existence of IWVs for general  $\text{NC}^1$  relations. Theorem 14 can be instantiated with any such IWV. Later, in Section 4.2, we will use Theorem 14 instantiated with the IWVs that were shown to exist (unconditionally) in Theorem 9.

► **Lemma 14.** *Suppose that for every parameter  $q$ , every NP relation computable in  $\text{NC}^1$  has an oblivious IWV with an efficient prover such that with respect to inputs of size  $n$  and witness of size  $m$ , the proof-system has soundness error  $\varepsilon = \varepsilon(n, m, q)$ , verifier complexity*

<sup>12</sup>We cannot afford to count the CRS as part of the communication of the interactive proof, since it is of length  $m \cdot k + \log(k) \gg m/100$ . Rather, observe that  $A$  can simply sample the CRS directly, in time  $\text{poly}(m, k)$  (rather than enumerating over all possible CRS strings).



$\mathcal{V}\text{time} = \mathcal{V}\text{time}(n, m, q)$ , prover complexity  $\mathcal{P}\text{time}(n, m, q)$  (assuming the prover is given access to the NP witness), round complexity  $r(n, m, q)$ , query complexity  $q$  and communication complexity  $\text{cc} = \text{cc}(n, m, q)$ .

Let  $R$  be a UP relation computable in  $\text{NC}^1$ , with witnesses of length  $m = m(n)$ , and let  $\delta > 0$  be a constant. There exists an interactive protocol between a prover  $\mathcal{P}$  and verifier  $\mathcal{V}$  such that the following holds. Both parties get as input  $\mathbf{x} = (x_1, \dots, x_k) \in (\{0, 1\}^n)^k$  and succinct descriptions  $\langle S \rangle$  and  $\langle \phi \rangle$ , of a set  $S \subseteq [k]$  of size  $s$  and circuit  $\phi$ , respectively. The prover  $\mathcal{P}$  also gets witnesses  $\mathbf{w} = (w_1, \dots, w_k) \in (\{0, 1\}^m)^k$  as an additional input. The two parties interact and at the end of the interaction  $\mathcal{V}$  either rejects or outputs succinct descriptions  $\langle S' \rangle$  and  $\langle \phi' \rangle$  of a subset  $S' \subseteq S$ , of size  $s^{1-\delta}$ , and circuit  $\phi'$ , respectively, such that:

- **(Completeness:)** If  $(x_j, w_j) \in R$  for all  $j \in S$ , and  $\phi(\mathbf{w}|_S) = 1$ , then, with probability 1, after interacting with  $\mathcal{P}$ , the verifier  $\mathcal{V}$  outputs  $\langle S' \rangle$  and  $\langle \phi' \rangle$  such that  $\phi'(\mathbf{w}|_{S'}) = 1$ .
- **(Soundness:)** If either (1) there exists  $j \in S$  such that  $R(x_j) = \emptyset$ , or (2)  $(x_j, w_j) \in R$  for all  $j \in S$  but  $\phi(\mathbf{w}|_S) = 0$ , then, for every prover strategy  $\mathcal{P}^*$ , with probability  $1 - \varepsilon$ , after interacting with  $\mathcal{P}^*$ , the verifier  $\mathcal{V}$  either rejects or outputs  $\langle S' \rangle$  and  $\langle \phi' \rangle$  such that one of the following holds:
  1.  $\exists j \in S'$  such that  $R(x_j) = \emptyset$ ; or
  2.  $\phi'(\mathbf{w}|_{S'}) = 0$ .
- **(Complexity:)** The protocol  $(\mathcal{P}, \mathcal{V})$  has verifier complexity  $\mathcal{V}\text{time}(n \cdot k + \text{poly}(\log n, \log k), s \cdot m, q)$ , prover complexity  $\mathcal{P}\text{time}(n \cdot k + \text{poly}(\log n, \log k), s \cdot m, q)$  (assuming the prover is given access to the  $k$  UP witnesses), round complexity  $r = r(n \cdot k + \text{poly}(\log n, \log k), s \cdot m, q)$  and communication complexity  $\text{cc} = \text{cc}(n \cdot k + \text{poly}(\log n, \log k), s \cdot m, q)$ , where  $q = s^{1-\delta}$ .

**Proof.** Let  $R$  be a UP relation computable in  $\text{NC}^1$ . We consider a related NP relation  $R_k$  defined as follows. The input to  $R_k$  is  $(x_1, \dots, x_k, \langle S \rangle, \langle \phi \rangle)$  and the witness is a sequence  $\mathbf{w}|_S = (w_j)_{j \in S}$ . The relation checks that (1) for every  $j \in S$  it holds that  $(x_j, w_j) \in R$ , and (2) that  $\phi(\mathbf{w}|_S) = 1$ . Observe that membership in  $R_k$  can be decided in (logspace uniform)  $\text{NC}^1$ , and therefore, by the lemma's hypothesis there exists an oblivious IWW  $(\mathcal{P}, \mathcal{V})$  for  $R_k$  where we use parameter  $q = s^{1-\delta}$ , where  $s$  is the size of the set  $S$ .

We use  $(\mathcal{P}, \mathcal{V})$  to construct a protocol  $(\mathcal{P}_k, \mathcal{V}_k)$  as required in the theorem's statement. Given as common input  $(x_1, \dots, x_k, \langle S \rangle, \langle \phi \rangle)$ , the verifier  $\mathcal{V}_k$  and prover  $\mathcal{P}_k$  run  $(\mathcal{P}, \mathcal{V})$  with respect to the common input  $(x_1, \dots, x_k, \langle S \rangle, \langle \phi \rangle)$ , where  $\mathcal{P}_k$  gets as an auxiliary input also  $\mathbf{w}|_S = (w_j)_{j \in S}$ .

If  $\mathcal{V}$  rejects then  $\mathcal{V}_k$  immediately rejects. Otherwise,  $\mathcal{V}$  outputs succinct  $\text{NC}^1$  descriptions  $\langle Q' \rangle$  and  $\langle \phi' \rangle$  where  $Q' \subseteq [k] \times [m]$ , of size  $k^{1-\delta}$  specifies which locations to read from  $\mathbf{w}|_S$  and  $\phi'$  is a predicate specifying whether  $\mathcal{V}$  would have accepted had it read those bits. For simplicity, and without loss of generality, we assume that  $Q$  specifies  $k^{1-\delta}$  of the witnesses  $w_1, \dots, w_k$  entirely and ignores the rest. Let  $S' \subseteq [k]$  denote the witnesses that the  $Q$  refers to. The verifier  $\mathcal{V}_k$  outputs  $\langle S' \rangle$  and  $\langle \phi' \rangle$ .

**Completeness.** Let  $x_1, \dots, x_k$  be a sequence of inputs,  $S \subseteq [k]$  a set and  $\phi$  a circuit such that there exist unique  $\mathbf{w}|_S = (w_j)_{j \in S}$  such that  $(x_j, w_j) \in R$  for all  $j \in [k]$  and  $\phi(\mathbf{w}|_S) = 1$ . The IWW protocol is run with respect to an input  $((x_1, \dots, x_k, \langle S \rangle, \langle \phi \rangle), \mathbf{w}_S) \in R_k$ . Thus, by the completeness of the IWW, with probability 1, it holds that  $\phi'(\mathbf{w}|_{S'}) = 1$ .

**Soundness.** Suppose that either there exists  $j \in S$  such that  $R(x_j) = \emptyset$ , or  $\mathbf{w}|_S = (w_j)_{j \in S}$  consists of the corresponding unique witnesses and  $\langle \phi \rangle$  is such that  $\phi(\mathbf{w}|_S) = 0$ . Let  $\mathcal{P}^*$  be a cheating prover strategy. To show that the soundness condition holds, it suffices to prove the following claim:

► **Claim 14.1.**

$$\Pr \left[ (\forall j \in S', R(x_j) \neq \emptyset) \text{ and } (\phi'(\mathbf{w}|_{S'}) = 0) \right] \leq \varepsilon$$

**Proof.** For every  $j \in S$ , if  $R(x_j) \neq \emptyset$  then define  $\hat{w}_j = w_j$  (i.e., the unique witness for  $x_j$ ), whereas if  $R(x_j) = \emptyset$ , then define  $\hat{w}_j$  as some arbitrary fixed string (e.g.,  $0^m$ ).

We view  $\mathcal{P}^*$  as an adversary for the oblivious IWW protocol, with respect to the a priori fixed witness string  $\hat{\mathbf{w}}|_S = (\hat{w}_j)_{j \in S}$ . By the soundness condition of the oblivious IWW (see Theorem 8), it holds that:

$$\Pr[\phi'(\hat{\mathbf{w}}_{S'}) = 1] \leq \varepsilon.$$

For all  $j \in S$  we have that, if  $R(x_j) \neq \emptyset$  then  $w_j = \hat{w}_j$ . Thus,

$$\Pr \left[ (\forall j \in S', R(x_j) \neq \emptyset) \text{ and } (\phi'(\mathbf{w}|_{S'}) = 0) \right] \leq \Pr [\phi'(\hat{\mathbf{w}}|_{S'}) = 0] \leq \varepsilon,$$

and the claim follows. ◀

**Complexity.** The stated complexity follows from the complexity of the IWW, which is run on an input of size  $n \cdot k + \text{poly}(\log n, \log k)$  (the concatenation of the  $k$  inputs and succinct representations  $\langle S \rangle$  and  $\langle \phi \rangle$ ), witness size  $s \cdot m$  (i.e., the length of  $(w_i)_{i \in S}$  - the concatenation of the relevant witnesses) and with respect to the parameter  $q = s^{1-\delta}$ . ◀

## 4.2 The Batch Verification Protocol: Proof of Theorem 13

Let  $(P_{\text{reduction}}, V_{\text{reduction}})$  be the protocol guaranteed by Theorem 14, with respect to UP relation  $R$  and the IWW protocol of Theorem 9. We construct a protocol  $(\mathcal{P}_{\text{batch}}, \mathcal{V}_{\text{batch}})$  satisfying the requirement of Theorem 13. The protocol is described in Fig. 1.

To complete the proof of Theorem 13 we need to show that completeness and soundness hold, as well as analyze the complexity of the protocol.

**Completeness.** Let  $x_1, \dots, x_k$  such that there exist (unique) witnesses  $\mathbf{w} = (w_1, \dots, w_k)$  such that  $(x_j, w_j) \in R$ , for every  $j \in [k]$ . Let  $S_1, \dots, S_\ell$  and  $\phi_1, \dots, \phi_\ell$  be the sets and formulas, respectively, generated in the interaction between  $\mathcal{P}_{\text{batch}}$  and  $\mathcal{V}_{\text{batch}}$ .

► **Claim 14.2.** *For every  $i \in [\ell]$ , with probability 1, it holds that  $\mathcal{V}_{\text{batch}}$  does not reject prior to the  $i^{\text{th}}$  iteration and  $\phi_i(\mathbf{w}_{S_i}) = 1$ .*

**Proof.** We prove by induction on  $i$ . In the case base  $\phi_1$  always outputs 1 and so the claim holds trivially. Suppose that the claim holds for some value  $i$ . Thus,  $\phi_i(\mathbf{w}_{S_i}) = 1$ .

Since  $(x_j, w_j) \in R$  for all  $j \in S_i$ , and  $\phi_i(\mathbf{w}_{S_i}) = 1$ , by the completeness of  $(P_{\text{reduction}}, V_{\text{reduction}})$  it holds that  $V_{\text{reduction}}$  does not reject and outputs  $\langle S_{i+1} \rangle$  and  $\langle \phi_{i+1} \rangle$  such that  $\phi_{i+1}(\mathbf{w}_{S_{i+1}}) = 1$ . The claim follows. ◀

Thus, at the end of the loop  $\phi_\ell(\mathbf{w}_{S_\ell}) = 1$ . Since  $\mathcal{P}_{\text{batch}}$  sends the correct (unique) witnesses  $\mathbf{w}_{S_\ell}$  in Step 3, by the completeness of the interactive proof-system of Theorem 6, the verifier  $\mathcal{V}_{\text{batch}}$  accepts.

**The UP Batching Protocol** ( $\mathcal{P}_{\text{batch}}, \mathcal{V}_{\text{batch}}$ )

Common Input:  $\mathbf{x} = (x_1, \dots, x_k) \in (\{0, 1\}^n)^k$ .

Prover's Auxiliary Input: witnesses  $\mathbf{w} = (w_1, \dots, w_k) \in (\{0, 1\}^m)^k$ .

1. Set  $\langle S_1 \rangle$  to be a concise description of the set  $S_1 = [k]$ , and  $\langle \phi_1 \rangle$  to be a concise description of a circuit  $\phi_1 : (\{0, 1\}^m)^k \rightarrow \{0, 1\}$  that always outputs 1.
2. For  $i = 1, \dots, \ell - 1$ , where  $\ell = O(1/\delta)$ :
  - a. Run  $(\mathcal{P}_{\text{reduction}}, \mathcal{V}_{\text{reduction}})$  on common input  $(\mathbf{x}, \langle S_i \rangle, \langle \phi_i \rangle)$ , and with respect to parameter  $q_i = s_i^{1-\delta}$ , where  $s_i$  is the size of the set  $S_i$ , and with soundness error  $\varepsilon = 1/(10\ell)$ .<sup>a</sup> More specifically,  $\mathcal{V}_{\text{batch}}$  emulates  $\mathcal{V}_{\text{reduction}}$  and  $\mathcal{P}_{\text{batch}}$  emulates  $\mathcal{P}_{\text{reduction}}$ , using  $\mathbf{w}|_{S_i}$  as the auxiliary input.
  - b. If  $\mathcal{V}_{\text{reduction}}$  rejects then  $\mathcal{V}_{\text{batch}}$  immediately rejects. Otherwise  $\mathcal{V}_{\text{reduction}}$  outputs  $\langle S_{i+1} \rangle$  and  $\langle \phi_{i+1} \rangle$ .
3.  $\mathcal{P}_{\text{batch}}$  sends to  $\mathcal{V}_{\text{batch}}$  the witnesses  $\mathbf{w}|_{S_\ell} = (w_j)_{j \in S_\ell}$ .
4. The verifier  $\mathcal{V}_{\text{batch}}$  expands  $\langle S_\ell \rangle$  to a full description of the set  $S_\ell$ . The prover and verifier then run the doubly efficient interactive proof of Theorem 6 on input  $((x_j)_{j \in S_\ell}, (w_j)_{j \in S_\ell}, \langle \phi_\ell \rangle)$  checking that for every  $j \in S_\ell$  it holds that  $(x_j, w_j) \in R$  and that  $\phi_\ell(\mathbf{w}|_{S_\ell}) = 1$  (the protocol of Theorem 6 is used with a sufficiently small parameter  $\tau > 0$  to be determined in the analysis). If all checks pass then  $\mathcal{V}_{\text{batch}}$  accepts and otherwise it rejects.<sup>b</sup>

<sup>a</sup> Such soundness amplification can be achieved by repeating the base protocol  $O(\log(\ell))$  times in parallel.

<sup>b</sup> This step could be replaced by having the verifier directly check by itself that  $(x_j, w_j) \in R$  for every  $j \in S_\ell$ . However, doing so introduces an additive overhead of  $\text{poly}(n, m)$  to the verifier's running time (arising from the complexity of the relation  $R$ ) which we can reduce to  $\tilde{O}(n + m)$  by using the interactive proof of Theorem 6.

■ **Figure 1** UP Batching.

**Soundness.** Let  $x_1, \dots, x_k \in \{0, 1\}^n$  such that  $\exists j^* \in [k]$  with  $R(x_{j^*}) = \emptyset$ . Let  $\mathcal{P}^*$  be a cheating prover strategy. For every  $j \in [k]$ , if  $R(x_j) \neq \emptyset$ , let  $w_j$  be the corresponding unique witness. Purely for notational convenience, we also define  $w_j$  to be an arbitrary string, for every  $j \in [k]$  such that  $R(x_j) = \emptyset$ . Let  $\mathbf{w} = (w_1, \dots, w_k)$ .

For every  $i \in [\ell]$ , let  $E_i$  denote the conjunction of the following three events:

- The verifier  $\mathcal{V}_{\text{batch}}$  has not rejected prior to the start of the  $i^{\text{th}}$  iteration; and
- For every  $j \in S_i$  it holds that  $R(x_j) \neq \emptyset$ ; and
- $\phi_i(\mathbf{w}|_{S_i}) = 1$

Setting  $\varepsilon = 1/(10\ell)$ , we have the following central claim:

► **Claim 14.3.** For every  $i \in [\ell]$ :

$$\Pr[E_i] \leq (i - 1) \cdot \varepsilon$$

**Proof.** We prove the claim by induction on  $i$ . For the base case  $i = 1$ , since  $R(x_{j^*}) = \emptyset$  and  $j^* \in [k] = S_1$ , it holds that

$$\Pr[E_1] \leq \Pr[\forall j \in S_1, R(x_j) \neq \emptyset] \leq \Pr[R(x_{j^*}) \neq \emptyset] = 0.$$

Assume that the claim holds for some  $i \in [\ell - 1]$ . By elementary probability theory,

$$\Pr[E_{i+1}] \leq \Pr[E_i] + \Pr[E_{i+1} | \neg E_i].$$

By the induction hypothesis the first term is bounded by  $(i - 1) \cdot \varepsilon$  and so to complete the proof we need to bound the second term by  $\varepsilon$ . This holds since:

$$\begin{aligned} \Pr[E_{i+1} \mid \neg E_i] &\leq \Pr[E_{i+1} \mid \mathcal{V}_{\text{batch}} \text{ rejects prior to iteration } i] \\ &\quad + \Pr \left[ E_{i+1} \mid \begin{array}{c} (\exists j \in S_i \text{ s.t. } R(x_j) = \emptyset) \\ \text{or} \\ (\phi_i(\mathbf{w}|_{S_i}) = 0) \end{array} \right] \\ &= \Pr \left[ E_{i+1} \mid \begin{array}{c} (\exists j \in S_i \text{ s.t. } R(x_j) = \emptyset) \\ \text{or} \\ (\phi_i(\mathbf{w}|_{S_i}) = 0) \end{array} \right] \\ &\leq \varepsilon, \end{aligned}$$

where the equality is since if  $\mathcal{V}_{\text{batch}}$  rejects prior to iteration  $i$  then clearly it also rejects prior to iteration  $i + 1$  (and so  $E_{i+1}$  does not occur), and the last inequality follows directly from the soundness condition of Theorem 14 (where recall that we used that protocol with soundness error  $\varepsilon = 1/(10\ell)$ ).  $\blacktriangleleft$

Thus, with probability at least  $1 - \ell \cdot \varepsilon \geq 0.9$ , one of the following events occurs at the end of the loop:

1.  $\mathcal{V}_{\text{batch}}$  has already rejected; or
2. There exists  $j \in [S_\ell]$  such that  $R(x_j) = \emptyset$ ; or
3.  $\phi_\ell(\mathbf{w}|_{S_\ell}) = 0$ .

We show that in each of these cases, the verifier rejects with high probability. For the first case this is immediate. In the second case, the cheating prover must send some incorrect witness  $w_j^*$ . Thus, the verifier and prover run the doubly efficient interactive proof-system of Theorem 6 on a false input, and by the soundness condition of that protocol, the verifier rejects with probability 0.9.

Lastly, if  $\phi_\ell(\mathbf{w}_{S_\ell}) = 0$ , then either  $\mathcal{P}^*$  sends witnesses that are not the unique witnesses, in which case again the protocol of Theorem 6 is run on a false statement or  $\mathcal{P}^*$  sends the unique witnesses but in this case the statement is still false since  $\phi_\ell(\mathbf{w}|_{S_\ell}) = 1$ . Thus, in both cases, by the soundness of Theorem 6, the verifier rejects with probability 0.9.

Thus, in all cases the verifier rejects with probability at least  $0.9^2 \geq 1/2$ .

**Complexity.** For every  $i \in [\ell]$ , let  $s_i$  denote the size of the set  $S_i$  generated in the interaction. By Theorem 14 it holds that  $s_i \leq k^{1-(i-1)\delta}$ .

Consider the  $i^{\text{th}}$  iteration of the loop, for some  $i \in [\ell - 1]$ . Let  $n_i = n \cdot k + \text{polylog}(n, k)$  and  $m_i = s_i \cdot m \leq k^{1-(i-1)\delta} \cdot m$  and recall that we set  $q_i = s_i^{1-\delta} \leq k^{1-i\delta}$ . By Theorem 9, together with Theorem 14, the  $i^{\text{th}}$  iteration takes a constant number of rounds and has:

- Communication complexity:

$$\begin{aligned} \text{cc}_i &= (m_i/q_i) \cdot m_i^\delta \cdot \text{polylog}(n, m) \\ &\leq \left( (k^{1-(i-1)\delta} \cdot m) / k^{1-i\delta} \right) \cdot (k \cdot m)^\delta \cdot \text{polylog}(n, m) \\ &= k^{2\delta} \cdot m^{1+\delta} \cdot \text{polylog}(n, m) \end{aligned}$$

- Verifier running time:

$$\begin{aligned} \mathcal{V}\text{time}_i &= n_i \cdot \text{polylog}(n_i, m_i) + \tilde{O}(\text{cc}_i) \\ &\leq n \cdot k \cdot \text{polylog}(n, k, m) + k^{2\delta} \cdot m^{1+\delta} \cdot \text{polylog}(n, m, k) \end{aligned}$$

- And prover running time (given the UP witnesses):

$$\begin{aligned} \mathcal{P}\text{time}_i &= \text{poly}(n_i, m_i) \\ &= \text{poly}(n, m, k) \end{aligned}$$

To analyze the last two steps of the protocol, first observe that  $s_\ell$  (i.e., the size of the final set  $S_\ell$ ) has size  $s_\ell \leq k^{1-(\ell-1)\delta} = O(1)$ . Thus, Step 3 only adds an additional  $s_\ell \cdot m = O(m)$  communication.

As for the verification time, generating the set  $S_\ell$  takes time  $\text{poly}(s_\ell, \log k) = \text{polylog}(k)$  (by the definition of concise description of sets, see Theorem 4). The protocol of Theorem 6, is run on a logspace computation and with its parameter  $\tau$  set to be a sufficiently small constant so that the communication is  $O(m)$ . This protocol takes an additional  $O(1)$  rounds, the verifier runs in time  $\tilde{O}(n+m)$ , the prover runs in time  $\text{poly}(n, m)$  and the communication complexity is  $O(m)$ .

The parameters stated in the theorem's statement now follows by taking resetting  $\delta$  to be sufficiently small (e.g., take  $\delta' = \delta/4$ ) and the fact that  $m$  and  $n$  are polynomially related.

---

## References

- 1 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.
- 2 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.
- 3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; A new characterization of NP. In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992*, pages 2–13. IEEE Computer Society, 1992. doi:10.1109/SFCS.1992.267824.
- 4 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 21–31. ACM, 1991. doi:10.1145/103418.103428.
- 5 László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991. doi:10.1007/BF01200056.
- 6 László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988. doi:10.1016/0022-0000(88)90028-1.
- 7 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 113–131. ACM, 1988. doi:10.1145/62212.62223.
- 8 Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. Cryptology ePrint Archive, Report 2016/116, 2016. <http://eprint.iacr.org/>.
- 9 Itay Berman, Ron D. Rothblum, and Vinod Vaikuntanathan. Zero-knowledge proofs of proximity. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 19:1–19:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.19.

- 10 Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. 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 474–482. ACM, 2017. doi:10.1145/3055399.3055497.
- 11 Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPICs*, pages 53:1–53:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPICs.ITCS.2018.53.
- 12 Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004. doi:10.1016/j.ic.2003.09.005.
- 13 Uriel Feige, Shafi Goldwasser, László 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.
- 14 Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In Moni Naor, editor, *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 483–500. ACM, 2014. doi:10.1145/2554797.2554841.
- 15 Lance Fortnow, John Rempel, 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.
- 16 Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. *Advances in Computing Research*, 5:429–442, 1989.
- 17 Oded Goldreich. *Modern cryptography, probabilistic proofs and pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, 1999.
- 18 Oded Goldreich. Overview of the doubly-efficient interactive proof systems of RRR. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:102, 2017. URL: <https://eccc.weizmann.ac.il/report/2017/102>.
- 19 Oded Goldreich, Tom Gur, and Ron D. Rothblum. Proofs of proximity for context-free languages and read-once branching programs - (extended abstract). In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 666–677. Springer, 2015. doi:10.1007/978-3-662-47672-7\_54.
- 20 Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Inf. Process. Lett.*, 67(4):205–214, 1998. doi:10.1016/S0020-0190(98)00116-1.
- 21 Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991. doi:10.1145/116825.116852.
- 22 Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1-2):1–53, 2002. doi:10.1007/s00037-002-0169-0.
- 23 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 113–122. ACM, 2008. doi:10.1145/1374376.1374396.
- 24 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. doi:10.1137/0218012.

- 25 Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. In Tim Roughgarden, editor, *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 133–142. ACM, 2015. doi:10.1145/2688073.2688079.
- 26 Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 39:1–39:43. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ITCS.2017.39.
- 27 Yael Tauman Kalai and Ran Raz. Probabilistically checkable arguments. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 143–159. Springer, 2009. doi:10.1007/978-3-642-03356-8\_9.
- 28 Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity - [extended abstract]. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 422–442. Springer, 2015. doi:10.1007/978-3-662-48000-7\_21.
- 29 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. doi:10.1145/146585.146605.
- 30 Omer Reingold, Guy N. Rothblum, and Ron Rothblum. Efficient batch verification for UP. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:22, 2018. URL: <https://eccc.weizmann.ac.il/report/2018/022>.
- 31 Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. 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 49–62. ACM, 2016. doi:10.1145/2897518.2897652.
- 32 Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. 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 793–802. ACM, 2013. doi:10.1145/2488608.2488709.
- 33 Adi Shamir.  $IP = PSPACE$ . *J. ACM*, 39(4):869–877, 1992. doi:10.1145/146585.146609.
- 34 L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.





# A Tight Lower Bound for Entropy Flattening

Yi-Hsiu Chen<sup>1</sup>

School of Engineering and Applied Sciences, Harvard University, USA  
yhsiuchen@g.harvard.edu

Mika Göös<sup>2</sup>

School of Engineering and Applied Sciences, Harvard University, USA  
mika@seas.harvard.edu

Salil P. Vadhan<sup>3</sup>

Computer Science and Applied Mathematics, Harvard University, USA  
salil\_vadhan@harvard.edu

Jiapeng Zhang<sup>4</sup>

University of California San Diego, USA  
jpeng.zhang@gmail.com

---

## Abstract

We study *entropy flattening*: Given a circuit  $\mathcal{C}_X$  implicitly describing an  $n$ -bit source  $X$  (namely,  $X$  is the output of  $\mathcal{C}_X$  on a uniform random input), construct another circuit  $\mathcal{C}_Y$  describing a source  $Y$  such that (1) source  $Y$  is nearly *flat* (uniform on its support), and (2) the Shannon entropy of  $Y$  is monotonically related to that of  $X$ . The standard solution is to have  $\mathcal{C}_Y$  evaluate  $\mathcal{C}_X$  altogether  $\Theta(n^2)$  times on independent inputs and concatenate the results (correctness follows from the asymptotic equipartition property). In this paper, we show that this is optimal among *black-box* constructions: Any circuit  $\mathcal{C}_Y$  for entropy flattening that repeatedly queries  $\mathcal{C}_X$  as an oracle requires  $\Omega(n^2)$  queries.

Entropy flattening is a component used in the constructions of pseudorandom generators and other cryptographic primitives from one-way functions [12, 22, 13, 6, 11, 10, 7, 24]. It is also used in reductions between problems complete for statistical zero-knowledge [19, 23, 4, 25]. The  $\Theta(n^2)$  query complexity is often the main efficiency bottleneck. Our lower bound can be viewed as a step towards proving that the current best construction of pseudorandom generator from arbitrary one-way functions by Vadhan and Zheng (STOC 2012) has optimal efficiency.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Computational complexity and cryptography

**Keywords and phrases** Entropy, One-way function

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.23

**Acknowledgements** S.V. thanks Iftach Haitner, Omer Reingold, and Colin Zheng for many illuminating discussions about this problem in the past.

---

<sup>1</sup> Supported by NSF grant CCF-1749750

<sup>2</sup> Supported by Michael O. Rabin Postdoctoral Fellowship

<sup>3</sup> Supported by NSF grant CCF-1749750

<sup>4</sup> Supported by NSF CCF-1614023



© Yi-Hsiu Chen, Mika Göös, Salil P. Vadhan,  
and Jiapeng Zhang;

licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 23; pp. 23:1–23:28



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**1 Introduction**

A *flat* source  $X$  is a random variable that is uniform on its support; equivalently, its Shannon entropy, min-entropy, and max-entropy are all equal:

$$\begin{aligned} H_{\text{sh}}(X) &= \mathbb{E}_{x \sim X} [\log(1/\Pr[X = x])], \\ H_{\text{min}}(X) &= \min_x \log(1/\Pr[X = x]), \\ H_{\text{max}}(X) &= \log |\text{Supp } X|. \end{aligned}$$

These can be far apart for *non-flat* sources, but we always have  $H_{\text{min}}(X) \leq H_{\text{sh}}(X) \leq H_{\text{max}}(X)$ .

**Entropy flattening.** Entropy flattening is the following task: Given a circuit  $\mathcal{C}_X$  implicitly describing an  $n$ -bit source  $X$  (namely,  $X$  is the output of  $\mathcal{C}_X$  on a uniform random input), efficiently construct another circuit  $\mathcal{C}_Y$  describing a “flattened” version  $Y$  of  $X$ . The goal is to have the output source  $Y$  (or a small statistical modification of it) be such that its min- and max-entropies are *monotonically* related to the Shannon entropy of  $X$ . Concretely, one interesting range of parameters is:

- if two *input* sources  $X$  and  $X'$  exhibit a 1-bit Shannon entropy gap,  $H_{\text{sh}}(X') \geq H_{\text{sh}}(X) + 1$ ,
- then the two respective *output* sources  $Y$  and  $Y'$  must witness  $H_{\text{min}}(Y') \geq H_{\text{max}}(Y) + 1$  (modulo a small modification to  $Y$  and  $Y'$ ).



Entropy flattening is used as an ingredient in constructions of pseudorandom generators and other cryptographic primitives from one-way functions [12, 22, 13, 6, 11, 10, 7, 24] and in reductions between problems complete for (non-interactive) statistical zero-knowledge [19, 23, 4, 25]. See Section 1.2 for a detailed discussion.

**A solution: repeat  $X$ .** The standard strategy for entropy flattening is to construct  $Y$  as the concatenation  $X^q$  of some  $q$  i.i.d. copies of the input source  $X$ . That is, in circuit language,  $\mathcal{C}_Y(x_1, \dots, x_q) = (\mathcal{C}_X(x_1), \dots, \mathcal{C}_X(x_q))$ . The well-known *asymptotic equipartition property* in information theory states that  $X^q$  is  $\epsilon$ -close<sup>5</sup> to having min- and max-entropies closely approximated by  $q \cdot H_{\text{sh}}(X)$ . (It is common to say that  $X^q$  has a certain  $\epsilon$ -smooth min- and max-entropy [21].)

<sup>5</sup> Random variables  $Z_1$  and  $Z_2$  are  $\epsilon$ -close if  $d_{\text{TV}}(Z_1, Z_2) \leq \epsilon$  where  $d_{\text{TV}}(Z_1, Z_2)$  is the usual statistical (or total variation) distance, given by  $d_{\text{TV}}(Z_1, Z_2) = \max_{T \subseteq \mathcal{Z}} |\Pr[Z_1 \in T] - \Pr[Z_2 \in T]|$ .

► **Lemma 1** ([12, 14]). *Let  $X$  be an  $n$ -bit random variable. For any  $q \in \mathbb{N}$  and  $\varepsilon > 0$  there is an  $nq$ -bit random variable  $Y'$  that is  $\varepsilon$ -close to  $X^q$  such that*

$$H_{\min}(Y'), H_{\max}(Y') \in q \cdot H_{\text{sh}}(X) \pm O(n\sqrt{q \log(1/\varepsilon)}).$$

In particular, it suffices to set  $q = \tilde{\Theta}(n^2)$  in order to flatten entropy in the aforementioned interesting range of parameters (1-bit Shannon gap implies at least 1-bit min/max gap). The analysis here is also tight by a reduction to standard anti-concentration results: it is *necessary* to have  $q = \Omega(n^2)$  in order for the construction  $Y = X^q$  to flatten entropy.

## 1.1 Our Result

We show that any *black-box* construction for entropy flattening – that is, a circuit  $\mathcal{C}_Y$  which treats  $\mathcal{C}_X$  as a black-box oracle – requires  $\Omega(n^2)$  oracle queries to  $\mathcal{C}_X$ . This is formalized in Theorem 2 below.

In particular, the simple “repeat- $X$ ” strategy is optimal among all black-box constructions. Besides querying  $\mathcal{C}_X$  on independent inputs, a black-box algorithm has the freedom to perform adaptive queries, and it can produce outputs that are arbitrary functions of its query/answer execution log (rather than merely concatenating the answers). For example, this allows the use of hash functions and randomness extractors, which is indeed useful for variations of the flattening task (e.g., Lemma 4 below).

**Query model.** In our black-box model, the input source is now encoded as the output distribution of an *arbitrary* function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  where  $m = \Theta(n)$  (not necessarily computed by a small circuit); namely, the input source is  $f(U_n)$  where  $U_n$  denotes the uniform distribution over  $n$ -bit strings. We consider oracle algorithms  $A^f$  that have query access to  $f$ . Given an  $n'$ -bit input  $w$  (thought of as a random seed) to  $A^f$ , the algorithm computes by repeatedly querying  $f$  (on query  $x \in \{0, 1\}^n$  it gets to learn  $f(x)$ ), until it finally produces some  $m'$ -bit output string  $A^f(w)$ . We denote by  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  the function computed by  $A^f$ . Thus  $A^f(U_{n'})$  is the output source.

**Inputs/outputs.** Our input sources come from the promise problem *Entropy Approximation* (EA); the circuit version of this problem is complete for the complexity class **NISZK** (non-interactive statistical zero-knowledge), as shown by Goldreich, Sahai, and Vadhan [4]. The EA promise problem is (here  $\tau \in \mathbb{N}$  is a threshold parameter):

- YES input:  $(f, \tau)$  such that  $H_{\text{sh}}(f(U_n)) \geq \tau + 1$ .
- NO input:  $(f, \tau)$  such that  $H_{\text{sh}}(f(U_n)) \leq \tau - 1$ .

The goal of a flattening algorithm  $A^f$  (which also gets  $\tau$  as input, but we suppress this in our notation) is to produce an output distribution that is statistically close to having high min-entropy or low max-entropy depending on whether the input source  $f$  is a YES or a NO instance. We say that  $A^f$  is an  $(\varepsilon, \Delta)$ -*flattening algorithm* if (here  $\kappa = \kappa(\tau)$  is a parameter that  $A^f$  gets to choose):

- If  $(f, \tau)$  is a YES input, then  $A^f(U_{n'})$  is  $\varepsilon$ -close to a distribution  $Z_H$  with  $H_{\min}(Z_H) \geq \kappa + \Delta$ .
- If  $(f, \tau)$  is a NO input, then  $A^f(U_{n'})$  is  $\varepsilon$ -close to a distribution  $Z_L$  with  $H_{\max}(Z_L) \leq \kappa - \Delta$ .

**The result.** Our main result is the following.

► **Theorem 2.** *There exist constants  $\varepsilon, \Delta > 0$  such that every  $(\varepsilon, \Delta)$ -flattening algorithm for  $n$ -bit oracles  $f$  requires  $\Omega(n^2)$  oracle queries.*

In fact, our proof yields an even more fine-grained lower bound. Suppose we allow  $\varepsilon$  and  $\Delta$  to vary subject to  $n/25 \geq \Delta \geq \log(1/\varepsilon)$ . Then our lower bound becomes  $\Omega(n^2 \log(1/\varepsilon))$ , which is tight in both  $n$  and  $\varepsilon$ .

## 1.2 Relevance to Cryptographic Constructions

**Pseudorandom generators from one-way functions.** The use of flattening in complexity-based cryptography originates with the celebrated work of Håstad, Impagliazzo, Levin, and Luby (HILL) [12], which showed how to construct a pseudorandom generator from any one-way function. The first step of their construction is to show how to obtain, from any one-way function, a *pseudoentropy generator*. That is, a polynomial-time computable function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  such that  $f(U_n)$  is computationally indistinguishable from a random variable  $Y$  such that  $H_{\text{sh}}(Y)$  is noticeably higher than  $H_{\text{sh}}(f(U_n))$ . In other words, for some threshold  $\tau_n$  and a nonnegligible gap parameter  $\Delta_n \geq 1/\text{poly}(n)$  it holds that:

1.  $f(U_n)$  is computationally indistinguishable from a random variable  $Y$  with Shannon entropy at least  $\tau_n + \Delta_n$ , and
2.  $f(U_n)$  has Shannon entropy at most  $\tau_n - \Delta_n$ .

Notice that if  $\Delta_n = 1$ , then Condition 2 says that the pair  $(f_n, \tau_n)$  is a NO instance of EA (where  $f_n$  is the restriction of  $f$  to instances of length  $n$ ). On the other hand, Condition 1 says that  $(f_n, \tau_n)$  appears to be a YES instance of EA to computationally bounded algorithms (that get to observe an output of  $f_n$  on a uniformly random input). In the HILL construction, it turns out that  $\Delta_n = \tilde{\Theta}(1/n)$  (rather than  $\Delta_n = 1$ ), corresponding to an appropriate variant of EA.

Given the similarity with EA, it is natural that the next step of the HILL construction is flattening. Specifically evaluating  $f$  on many independent inputs yields a distribution that is close to having low max-entropy yet is computationally indistinguishable from having high min-entropy. Since  $\Delta_n$  is not 1, but rather  $\tilde{\Theta}(1/n)$ , the number of copies needed for flattening becomes  $q = \tilde{O}(n/\Delta_n)^2 = \tilde{\Theta}(n^4)$ .

After flattening, universal hashing (or randomness extraction) is applied to obtain a pseudorandom generator  $G^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$ , where  $G^f(U_{n'})$  is computationally indistinguishable from  $U_{m'}$  (i.e. indistinguishable from min-entropy at least  $m'$ ) yet has max-entropy at most  $n' \leq m' - 1$  (due to having a seed length of  $n'$ ). (This step is a computational analogue of Lemma 4 below.)

As described, the pseudorandom generator  $G^f$  makes  $q = \tilde{\Theta}(n^4)$  queries to the pseudoentropy generator  $f$  and hence to the one-way function. The actual HILL construction is more complex and inefficient, due in part to the fact that the entropy threshold  $\tau_n$  is not known. To deal with the latter issue, they enumerate all  $t = \Theta(n/\Delta_n) = \tilde{\Theta}(n^2)$  possibilities for the threshold  $\tau_n$  to within precision  $\Delta_n$ , construct a pseudorandom generator for each choice, and then combine the generators (which has a further cost in efficiency).

A series of subsequent works [13, 6, 9, 24] improved the efficiency of the HILL construction. The state-of-the-art constructions [9, 24] replace “pseudoentropy” with a more relaxed computational analogue of Shannon entropy (“next-bit pseudoentropy”) and thereby obtain  $\Delta_n = 1$  (or even  $\Delta_n = \log n$ ), reducing the cost of flattening to  $q = \tilde{\Theta}(n/\Delta_n)^2 = \tilde{\Theta}(n^2)$ . In these constructions, the entropy threshold  $\tau_n$  is also known (in fact  $\tau_n = n$ ), but there still is an analogous cost of  $\tilde{\Theta}(n)$  due to the fact that we don’t know how the entropy is spread out among the bits of the output of the next-bit pseudoentropy generator  $f$ .

Overall, with the most efficient constructions to date, the pseudorandom generator makes  $\tilde{\Theta}(n^3)$  queries to the one-way function, of which a  $\tilde{\Theta}(n^2)$  factor is due to flattening. This complexity renders the constructions too inefficient for practice, and thus it is important to know whether a more efficient construction is possible.

**Lower Bounds.** The work of Gennaro, Gertner, Katz, and Trevisan [2] gave the first lower bound on constructing pseudorandom generators from one-way functions. Specifically they proved that any “black-box” construction of a pseudorandom generator  $G^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  from a one-way function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  requires  $\Omega((m' - n') / \log n)$  queries to  $f$ . Thus, many queries are needed to construct a pseudorandom generator with large stretch. However, their lower bound says nothing about the number of queries needed to obtain a pseudorandom generator with small stretch (i.e., where  $m' = n' + O(\log n)$ ), and indeed it applies even to one-way permutations  $f$ , where no flattening is needed and a pseudorandom generator with small stretch can be obtained with a single query to the one-way function [3].

For constructing pseudorandom generators with small stretch from one-way functions, Holenstein and Sinha [15] proved that any black-box construction requires  $\tilde{\Omega}(n)$  queries. Their lower bound also does not tell us about flattening, as it applies even to *regular* one-way functions, which directly (with one query) give a separation between pseudo-*min*-entropy and max-entropy. Rather, their lower bound corresponds to the efficiency costs coming from not knowing the entropy thresholds  $\tau_n$  mentioned above (or how the entropy is spread across the bits in the case of next-bit pseudoentropy).

Our lower bound for flattening (Theorem 2) can be viewed as a first-step towards proving that any black-box construction of pseudorandom generators from one-way functions requires  $\tilde{\Omega}(n^2)$  queries. One might hope to also combine this with [15] and obtain a lower bound of  $\tilde{\Omega}(n^3)$  queries, which would match the best-known construction of [24].

**Seed length.** Another important and well-studied efficiency criterion for pseudorandom generator constructions is how the seed length  $n'$  of the pseudorandom generator  $G^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  depends on the input length  $n$  of the one-way function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ . The standard method for flattening (Lemma 1) requires independent samples from the distribution being flattened, and thus the query complexity of flattening contributes a multiplicative factor to the seed length of the pseudorandom generator. For example, the construction of [24] gives a pseudorandom generator with seed length  $\tilde{\Theta}(n^2) \cdot n = \tilde{\Theta}(n^3)$ , as  $\tilde{\Theta}(n^2)$  independent evaluations of the one-way function (or corresponding pseudoentropy generator) are used for flattening. An interesting open problem is to show that independent evaluations are indeed necessary, and extend our lower bound on query complexity to a lower bound on the input length  $n'$  of the flattening algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$ . This could be a first step towards proving a superlinear lower bound on the seed length of pseudorandom generators constructed (in a black-box way) from one-way functions, a long-standing open problem. We note that the existing lower bounds on query complexity of [2, 15] cannot be turned into seed length lower bounds, as there are constructions of large-stretch pseudorandom generators from regular one-way functions with seed length  $\tilde{O}(n)$  [6]. That is, although those constructions make polynomially many queries to the one-way functions, the queries are highly correlated (and even adaptive).

**Other Cryptographic Primitives.** Flattening is also an efficiency bottleneck in the constructions of other cryptographic primitives from arbitrary one-way functions, such as universal one-way hash functions [22, 16, 7] and statistically hiding commitment schemes [8, 11]. In both cases, the state-of-the-art constructions begin by constructing a function  $f$  where there is a gap between its output entropy  $H(f(U_n))$  and a computational analogue of Shannon entropy (namely, a form of “inaccessible entropy”). Then flattening is applied, after which some (possibly interactive) hashing techniques are used to obtain the final cryptographic primitive. Again, our lower bound on flattening can be viewed as a first step towards proving an efficiency lower bound on black-box constructions.

We note that there was a very fruitful interplay between this sequence of works on constructions of cryptographic primitives from one-way functions and general results about **SZK** and **NISZK**, with inspirations going in both directions (e.g., [18, 8, 20, 11]). This reinforces the feeling that our lower bound for Flattening the **NISZK**-complete problem EA can help in understanding the aforementioned constructions.

## 2 Proof Overview

Our proof builds on the recent result of Lovett and Zhang [17], who showed that there is no efficient black-box reduction (making polynomially many queries) from EA to its complement, thereby giving evidence that **NISZK** is not closed under complement and hence that **NISZK**  $\neq$  **SZK**. The result of [17] is a qualitative one, whereas here we are concerned with a quantitative question: What is the exact query complexity of flattening? Nevertheless, we use a similar construction of hard instances as [17] and make use of a variation of their key lemma.

### 2.1 Simplification: The SDU Problem

We find it convenient to work with a slightly simplified version of the flattening task, having one fewer parameter to worry about.

► **Definition 3** (Statistical distance from uniform (SDU)). We say an algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  is a  $k$ -SDU algorithm if for all  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ , we have

- If  $(f, \tau)$  is a YES input to EA, then  $A^f(U_{n'})$  is  $2^{-k}$ -close to  $U_{m'}$ .
- If  $(f, \tau)$  is a NO input to EA, then  $|\text{Supp}(A^f(U_{n'}))| \leq 2^{m'-k}$ .

Note that a  $k$ -SDU algorithm is a  $(2^{-k}, k/2)$ -flattening algorithm (with threshold  $\kappa = m' - k/2$ ). Conversely, we can transform any flattening algorithm to a SDU algorithm using hashing similar to [4]:

► **Lemma 4.** *If there exists a  $(\varepsilon, \Delta)$ -flattening algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  for function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  with query complexity  $q$ , then there exists a  $k$ -SDU algorithm  $A^f : \{0, 1\}^{n''} \rightarrow \{0, 1\}^{m''-3k}$  where  $n'' = O(n' + m')$  for function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  with query complexity  $q$  and  $k = \Omega(\min\{\Delta, \log(1/\varepsilon)\})$ . In particular, there exists such a  $k$ -SDU algorithm with query complexity  $O(k \cdot \min\{n, m\}^2)$ .*

► **Remark.** Note that Lemma 2.2 guarantees not only that  $A$  is a  $k$ -SDU algorithm but also that its output length is only  $3k$  bits shorter than its input length. This additional property will be useful in our proof.

Here for our main result (Theorem 2), it suffices to prove an  $\Omega(kn^2)$  query lower bound for any  $k$ -SDU algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  with  $m' = n' - 3k$  and  $k \leq n/25$ .

► **Theorem 5.** *Let  $k \leq n$ . Every  $k$ -SDU algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  for function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  has query complexity  $\Omega(kn^2)$ .*

### 2.2 Hard Instances

We consider two input distributions  $\mathcal{D}_H$  and  $\mathcal{D}_L$  over functions  $f : \{0, 1\}^n \rightarrow \{0, 1\}^{3n}$  such that the entropies of most functions in  $\mathcal{D}_H$  and  $\mathcal{D}_L$  are at least  $\tau + 1$  and at most  $\tau - 1$  (where  $\tau = \Theta(n)$ ), respectively. To sample a function from  $\mathcal{D}_H$ , we randomly partition the domain of  $f$  into many blocks  $B_1, B_2, \dots, B_s$ , each of size  $2^n/s$  where  $s = 2^{3n/4}$ . For each block  $B_i$ ,

- with probability  $1/2 + \Theta(1/n)$  we insert a high-entropy block:  $f|_{B_i}$  will be a uniformly random mapping from  $B_i$  to  $\{0, 1\}^{3n}$ ; and
- with the remaining probability  $1/2 - \Theta(1/n)$ , we insert a low-entropy block: all elements of  $B_i$  are mapped to the same random element of  $\{0, 1\}^{3n}$ .

The distribution  $\mathcal{D}_L$  is the same, except we swap the two  $1/2 \pm \Theta(1/n)$  probabilities.

Note that since the range  $\{0, 1\}^{3n}$  is so much larger than the domain  $\{0, 1\}^n$ , with high probability  $f$  will be injective on the high-entropy blocks and will also have no collisions between different blocks. Under this condition, if we let  $B(x)$  denote the block containing  $x$  (which is determined by  $f(x)$ ) and  $p$  be the fraction of high entropy blocks, we have

$$H_{\text{sh}}(f(U_n)) = H_{\text{sh}}(B(U_n)) + H_{\text{sh}}(f(U_n) | B(U_n)) \quad (1)$$

$$= \log_2 s + p \cdot \log_2 \left( \frac{2^n}{s} \right) + (1-p) \cdot 0 = \frac{3n}{4} + p \cdot \frac{n}{4}. \quad (2)$$

Under  $\mathcal{D}_H$  we have  $p = \frac{1}{2} + \Theta(\frac{1}{n})$  whp, and under  $\mathcal{D}_L$  we have  $p = \frac{1}{2} - \Theta(\frac{1}{n})$  whp, which yields a constant gap in Shannon entropies, as desired.

### 2.3 Basic Intuition – and a Warning!

The first natural instinct – but too naive, we argue – is that since the bias between observing a high-entropy block versus a low-entropy block is only  $\Theta(1/n)$ , an anti-concentration bound should imply that distinguishing the two distributions takes  $\Omega(n^2)$  queries.

This intuition indeed applies to simple bounded-error randomized decision trees (which output just a 1-bit answer). Concretely, suppose for simplicity that our input is just an  $n^2$ -bit string  $x$  (instead of an exponentially large oracle  $f$ ): each bit  $x_i$  represents either a high-entropy block ( $x_i = 1$ ) or a low-entropy block ( $x_i = 0$ ). We are given the following *gap-majority* promise: the relative Hamming weight  $|x|/n^2$  is either  $1/2 + 1/n$  or  $1/2 - 1/n$ . It is a well-known fact that any bounded-error query algorithm needs  $\Omega(n^2)$  queries to distinguish these two cases.

But surprisingly enough, there does exist<sup>6</sup> a flattening/SDU algorithm  $A^x$  that solves the *gap-majority* promise problem with only  $O(n)$  queries! This suggests that any superlinear lower bound must somehow hide from the algorithm the type (high vs. low) of a queried block. Our choice of distributions  $\mathcal{D}_H$  and  $\mathcal{D}_L$  does indeed achieve this: since there are so many blocks, a single run of the algorithm is unlikely to query more than one point in a single block, and the marginal distribution of such a single query is the same in both  $\mathcal{D}_H$  and  $\mathcal{D}_L$ . The more precise way in which we exploit the hidden type of a block is in invoking the main result of [17]: when switching a high-entropy block in an  $f$  to a low-entropy block, the support of an SDU algorithm's output distribution,  $\text{Supp}(A^f(U_{n'}))$ , cannot increase by much.

<sup>6</sup> Consider the following algorithm  $A^x$  on input a random seed  $w$ : query a sequence of random positions  $i$  (according to  $w$ ) until a position with  $x_i = 1$  is found. Output  $A^x(w) = i$ . It is easy to verify that this is an  $(0, \Theta(1/n))$ -flattening algorithm with expected query complexity  $O(1)$ . Repeating the algorithm some  $\Theta(n)$  many times yields an  $(0, \Omega(1))$ -flattening algorithm with expected query complexity  $O(n)$ . Finally, we can make the algorithm abort if any run exceeds the expected query complexity by a large constant factor; this results in an  $(\epsilon, \Omega(1))$ -flattening algorithm of worst-case query complexity  $O(n)$ .

## 2.4 Technical Outline

Recall that  $A^f(U_{n'})$  is almost-uniform when  $f \sim \mathcal{D}_H$  has high entropy. For almost all  $z \in \{0, 1\}^{m'}$ , most of the high-entropy functions  $f$  make the algorithm  $A^f$  output  $z$  (on some random seed):

$$\Pr_{f \sim \mathcal{D}_H} [\exists w \in \{0, 1\}^{n'}, A^f(w) = z] \geq 1 - 2^{-\Omega(k)}. \quad (3)$$

On the other hand, since the support of  $A^f(U_{n'})$  is small when  $f$  has low entropy, there should be many  $z$  such that when we sample  $f$  from  $\mathcal{D}_L$ , with high probability  $A^f(w)$  does not output  $z$ :

$$\Pr_{f \sim \mathcal{D}_L} [\exists w \in \{0, 1\}^{n'}, A^f(w) = z] \leq 2^{-\Omega(k)}. \quad (4)$$

To connect the high-entropy and low-entropy cases, we essentially prove that for many  $z \in \{0, 1\}^{m'}$  and every algorithm  $A$  making  $o(kn^2)$  queries, we have

$$\Pr_{f \sim \mathcal{D}_H} [\exists w \in \{0, 1\}^{n'}, A^f(w) = z] \leq 2^{o(k)} \cdot \Pr_{f \sim \mathcal{D}_L} [\exists w \in \{0, 1\}^{n'}, A^f(w) = z] + O(2^{-k}). \quad (5)$$

As long as there exists  $z$  such that Equation (3), (4) and (5), the combination of those equations contradict inequality (5).

Our inequality (5) is similar to the key lemma of Lovett and Zhang [17] except the inequality is reversed, we have an extra multiplicative factor of  $2^{o(k)}$  and our lemma (necessarily) only applies to algorithms making  $o(kn^2)$  queries (where the [17] lemma applies even to exponentially many queries).

One key step toward the inequality (5) is to reverse the direction of the inequality by the following trick. We name elements of  $\{0, 1\}^{n'}$  as  $w_1, \dots, w_{2^{n'}}$  in some arbitrary fixed order. Then

$$\begin{aligned} & \Pr_f [\exists w \in \{0, 1\}^{n'}, A^f(w) = z] \\ &= \sum_{\ell=1}^{2^{n'}} \Pr_f [A^f(w_\ell) = z \text{ and } \nexists w \in \{w_1, \dots, w_{\ell-1}\}, A^f(w) = z] \\ &= \sum_{\ell=1}^{2^{n'}} \left( 1 - \Pr_f [\exists w \in \{w_1, \dots, w_{\ell-1}\}, A^f(w) = z \mid A^f(w_\ell) = z] \right) \cdot \Pr_f [A^f(w_\ell) = z]. \end{aligned}$$

Having a negative sign, now we wish to relate the probability of

$$\Pr_f [\exists v \in \{w_1, \dots, w_{\ell-1}\}, A^f(w) = z \mid A^f(w_\ell) = z]$$

over  $\mathcal{D}_H$  and  $\mathcal{D}_L$  in the same direction as [17]. It is not a direct application of their lemma due to the fact that the block size is constant in their construction and our probability is conditioned on the event  $A^f(w_\ell) = z$ , but we prove a generalization (Lemma A.1) of their lemma that suffices. In fact, the proof we provide in Appendix B is simpler than the one in [17] and yields better parameters.

Like in [17], instead of considering the event  $\exists w, A^f(w) = z$  in all the probabilities above, we further impose the restriction that  $A^f(w)$  queries each block  $B_i$  of the domain at most once, since this event happens with high probability. Furthermore (unlike [17]), we also restrict to the case that the number of high-entropy block queries is in the range  $q \cdot (1/2 \pm (O(1/n) + O(1/\sqrt{q})))$  out of a total of  $q$  queries, which also occurs with high probability.



### 3 The Hard Distribution

Let  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  be a potential  $k$ -SDU algorithm for functions  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ . Throughout, we will consider a fixed oracle algorithm  $A^f$  with query complexity  $q$ , and will omit the dependency of  $A$  in most notations. For a vector  $\vec{X}$ , we use  $\vec{X}(j)$  to denote the  $j$ -th element of  $\vec{X}$ , and  $X$  means the unordered set  $\{\vec{X}(j) : j \in [|\vec{X}|]\}$ .

It is equivalent to interpret an element  $\{0, 1\}^n$  as an integer in  $[N]$ , since we do not make use of any structure in  $\{0, 1\}^n$ . Under this notation, we are considering the fixed oracle algorithm  $A^f : [N'] \rightarrow [M']$  for functions  $f : [N] \rightarrow [M]$  where  $N' = 2^{n'}$ ,  $M' = 2^{m'}$ ,  $N = 2^n$  and  $M = 2^m$ .

**Partition.** Given parameters  $s, t \in \mathbb{N}$  where  $st = N$ , and a function  $f : [N] \rightarrow [M]$ , we partition the domain  $[N]$  into  $s$  blocks  $X_1, \dots, X_s$  each of size  $t$ . We also fix an order for the blocks:  $\vec{X} = (\vec{X}_1, \dots, \vec{X}_s)$ . Given a vector  $\vec{Y}_i \in [M]^t$ , we use the shorthand  $f(\vec{X}_i) = \vec{Y}_i$  to mean  $f(\vec{X}_i(j)) = \vec{Y}_i(j)$ , for all  $j \in [t]$ . Therefore, once vectors  $\vec{Y}_1, \dots, \vec{Y}_s \in [M]^t$  and a partition  $\vec{X}$  are determined, the function  $f$  is fully defined as  $f(\vec{X}_i) = \vec{Y}_i$  for all  $i \in [s]$ .

#### Distributions.

- Let  $\mathcal{X}_s$  be a uniform distribution over an ordered partitions  $\vec{X} = (\vec{X}_1, \dots, \vec{X}_s)$  of  $[N]$  where  $|\vec{X}_i| = N/s = t$  for all  $i \in [s]$ .
- Let  $\mathcal{Y}_0$  and  $\mathcal{Y}_1$  be distributions on  $[M]^t$  defined as follows,
  - For  $\mathcal{Y}_0$ , uniformly sample a string  $z$ , and output  $\vec{Y}(1) = \dots = \vec{Y}(t) = z$ .
  - For  $\mathcal{Y}_1$ , uniformly and independently sample  $\vec{Y}(1), \dots, \vec{Y}(t)$  from  $[M]$ .
- Given a vector  $\vec{b} \in \{0, 1\}^s$  and a partition  $\vec{X} = (\vec{X}_1, \dots, \vec{X}_s)$  of  $[N]$ , we define the distribution  $\mathcal{F}(\vec{X}, \vec{b})$  of function  $f : [N] \rightarrow [M]$  such that  $f(\vec{X}_i) = \vec{Y}_i$  where  $\vec{Y}_i \leftarrow \mathcal{Y}_{\vec{b}(i)}$ . Essentially,  $\vec{b}$  indicates whether each block is “high entropy” or “low entropy”.
- For  $0 \leq \alpha \leq 1$ , let  $\mathcal{B}_\alpha$  be a distribution over a vector  $\vec{b} \in \{0, 1\}^s$ , so that each entry of  $\vec{b}$  is sampled from  $\text{Bern}(\alpha)$  independently.
- For  $0 \leq \alpha \leq 1$ ,  $\mathcal{D}_\alpha$  is a distribution a function  $f : [N] \rightarrow [M]$ , a partition  $\vec{X}$ , and an indicator vector  $\vec{b}$ :  $(f, \vec{b}, \vec{X}) \sim \mathcal{D}_\alpha$  means  $\vec{b} \sim \mathcal{B}_\alpha$ ,  $\vec{X} \sim \mathcal{X}_s$  and  $f \sim \mathcal{F}(\vec{X}, \vec{b})$ .

**Block-Compatibility.** When an algorithm  $A$  runs with input  $w$  and oracle  $f$ , let  $\text{Query}_f(w)$  be the set of the queries made by the algorithm  $A^f(w)$  to  $f$ . We say  $w$  is *block-compatible* with  $(f, \mathbf{X})$  if  $|\text{Query}_f(w) \cap X| \leq 1$  for all  $X \in \mathbf{X}$ . The set of block-compatible inputs with  $(f, \mathbf{X})$  is denoted

$$\text{BC}(f, \mathbf{X}) = \{w : w \text{ is block-compatible with } (f, \mathbf{X})\}$$

**Construction.** Set  $m = 3n$ , so  $M = N^3$ . Also, set  $s = 2^{3n/4} = N^{3/4}$  and  $t = 2^{n/4} = N^{1/4}$ . Let the high entropy distribution be  $\mathcal{D}_H \stackrel{\text{def}}{=} \mathcal{D}_{1/2+5/n}$  and the low entropy distribution be  $\mathcal{D}_L \stackrel{\text{def}}{=} \mathcal{D}_{1/2+5/n}$ . We claim that with high probability, a function  $f$  from  $\mathcal{D}_H$  and  $\mathcal{D}_L$  has entropy at least  $\tau + 1$  and at most  $\tau - 1$  for  $\tau = 7n/8$ .

► **Lemma 6.** *Let the parameters be as above. Then we have*

$$\begin{aligned} \Pr_{(f, \vec{b}, \vec{X}) \sim \mathcal{D}_H} [\text{H}_{\text{sh}}(f) \geq \tau + 1] &\geq 1 - 2^{-0.9n} && \text{and} \\ \Pr_{(f, \vec{b}, \vec{X}) \sim \mathcal{D}_L} [\text{H}_{\text{sh}}(f) \leq \tau - 1] &\geq 1 - 2^{-0.9n} \end{aligned}$$

## 23:10 A Tight Lower Bound for Entropy Flattening

**Proof.** For any pair of independent and random mappings to  $M$ , the collision probability is  $1/M$ . There are no more than  $N^2$  pairs of inputs, so with probability at least  $1 - N^2/M = 1 - 2^{-n}$ , there is no collision when two images are sampled independently. Under that condition, as shown by Equation (1), let  $p$  be the fraction of high entropy blocks, namely  $p$  is the hamming weight of  $\vec{b}$  divided by  $s$ , the entropy of the function  $f$  is

$$H_{\text{sh}}(f(U_n)) = \frac{3n}{4} + p \cdot \frac{n}{4}.$$

Recall that when we sample  $\vec{b}$  from  $\mathcal{D}_H$ ,  $\vec{b}(i) \sim \text{Bern}(1/2 + 5/n)$  for all  $i \in [s]$ . By the Chernoff bound,

$$\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} \left[ p \geq \frac{1}{2} + \frac{4}{n} \right] \geq 1 - 2^{\frac{1}{4} \cdot s \cdot (1/n)^2},$$

which implies

$$\begin{aligned} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} \left[ H_{\text{sh}}(f) \geq \frac{3n}{4} + \left( \frac{1}{2} + \frac{4}{n} \right) \cdot \frac{n}{4} = \frac{7n}{8} + 1 \right] &\geq 1 - 2^{-\frac{1}{4} \cdot s \cdot (1/n)^2} - 2^{-n} \\ &\geq 1 - 2^{-0.9n}. \end{aligned}$$

Similarly, when sampling from  $\mathcal{D}_L$ ,

$$\begin{aligned} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} \left[ H_{\text{sh}}(f) \leq \frac{3n}{4} + \left( \frac{1}{2} - \frac{4}{n} \right) \cdot \frac{n}{4} = \frac{7n}{8} - 1 \right] &\geq 1 - 2^{-\frac{1}{4} \cdot s \cdot (1/n)^2} - 2^{-n} \\ &\geq 1 - 2^{-0.9n}. \end{aligned}$$

Taking  $\tau = \frac{7n}{8}$  concludes the lemma. ◀

## 4 Query Lower Bound for SDU Algorithms

### 4.1 Proof Strategy

Let  $A^f$  be a  $k$ -SDU algorithm making  $q = o(kn^2)$  queries. We may assume wlog that the algorithm makes exactly  $q$  oracle queries to  $f$ , and all the query positions are distinct. (It is useless to query the same positions, and if the number of queries is less than  $q$ , we simply make some dummy queries.) We derive a contradiction from the following two lemmas to conclude the lower bound (Theorem 5). For every  $z \in [M']$  that satisfies

$$\mathbb{E}_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [|\{w : A^f(w) = z\}|] \leq 2^{4k}, \quad (6)$$

we have

$$\begin{aligned} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \\ \leq 2^{o(k)} \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] + O(2^{-k}) \end{aligned} \quad (7)$$

There exists a universal constant  $c > 0$  such that for every sufficiently large  $n$  and  $25k \leq n$ , there is an output  $z \in [M']$  that satisfies

1.  $\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \geq 1 - 2^{-ck}$ .
2.  $\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \leq 2^{-ck}$ .

$$3. \mathbb{E}_{(f, \vec{b}, \vec{X}) \sim \mathcal{D}_H} [|\{w : A^f(w) = z\}|] \leq 2^{4k}$$

The contradiction directly came from plugging  $z$  that satisfies the inequalities in Lemma 4.1 into Inequality (7).

In the following section, we prove that most inputs are block-compatible and hence we can only consider the block-compatible inputs rather than the whole domain  $[N']$ . Then we prove Lemma 4.1 and 4.1 in Section 4.3 and 4.4, respectively.

## 4.2 Block-Compatible Inputs

As in [17], we only consider block-compatible inputs, where each block is queried at most once. In that case, it is easier to compare the behavior of the SDU algorithms. Since there are exponentially many blocks but only polynomially many queries, intuitively, the probability of having block-compatible property is overwhelming if we randomly partition the domain of  $f$ . Formally,

► **Lemma 7.** *For every  $w \in [N']$  and  $\alpha \in [0, 1]$ ,*

$$\Pr_{(f, \vec{b}, \vec{X}) \sim \mathcal{D}_\alpha} [w \notin \text{BC}(f, \mathbf{X})] \leq \frac{q^2}{s} \leq 2^{-0.6n}.$$

**Proof.** In order to handle adaptive algorithms, we consider the following procedure to sample  $(f, \vec{b}, \vec{X})$ , which is equivalent to sampling from  $\mathcal{D}_\alpha$ . Specifically, we sample the parts that are related to  $w$  first.

### Procedure 4.1

1. Initially,  $\vec{X}_i(j) = *$  and  $\vec{b}(i) = *$  for all  $i \in [s], j \in [t]$ .
2. Simulate  $A^f(w)$  handling the  $r$ -th oracle query  $x_r$  as follows. For  $r = 1, \dots, q$ ,
  - a. Based on previous queries and results as well as  $w$ , let the  $r$ -th query be  $x_r$ . Select  $(i, j)$  uniformly at random from  $[s] \times [t]$  subject to  $\vec{X}_i(j) = *$  and assign  $\vec{X}_i(j) = x_r$ .
  - b. If  $\vec{b}(i) = *$ , then assign  $\vec{b}(i) \sim \text{Bern}(\alpha)$  and  $\vec{Y}_i \sim \mathcal{Y}_{\vec{b}(i)}$ .
  - c. Set  $f(x_r) = Y_i(j)$  and return  $f(x_r)$  as the answer to the query.
3. Assign the rest of the vectors  $\vec{X}$  and  $\vec{b}$  by executing Step 2(a)–2(c) for all  $x \in [N] \setminus \{x_1, \dots, x_q\}$ .

By the principle of deferred decisions, it can be verified that the joint distribution of  $(f, \vec{X}, \vec{b})$  is identical to  $\mathcal{D}_\alpha$ .

Notice that  $w \in \text{BC}(f, \vec{X}, \vec{b})$  if and only if the sequence of  $q$  values of  $i$  selected in Step 2(a) are all distinct. The probability that the  $(r+1)^{\text{st}}$  value of  $i$  is the same one comparing to the previous  $r$  values is at most  $rt/(st-r) \leq q/s$ , since  $r \leq q-1$  and  $qr \leq st$ . So the probability that there are any repetitions is at most  $q^2/s$ . ◀

By Markov's inequality, almost all inputs are block-compatible.

► **Corollary 8.** *For every  $\alpha \in [0, 1]$ ,*

$$\Pr_{(f, \vec{b}, \vec{X}) \sim \mathcal{D}_\alpha} [|\text{BC}(f, \mathbf{X})| > N' \cdot (1 - 2^{-0.3n})] \geq 1 - 2^{-0.3n}$$

### 4.3 Proof of Lemma 4.1

For every  $z \in [M']$  that satisfies

$$\mathbb{E}_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [|\{w : A^f(w) = z\}|] \leq 2^{4k}, \quad (6)$$

we have

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \\ & \leq 2^{o(k)} \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] + O(2^{-k}) \end{aligned} \quad (7)$$

**Proof.** Define the set

$$W_z(f, \mathbf{X}) = \{w : w \in \text{BC}(f, \mathbf{X}), A^f(w) = z\}.$$

Let  $w_1, \dots, w_{N'}$  be all possible inputs in arbitrary but fixed order. The first step is to break the event  $\exists w \in W_z(f, \mathbf{X})$  to the events that  $w_\ell$  is the “first” one in  $W_z(f, \mathbf{X})$  for all  $\ell \in [N']$ ,

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [\exists w \in W_z(f, \mathbf{X})] \\ & = \sum_{\ell=1}^{N'} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_\ell \in W_z(f, \mathbf{X}) \wedge w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X})] \\ & = \sum_{\ell=1}^{N'} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] \\ & \quad \times \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_\ell \in W_z(f, \mathbf{X})] \end{aligned}$$

Our goal is to switch the distribution from  $\mathcal{D}_H$  to  $\mathcal{D}_L$  and see how the probability changes. We do the switch using the following two claims. For every  $w_\ell \in [N']$ ,  $\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_\ell \in W_z(f, \mathbf{X})]$  does not depend on  $\alpha \in [0, 1]$ . In particular,

$$\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [w_\ell \in W_z(f, \mathbf{X})] = \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [w_\ell \in W_z(f, \mathbf{X})].$$

For every  $w_\ell \in [N']$  and  $z \in [M']$ ,

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] \\ & \leq 2^{o(k)} \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] + O\left(\frac{q^3 t^2}{s}\right) + 2^{-5k} \end{aligned}$$

The intuition behind the first claim is that as long as  $w_\ell$  is block-compatible, the query results are independently uniform over  $[M]$  in both  $\mathcal{D}_H$  or  $\mathcal{D}_L$  case. For the second claim, we will apply a variation of the main lemma in [17]. Notice that the direction of the inequality in the second claim is reversed by our first step, and thus is consistent to the one in [17]. The formal proofs of those Claims are shown in Section 4.3.1 and 4.3.2.

Once we have the above claims, we can prove the lemma:

$$\begin{aligned}
& \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\exists w \in W_z(f, \mathbf{X})] \\
& \leq 2^{o(k)} \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in W_z(f, \mathbf{X})] \\
& \quad + \left( O\left(\frac{q^3 t^2}{s}\right) + 2^{-5k} \right) \cdot \sum_{\ell=1}^{2^{n'}} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [w_\ell \in W_z(f, \mathbf{X})] \\
& \leq 2^{o(k)} \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in W_z(f, \mathbf{X})] \\
& \quad + \left( O\left(2^{-n/5}\right) + 2^{-5k} \right) \cdot \mathbb{E}_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\#\{w : A^f(w) = z\}] \\
& \leq 2^{o(k)} \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in W_z(f, \mathbf{X})] + O(2^{-k}).
\end{aligned}$$

The second inequality is by the assumption of  $n > 25k$ , and the last inequality is by Inequality (6). ◀

### 4.3.1 Proof of Claim 4.3

For every  $w_\ell \in [N']$ ,  $\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_\ell \in W_z(f, \mathbf{X})]$  does not depend on  $\alpha \in [0, 1]$ . In particular,

$$\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [w_\ell \in W_z(f, \mathbf{X})] = \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [w_\ell \in W_z(f, \mathbf{X})].$$

**Proof.** We factorize the probability into two parts and prove both of them are independent of  $\alpha$ .

$$\begin{aligned}
& \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_\ell \in W_z(f, \mathbf{X})] \\
& = \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [A^f(w_\ell) = z \mid w_\ell \in \text{BC}(f, \mathbf{X})] \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_\ell \in \text{BC}(f, \mathbf{X})]
\end{aligned}$$

We use Procedure 4.1 to sample  $(f, \vec{b}, \vec{\mathbf{X}})$ . We will prove the second factor is independent of  $\alpha$  by induction over  $r$ . Conditioning on the first  $(r-1)$  values of  $i$  selected in Step 2(a) being all distinct, that is, the block-compatible property has not been violated in the first  $r$  rounds, we have  $\vec{b}(i) = *$  at the beginning of Step 2(b) in the  $r$ -th round. Thus no matter what  $\alpha$  is and what  $\vec{b}(i)$  is assigned,  $Y_i(j)$  is uniform over  $[M]$  in the  $r$ -th round. Therefore, under the assumed condition, the distribution of  $x_r$  and  $f(x_r)$  are independent of  $\alpha$  and the probability of maintaining the block-compatible property in the  $r$ -th round is independent of  $\alpha$ . By induction, we know that the probability of maintaining the block-compatible property in all  $q$  rounds is independent of  $\alpha$ .

For the first factor, as discussed above, conditioning on the block-compatible property, the distributions of  $x_r$  and  $f(x_r)$  are independent of  $\alpha$ , so the probability of getting  $z$  as the output of  $A^f(w_\ell)$  is also independent of  $\alpha$ . ◀

### 4.3.2 Proof of Claim 4.3

For every  $w_\ell \in [N']$  and  $z \in [M']$ ,

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] \\ & \leq 2^{o(k)} \cdot \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] + O\left(\frac{q^3 t^2}{s}\right) + 2^{-5k} \end{aligned}$$

**Proof.** We consider the following sampling procedure which is equivalent to sampling  $(f, \vec{b}, \vec{\mathbf{X}})$  from  $\mathcal{D}_\alpha$  conditioned on  $w_\ell \in W_z(f, \mathbf{X})$  (Namely,  $A^f(w_\ell) = z$  and  $w_\ell \in \text{BC}(f, \mathbf{X})$ ). We denote such a distribution as  $(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha(w_\ell, z)$ . It follows the same idea as in Procedure 4.1 – sampling the blocks that are queried by  $A^f(w_\ell)$  first, and uses the rejection sampling to handle the condition  $w_\ell \in W_z(f, \mathbf{X})$ .

#### Procedure 4.2

1. Initially,  $\vec{X}_i(j) = *$  and  $\vec{b}(i) = *$  for all  $i \in [s], j \in [t]$  and  $f(x) = *$  for all  $x \in [N]$ .
2. Simulate  $A^f(w_\ell)$  handling the  $r$ -th oracle query  $x_r$  as follows. For  $r = 1 \dots, q$ ,
  - a. Based on previous queries and results as well as  $w$ , let the  $r$ -th query be  $x_r$ . Select  $(i, j)$  uniformly at random from  $[s] \times [t]$  subject to  $X_i(j) = *$  and assign  $\vec{X}_i(j) = x_r$ .
  - b. If  $\vec{b}(i) = *$ , then assign  $\vec{b}(i) \sim \text{Bern}(\alpha)$  and  $Y_i \sim \mathcal{Y}_{\vec{b}(i)}$ .
  - c. Set  $f(x_r) = Y_i(j)$  and return  $f(x_r)$  as the answer to the query.
3. If  $q$  values of  $i$  in Step 2(a) are not all distinct, or  $A^f(w_\ell) \neq z$ , **restart**.
4. For all  $(i, j)$  such that  $\vec{b}(i) \neq *$  and  $\vec{X}_i(j) = *$ , randomly sample  $x \in [N]$  that has not been assigned to any partition. Set  $\vec{X}_i(j) = x$  and  $f(x) = Y_i(j)$ .
5. Denote the partially assigned (some of them are mapped to  $*$ ) function and vectors sampled so far as  $f^*, \vec{b}^*, (\vec{\mathbf{X}}^*) \sim \mathcal{D}_\alpha^*(w_\ell, z)$ .
6. Assign the rest of the vectors  $\vec{\mathbf{X}}, \vec{b}$  and the mapping  $f$  by executing Step 2(a)–(c) for all  $x \in [N] \setminus \{x_1, \dots, x_q\}$  (instead of  $x_r$ ).

Notice that until Step 5, information (including the partition  $\vec{\mathbf{X}}^*$ , function mapping  $f^*$  and the indicator  $\vec{b}^*$ ) on exactly  $q$  blocks is decided.

The probability we consider then can be written as

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] \\ & = \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha(w_\ell, z)} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X})] \\ & = \sum_{(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha(w_\ell, z)} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \\ & \quad \times \Pr_{\mathcal{D}_\alpha^*(w_\ell, z)} [(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \end{aligned}$$

Now we introduce a property of a partial indicator. We say a partial indicator is balanced if the number of zeros (low entropy block) and ones (high entropy block) are about the same.

► **Definition 9 (Balance).** Let  $\vec{b}^* \in \{0, 1, *\}^s$  be a “partial” indicator vector where there are  $q$  non-star entries. We say it is *balanced* if the number of 1s is in  $[q \cdot (1/2 - 5/n - \sqrt{25k/q}), q \cdot (1/2 + 5/n + \sqrt{25k/q})]$ .

According to Procedure 4.2, each non-star entry of  $\vec{b}^*$  is sampled uniformly and independently from  $\text{Bern}(\alpha)$ . When  $\alpha \in [1/2 - 5/n, 1/2 + 5/n]$ , by Chernoff bound, we have

$$\Pr_{f^*, \vec{b}^*, (\vec{\mathbf{X}}^*) \sim \mathcal{D}_\alpha^*(w_\ell, z)} \left[ \vec{b}^* \text{ is balanced} \right] \geq 1 - 2^{-5k}.$$

And thus we can sum over only balanced  $\vec{b}^*$  by paying an additive term.

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid w_\ell \in W_z(f, \mathbf{X})] \\ & \leq 2^{-5k} + \sum_{\substack{(f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \\ \text{where } \vec{b}^* \text{ is balanced}}} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha(w_\ell, z)} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \\ & \qquad \qquad \qquad \times \Pr_{\mathcal{D}_\alpha(w_\ell, z)^*} [(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \end{aligned} \quad (8)$$

Now we use the following two claims (proved in the later paragraphs) to connect the high entropy case ( $\mathcal{D}_H$ ) and the low entropy case ( $\mathcal{D}_L$ ) on those two factors.

For every  $w_\ell \in [N']$ ,  $z \in [M']$  and every possible  $(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)$  from  $\mathcal{D}_H^*(w_\ell, z)$ , we have

$$\begin{aligned} & \Pr_{\mathcal{D}_H(w_\ell, z)} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \\ & \leq \Pr_{\mathcal{D}_L(w_\ell, z)} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] + O\left(\frac{q^3 t^2}{s}\right) \end{aligned} \quad (9)$$

For every  $w_\ell \in [N']$ ,  $z \in [M']$  and every  $(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)$  where  $\vec{b}^*$  is balanced,

$$\Pr_{\mathcal{D}_H^*(w_\ell, z)} [(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \leq 2^{o(k)} \cdot \Pr_{\mathcal{D}_L^*(w_\ell, z)} [(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \quad (10)$$

Inserting Inequalities (9) and (10) to Equation (8) with  $\alpha = 1/2 + 5/n$ , we conclude the claim.  $\blacktriangleleft$

**Proof of Claim 4.3.2** For every  $w_\ell \in [N']$ ,  $z \in [M']$  and every possible  $(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)$  from  $\mathcal{D}_H^*(w_\ell, z)$ , we have

$$\begin{aligned} & \Pr_{\mathcal{D}_H(w_\ell, z)} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] \\ & \leq \Pr_{\mathcal{D}_L(w_\ell, z)} [w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*)] + O\left(\frac{q^3 t^2}{s}\right) \end{aligned} \quad (9)$$

**Proof.** We will use a variation of the main lemma (Lemma 3) in [17]. Let  $\hat{A}^{\hat{f}} : [\hat{N}'] \rightarrow [\hat{M}']$  be an algorithm making at most  $q$  oracle queries to  $\hat{f} : [\hat{N}] \rightarrow [\hat{M}]$ . Let  $\hat{\mathcal{D}}_H = \hat{\mathcal{D}}_{1/2+5/n}$  and  $\hat{\mathcal{D}}_L = \hat{\mathcal{D}}_{1/2-5/n}$  be the distribution over a function  $\hat{f} : [\hat{N}] \rightarrow [\hat{M}]$ , a partition  $\hat{\mathbf{X}} \in ([\hat{N}']^{\hat{s}})$ , and the indication vector  $\vec{\hat{b}} \in \{0, 1\}^{\hat{s}}$  as defined in Section 3. If  $\hat{t} > q$ , then for all  $z \in [\hat{N}']$ ,

$$\begin{aligned} & \Pr_{(\hat{f}, \vec{\hat{b}}, \hat{\mathbf{X}}) \sim \hat{\mathcal{D}}_L} [\exists w \in \text{BC}(\hat{f}, \hat{\mathbf{X}}), \hat{A}^{\hat{f}}(w) = z] \\ & \quad - \Pr_{(\hat{f}, \vec{\hat{b}}, \hat{\mathbf{X}}) \sim \hat{\mathcal{D}}_H} [\exists w \in \text{BC}(\hat{f}, \hat{\mathbf{X}}), \hat{A}^{\hat{f}}(w) = z] \leq \frac{O(q^3 \cdot \hat{t}^2)}{\hat{s}} \end{aligned}$$

## 23:16 A Tight Lower Bound for Entropy Flattening

**Proof.** See Appendix A.1. ◀

For a fixed  $(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)$ , apply the above lemma in the following way:

- Let  $\hat{s} = s - q$ ,  $\hat{t} = t$ , and so  $\hat{N} = \hat{s} \cdot \hat{t} = N - qt$ .
- Let  $S = \{x \mid f^*(x) = *\} \subseteq [N]$ ,  $I = \{i \mid \vec{b}^*(i) = *\} \subseteq [s]$  and  $\pi_X : S \rightarrow [\hat{N}]$ ,  $\pi_I : I \rightarrow [\hat{s}]$  be arbitrary bijection mappings. Then we define  $\hat{f}$ ,  $\vec{\hat{\mathbf{X}}}$  and  $\vec{\hat{b}}$  as follows.

$$\begin{cases} \forall \hat{x} \in [\hat{N}] & , \hat{f}(\hat{x}) \stackrel{\text{def}}{=} f(\pi_X^{-1}(\hat{x})) \\ \forall (\hat{i}, \hat{j}) \in [\hat{s}] \times [\hat{t}] & , \vec{\hat{\mathbf{X}}}_{\hat{i}}(\hat{j}) \stackrel{\text{def}}{=} \pi_X(\vec{\mathbf{X}}_{\pi_I^{-1}(\hat{i})}(\hat{j})) . \\ \forall \hat{i} \in [\hat{s}] & , \vec{\hat{b}}(\hat{i}) \stackrel{\text{def}}{=} \vec{b}(\pi_I^{-1}(\hat{i})) \end{cases}$$

- For  $\hat{w} \in [\hat{N}]$ , define  $\hat{A}^{\hat{f}}(\hat{w})$  to simulate  $A^{\hat{f}}(w)$  and  $w \in \{w_1, \dots, w_{\ell-1}\}$  in the following way. It first check that if  $w \notin \{w_1, \dots, w_{\ell-1}\}$ , output something not equal to  $z$ . Otherwise simulate  $A^{\hat{f}}(w)$  and when  $A$  makes a query  $x \in \mathbf{X}^*$ ,  $\hat{A}$  hardwire the result  $f(x)$  as the answer. When  $x \in S$ , return  $\hat{f}(\pi_X(x))$  as the answer.

By the above mapping, we have

$$\begin{aligned} \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_\alpha(w_\ell, z)} \left[ \exists w \in \text{BC}(f, \mathbf{X}) \cap \{w_1, \dots, w_{\ell-1}\}, A^f(w) = z \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right] \\ = \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \hat{\mathcal{D}}_\alpha} \left[ \exists w \in \text{BC}(\hat{f}, \vec{\hat{\mathbf{X}}}), \hat{A}^{\hat{f}}(w) = z \right]. \end{aligned}$$

By Lemma 4.3.2,

$$\begin{aligned} & \Pr_{\mathcal{D}_H(w_\ell, z)} \left[ w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right] \\ &= 1 - \Pr_{\mathcal{D}_H(w_\ell, z)} \left[ \exists w \in \text{BC}(f, \mathbf{X}) \cap \{w_1, \dots, w_{\ell-1}\}, A^f(w) = z \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right] \\ &= 1 - \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \hat{\mathcal{D}}_H} \left[ \exists w \in \text{BC}(\hat{f}, \vec{\hat{\mathbf{X}}}), \hat{A}^{\hat{f}}(w) = z \right] \\ &\leq 1 - \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \hat{\mathcal{D}}_L} \left[ \exists w \in \text{BC}(\hat{f}, \vec{\hat{\mathbf{X}}}), \hat{A}^{\hat{f}}(w) = z \right] + O\left(\frac{q^3 \cdot \hat{t}^2}{\hat{s}}\right) \\ &= 1 - \Pr_{\mathcal{D}_L(w_\ell, z)} \left[ \exists w \in \text{BC}(f, \mathbf{X}) \cap \{w_1, \dots, w_{\ell-1}\}, A^f(w) = z \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right] + O\left(\frac{q^3 t^2}{s}\right) \\ &= \Pr_{\mathcal{D}_L(w_\ell, z)} \left[ w_1, \dots, w_{\ell-1} \notin W_z(f, \mathbf{X}) \mid (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right] + O\left(\frac{q^3 t^2}{s}\right). \quad \blacktriangleleft \end{aligned}$$

**Proof of Claim 4.3.2** For every  $w_\ell \in [N']$ ,  $z \in [M']$  and every  $(f^*, \vec{b}^*, \vec{\mathbf{X}}^*)$  where  $\vec{b}^*$  is balanced,

$$\Pr_{\mathcal{D}_H^*(w_\ell, z)} \left[ (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right] \leq 2^{o(k)} \cdot \Pr_{\mathcal{D}_L^*(w_\ell, z)} \left[ (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right] \quad (10)$$

**Proof.** The only difference between  $\mathcal{D}_L(w_\ell, z)$  and  $\mathcal{D}_H(w_\ell, z)$  is when sampling  $\vec{b}^*$ . Recall that a balanced partial indicator means the hamming weight is within the range  $q \cdot \left(1/2 \pm \left(1/n + \sqrt{25k/q}\right)\right)$ . Since we only consider the cases where  $\vec{b}^*$  is balanced, the



ratio can be bounded as follows.

$$\begin{aligned}
\frac{\Pr_{\mathcal{D}_H^*(w_\ell, z)} \left[ (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right]}{\Pr_{\mathcal{D}_L^*(w_\ell, z)} \left[ (f^*, \vec{b}^*, \vec{\mathbf{X}}^*) \right]} &\leq \left( \frac{\frac{1}{2} + \frac{5}{n}}{\frac{1}{2} - \frac{5}{n}} \right)^{q \left( \frac{1}{2} + \left( \frac{1}{n} + \sqrt{\frac{25k}{q}} \right) \right)} \left( \frac{\frac{1}{2} - \frac{5}{n}}{\frac{1}{2} + \frac{5}{n}} \right)^{q \left( \frac{1}{2} - \left( \frac{1}{n} + \sqrt{\frac{25k}{q}} \right) \right)} \\
&\leq \left( 1 + \frac{10}{n} \right)^{2q \left( \frac{1}{n} + \sqrt{\frac{25k}{q}} \right)} \left( 1 - \frac{10}{n} \right)^{-2q \left( \frac{1}{n} + \sqrt{\frac{25k}{q}} \right)} \\
&\leq 2^{O \left( \frac{q}{n^2} + \sqrt{\frac{kq}{n^2}} \right)} \leq 2^{o(k)}
\end{aligned} \tag{11}$$

◀

#### 4.4 Proof of Lemma 4.1

There exists a universal constant  $c > 0$  such that for every sufficiently large  $n$  and  $25k \leq n$ , there is an output  $z \in [M']$  that satisfies

1.  $\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \geq 1 - 2^{-ck}$ .
2.  $\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \leq 2^{-ck}$ .
3.  $\mathbb{E}_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [|\{w : A^f(w) = z\}|] \leq 2^{4k}$

**Proof.** In this proof, we abuse notation by denoting  $\text{BC}(f, \mathbf{X})$  also to be the uniform distribution over the set  $\text{BC}(f, \mathbf{X})$ . We will show that that for a random  $z$  sampled from  $[M']$ , it satisfies each property with probability at least  $1 - 2^{-\Omega(k)}$ , and hence by the union bound, it satisfies all three properties with probability at least  $1 - 2^{-\Omega(k)}$ . In particular, there exists  $z \in [M']$  satisfying all three conditions simultaneously.

1.

$$\begin{aligned}
\Pr_{z \sim \{0,1\}^{m'}} [z \notin A^f(\text{BC}(f, \mathbf{X}))] &= 1 - \frac{|\text{Supp}(A^f(\text{BC}(f, \mathbf{X})))|}{[M']} \\
&\leq d_{\text{TV}}(A^f(\text{BC}(f, \mathbf{X})), U_{m'}) \\
&\leq d_{\text{TV}}(A^f(U_{n'}), U_{m'}) + d_{\text{TV}}(\text{BC}(f, \mathbf{X}), U_{m'}) \\
&= d_{\text{TV}}(A^f(U_{n'}), U_{m'}) + 1 - \frac{|\text{BC}(f, X)|}{[N']}
\end{aligned} \tag{12}$$

Take the expectation over  $(f, \vec{b}, \vec{\mathbf{X}})$  from  $\mathcal{D}_H$  for Equation (12). By Lemma 6, Definition 3 and Corollary 8 we have

$$\begin{aligned}
\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H, z \sim [M']} [z \notin A^f(\text{BC}(f, \mathbf{X}))] \\
&\leq \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\text{H}_{\text{sh}}(f) < \tau + 1] + 2^{-k} + 2^{-0.3n} \\
&\leq 2^{-0.9n} + 2^{-k} + 2^{-0.3n} \leq 2^{-0.2k}
\end{aligned}$$

By the Markov inequality,

$$\Pr_{z \in [M']} \left[ \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \geq 1 - 2^{-0.1k} \right] \geq 1 - 2^{-0.1k}.$$

2. By Lemma 6 and Definition 3, we have

$$\begin{aligned}
& \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L, z \sim [M']} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \\
& \leq \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L, z \sim [M']} [\exists w \in [N'], A^f(w) = z] \\
& \leq \Pr_{z \sim [M']} [\exists w \in [N'], A^f(w) = z \mid \text{H}_{\text{sh}}(f) \leq \tau - 1] + \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\text{H}_{\text{sh}}(f) > \tau - 1] \\
& \leq 2^{-k} + 2^{-0.9n} \leq 2^{-0.8k}.
\end{aligned}$$

By the Markov inequality,

$$\Pr_{z \in [M']} \left[ \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \leq 2^{-0.1k} \right] \geq 1 - 2^{-0.7k}.$$

3. Since  $m' = n' + 3k$ ,

$$\mathbb{E}_{z \in [M']} [|\{w : A^f(w) = z\}|] = \Pr_{z \in [M']} \left[ \sum_{w \in [N']} I(A^f(w) = z) \right] = 2^{n'} \cdot 2^{-m'} = 2^{3k}.$$

In particular,

$$\mathbb{E}_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H, z \in [M']} [|\{w : A^f(w) = z\}|] = 2^{3k}.$$

By the Markov inequality,

$$\Pr_{z \in [M']} \left[ \mathbb{E}_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [|\{w : A^f(w) = z\}|] \leq 2^{4k} \right] \geq 1 - 2^{-k}. \quad \blacktriangleleft$$

---

## References

- 1 Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008. doi:10.1137/060651380.
- 2 Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005. doi:10.1137/S0097539704443276.
- 3 Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 25–32. ACM, 1989. doi:10.1145/73007.73010.
- 4 Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Can statistical zero knowledge be made non-interactive? or on the relationship of SZK and NISZK. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 467–484. Springer, 1999. doi:10.1007/3-540-48405-1\_30.
- 5 Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Can statistical zero knowledge be made non-interactive? or on the relationship of SZK and NISZK. *Electronic Colloquium on Computational Complexity (ECCC)*, 6(13), 1999. URL: <http://eccc.hpi-web.de/eccc-reports/1999/TR99-013/index.html>.

- 6 Iftach Haitner, Danny Harnik, and Omer Reingold. Efficient pseudorandom generators from exponentially hard one-way functions. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 228–239. Springer, 2006. doi:10.1007/11787006\_20.
- 7 Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 616–637. Springer, 2010. doi:10.1007/978-3-642-13190-5\_31.
- 8 Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 39(3):1153–1218, 2009. doi:10.1137/080725404.
- 9 Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 437–446. ACM, 2010. doi:10.1145/1806689.1806750.
- 10 Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM J. Comput.*, 42(3):1405–1430, 2013. doi:10.1137/100814421.
- 11 Iftach Haitner, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Inaccessible entropy. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 611–620. ACM, 2009. doi:10.1145/1536414.1536497.
- 12 Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. doi:10.1137/S0097539793244708.
- 13 Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2006. doi:10.1007/11681878\_23.
- 14 Thomas Holenstein and Renato Renner. On the randomness of independent experiments. *IEEE Trans. Information Theory*, 57(4):1865–1871, 2011. doi:10.1109/TIT.2011.2110230.
- 15 Thomas Holenstein and Makrand Sinha. Constructing a pseudorandom generator requires an almost linear number of calls. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 698–707. IEEE, 2012.
- 16 Jonathan Katz and Chiu-Yuen Koo. On constructing universal one-way hash functions from arbitrary one-way functions. *IACR Cryptology ePrint Archive*, 2005:328, 2005. URL: <http://eprint.iacr.org/2005/328>.
- 17 Shachar Lovett and Jiapeng Zhang. On the impossibility of entropy reversal, and its application to zero-knowledge proofs. In *Theory of Cryptography Conference*, pages 31–55. Springer, 2017.
- 18 Minh-Huyen Nguyen and Salil P. Vadhan. Zero knowledge with efficient provers. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on The-*

- ory of Computing, Seattle, WA, USA, May 21-23, 2006, pages 287–295. ACM, 2006. doi:10.1145/1132516.1132559.
- 19 Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000. doi:10.1006/jcss.1999.1664.
  - 20 Shien Jin Ong and Salil P. Vadhan. An equivalence between zero knowledge and commitments. In Ran Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, volume 4948 of *Lecture Notes in Computer Science*, pages 482–500. Springer, 2008. doi:10.1007/978-3-540-78524-8\_27.
  - 21 Renato Renner and Stefan Wolf. Smooth rényi entropy and applications. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 233. IEEE, 2004.
  - 22 John Rompel. One-way functions are necessary and sufficient for secure signatures. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 387–394. ACM, 1990. doi:10.1145/100216.100269.
  - 23 Amit Sahai and Salil P. Vadhan. A complete promise problem for statistical zero-knowledge. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 448–457. IEEE Computer Society, 1997. doi:10.1109/SFCS.1997.646133.
  - 24 Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudo-random generator constructions. 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 817–836. ACM, 2012. doi:10.1145/2213977.2214051.
  - 25 Salil Pravin Vadhan. *A study of statistical zero-knowledge proofs*. PhD thesis, Citeseer, 1999.

## A Missing Proofs

### A.1 Proof of Lemma 4.3.2

We restate the lemma as follows. Note that it is not necessarily the case that  $N$  is a power of two (similarly for  $M, N'$  and  $M'$ ).

Let  $A^f : [N'] \rightarrow [M']$  be an algorithm making at most  $q$  oracle queries to  $f : [N] \rightarrow [M]$ . Let  $\mathcal{D}_H = \mathcal{D}_{1/2+5/n}$  and  $\mathcal{D}_L = \mathcal{D}_{1/2-5/n}$  be the distribution over a function  $f : [N] \rightarrow [M]$ , a partition  $\vec{\mathbf{X}} \in ([N]^t)^s$ , and the indication vector  $\vec{b} \in \{0, 1\}^s$  as defined in Section 3. If  $t > q$ , then for all  $z \in [N]$ ,

$$\Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_L} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] - \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_H} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \leq \frac{O(q^3 t^2)}{s}.$$

Besides the parameters difference, a key difference between Lemma A.1 and the key lemma in [17] is that in our construction, the indicator vectors  $b$  consist of  $s$  independent Bernoulli random variables, while in their case, the number of ones, namely the Hamming weight is fixed. Formally, they consider the following distribution.

► **Definition 10.** For  $i \in [s]$ ,  $\tilde{\mathcal{D}}_i$  is a distribution over the function  $f : [N] \rightarrow [M]$  and a partition  $\vec{\mathbf{X}}$ . Define  $\vec{b}_i = (\underbrace{1, \dots, 1}_i, \underbrace{0, \dots, 0}_{s-i})$ . Then  $(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_i$  denotes that  $\vec{\mathbf{X}} \sim \mathcal{X}_s$  and

$$f \sim \mathcal{F}(\vec{\mathbf{X}}, \vec{b}_i).$$

A direct generalization of the key lemma in [17] can be stated using our notation: Let  $A^f : [N'] \rightarrow [M']$  be an algorithm, which makes at most  $q$  queries to its oracle  $f : [N] \rightarrow [M]$ . If  $t > q$ , then for all  $z \in \{0, 1\}^{m'}$  and  $i \in [s]$ ,

$$\Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_{i-1}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] - \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_i} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \leq \frac{O(q^3 t^2)}{i^2}.$$

We provide a simpler proof of Lemma A.1 in Appendix B. Now we prove Lemma A.1 using Lemma A.1.

**Proof of Lemma A.1.** By telescoping over  $i$  in Lemma A.1, we get that for  $\frac{1}{4} \leq \alpha < \beta \leq 1$  where  $\alpha s$  and  $\beta s$  are integers, we have

$$\begin{aligned} & \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_{\alpha s}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \\ & - \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_{\beta s}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \leq \frac{O(q^3 t^2 (\beta - \alpha))}{s}. \end{aligned}$$

Conditioning on the Hamming weight of  $\vec{b}$  being  $\alpha s$  when we sample  $\mathcal{D}_{1/2-4/n}$  or  $\mathcal{D}_{1/2+4/n}$ , the probability of the event  $\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z$  is same to sampling from  $\tilde{\mathcal{D}}_{\alpha s}$ , because this event is invariant to permuting the indices of the  $s$  blocks, so the vector  $\vec{b} = (\underbrace{1, \dots, 1}_{\alpha s}, \underbrace{0, \dots, 0}_{s-\alpha s})$  is equivalent to any other vector of the same Hamming weight. Hence, we have

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_{1/2 \pm 4/n}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \\ & = \sum_{h=0}^s \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_h} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \cdot \Pr[\text{Bin}(s, 1/2 \pm 4/n) = h], \end{aligned}$$

where Bin is the binomial distribution. By the Chernoff bound,

$$\begin{aligned} & \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_{1/2-4/n}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \\ & - \Pr_{(f, \vec{b}, \vec{\mathbf{X}}) \sim \mathcal{D}_{1/2+4/n}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \\ & \leq 2^{-\Omega(s)} + \sum_{s/4 < h < 3s/4} \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_h} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \cdot \Pr[\text{Bin}(s, 1/2 + 4/n) = h] \\ & - \sum_{s/4 < h < 3s/4} \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_h} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \cdot \Pr[\text{Bin}(s, 1/2 - 4/n) = h] \end{aligned}$$

Then by symmetry ( $\Pr[\text{Bin}(s, p) = h] = \Pr[\text{Bin}(s, 1-p) = s-h]$ ) and the bound we got at the beginning by telescoping, the difference is bounded by

$$\begin{aligned} & \sum_{s/4 < h < 3s/4} \left( \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_h} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \right. \\ & \quad \left. - \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_{s-h}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \right) \\ & \quad \times \Pr[\text{Bin}(s, 1/2 - 4/n) = h] + 2^{-\Omega(s)} \\ & \leq \frac{O(q^3 t^2)}{s}. \end{aligned} \quad \blacktriangleleft$$

## A.2 Proof of Lemma 4

► **Claim 1.** *If there exists a  $(\varepsilon, \Delta)$ -flattening algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  for function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  with query complexity  $q$ , then there exists a  $k$ -SDU algorithm  $B^f : \{0, 1\}^{n''} \rightarrow \{0, 1\}^{m''}$  where  $n'' = O(n' + m')$  and  $m'' = O(n' + m')$  for function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  with query complexity  $q$  and  $k = \Omega(\min\{\Delta, \log(1/\varepsilon)\})$ .*

► **Claim 2.** *If there exists a  $k$ -SDU algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$  for function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  with query complexity  $q$ , then there exists an  $(k-1)$ -SDU algorithm  $B^f : \{0, 1\}^{n''} \rightarrow \{0, 1\}^{m''}$  where  $n'' = O(n')$  and  $m'' = n'' - 3k$  for function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$  with query complexity  $q$ .*

**Proof of Claim 1.** This proof mostly follows the idea in [5]. It suffices to prove the existence of  $\Omega(k)$ -SDU algorithm for  $k = \min\{\Delta, \log(1/\varepsilon)\}$ . Let  $\mathcal{H}_{a,b}$  be a family of 2-universal hash function from  $a$  bits to  $b$  bits. We sample hash functions  $h_1$  and  $h_2$  from  $\mathcal{H}_{m', \kappa}$  and  $\sim \mathcal{H}_{n', n' - \kappa - k/3}$ , respectively, where  $\kappa$  is the parameter chosen by the flattening algorithm  $A^f$ . We will show that

$$B^f(w, h_1, h_2) = (h_1, h_1(A^f(w)), h_2, h_2(w))$$

is a  $\Omega(k)$ -SDU algorithm. We denote the output of  $B^f(w, h_1, h_2)$  as a jointly distributed random variables  $(H_1, Z_1, H_2, Z_2)$  when  $w \sim U_{n'}$ ,  $h_1 \sim \mathcal{H}_{m', \kappa}$  and  $h_2 \sim \mathcal{H}_{n', n' - \kappa - k/3}$ .

1. When  $(f, \tau) \in \text{EA}_Y$ , there exists a distribution  $Z_H$  with  $\mathbf{H}_{\min}(Z_H) \geq \kappa + \Delta$  such that  $d_{\text{TV}}(A^f(U_{n'}), Z_H) \leq \varepsilon$ . First, we show that  $(H_1, Z_1)$  is close to uniform. By the Leftover Hash Lemma,  $d_{\text{TV}}((H_1, H_1(Z_H)), (H_1, U_\kappa)) \leq 2^{-\Delta/3}$ , and so

$$\begin{aligned} d_{\text{TV}}((H_1, Z_1), (H_1, U_\kappa)) &\leq d_{\text{TV}}(A^f(U_{n'}), Z_H) + d_{\text{TV}}((H_1, H_1(Z_H)), (H_1, U_\kappa)) \\ &\leq 2^{-\Delta/3} + \varepsilon \leq 2^{-\Omega(k)}. \end{aligned}$$

For the  $(H_2, Z_2)$  of part, we will show that with high probability over sampling  $(h_1, z_1)$  from  $(H_1, Z_1)$ , the distribution  $(H_2, Z_2)$  conditioned on  $(h_1, z_1)$  is close to uniform. Since  $(H_1, Z_1)$  is  $2^{-\Omega(k)}$ -close to uniform, by the Markov inequality, with probability at least  $1 - 2^{-\Omega(k)}$  over choosing  $(h_1, z_1)$  from  $(H_1, Z_1)$ , we have

$$\Pr[h_1(A^f(U_{n'})) = z_1] = \Pr[Z_1 = z_1 \mid H_1 = h_1] \geq \frac{1}{2} \cdot 2^{-\kappa}.$$

Thus, except for  $2^{-\Omega(k)}$  probability over  $(h_1, z_1)$ , the number of  $w$  such that  $h_1(A^f(w)) = z_1$  is at least  $2^{n' - \kappa - 1}$ . Again, by the Leftover Hash Lemma,  $(H_2, Z_2)$  is  $2^{-\Omega(k)}$ -close to uniform conditioned on any such  $(h_1, z_1)$ . We then can conclude that  $(H_1, Z_1, H_2, Z_2)$  is  $2^{-\Omega(k)}$ -close to uniform.

2. When  $(f, \tau) \in \text{EA}_N$ , there exists a distribution  $Z_L$  with  $\mathbf{H}_{\max}(Z_L) \leq \kappa - \Delta$  such that  $d_{\text{TV}}(A^f(U_{n'}), Z_L) \leq \varepsilon$ . For every fixed  $h_1$  and  $h_2$ , we will bound the support size of  $(Z_1, H_2, Z_2)$  conditioned on  $H_1 = h_1$  and  $H_2 = h_2$ . We divide  $\text{Supp}(Z_1, Z_2)$  into three subset according to  $z_1 \in \text{Supp}(Z_1)$ .

$$\begin{cases} S_1 = \{(z_1, z_2) : z_1 \in \text{Supp}(Z_L)\} \\ S_2 = \{(z_1, z_2) : \Pr[Z_1 = z_1] \geq 2^{-\kappa - 2k/3} \text{ and } z_1 \notin \text{Supp}(Z_L)\} \\ S_3 = \{(z_1, z_2) : \Pr[Z_1 = z_1] < 2^{-\kappa - 2k/3} \text{ and } z_1 \notin \text{Supp}(Z_L)\} \end{cases}$$

Since,  $\text{Supp}(Z_1, Z_2) = S_1 \cup S_2 \cup S_3$ , it suffices to show that

$$|S_i| \leq 2^{-\Omega(k)} \cdot \left| \{0, 1\}^\kappa \times \{0, 1\}^{n' - \kappa - k/3} \right|$$

for all  $i = 1, 2, 3$ .

- a. For  $S_1$ , by definition,  $H_{\max}(Z_L) \leq \kappa - \Delta$  implies that  $|\text{Supp}(Z_L)| / |\{0, 1\}^\kappa| \leq 2^{-\Delta}$ , and so

$$|S_1| \leq 2^{-\Delta} \cdot \left| \{0, 1\}^\kappa \times \{0, 1\}^{n'-\kappa-k/3} \right| \leq 2^{-\Omega(k)} \cdot \left| \{0, 1\}^\kappa \times \{0, 1\}^{n'-\kappa-k/3} \right|.$$

- b. For  $S_2$ , since  $d_{\text{TV}}(A^f(U_{n'}), Z_L) \leq \varepsilon$ ,  $\sum_{z_1 \notin \text{Supp}(Z_L)} \Pr[Z_1 = z_1] \leq \varepsilon$ . Each  $z_1$  such that  $\Pr[Z_1 = z_1] \geq 2^{-\kappa-2k/3}$  contributes at least  $2^{-\kappa-2k/3}$  towards  $\varepsilon$ , so

$$\left| \{z_1 : \Pr[Z_1 = z_1] \geq 2^{-\kappa-2k/3} \text{ and } z_1 \notin \text{Supp}(Z_L) \} \right| \leq \varepsilon \cdot 2^{\kappa+2k/3}.$$

Then we have  $|S_2| \leq 2^{-\Omega(k)} \left| \{0, 1\}^\kappa \times \{0, 1\}^{n'-\kappa-k/3} \right|$ , since  $k \leq \log(1/\varepsilon)$ .

- c. For  $S_3$ , if  $\Pr[Z_1 = z_1] < 2^{-\kappa-2k/3}$ , then the number of  $w \in \{0, 1\}^{n'}$  such that  $h_1(A^f(w)) = z_1$  is at most  $2^{n'-\kappa-2k/3}$ , which is at most a  $2^{-k/3}$  fraction of  $\{0, 1\}^{n'-\kappa-k/3}$ . Therefore,  $|S_3| \leq 2^{-\Omega(k)} \cdot \left| \{0, 1\}^\kappa \times \{0, 1\}^{n'-\kappa-k/3} \right|$ .

Thus, we conclude that  $B^f$  is a  $\Omega(k)$ -SDU algorithm.  $\blacktriangleleft$

### Proof of Claim 2.

► **Definition 11** (average min-entropy [1]). Let  $(X, Y)$  be jointed distributed random variables. The average min-entropy of  $X$  conditioned on  $Z$  is

$$H_{\min}(X|Y) \stackrel{\text{def}}{=} \log \left( \frac{1}{\mathbb{E}_{y \leftarrow Y} [\max_x \Pr[X = x | Y = y]]} \right)$$

► **Lemma 12** (Generalized Leftover Hash Lemma [1]). Let  $(X, Y)$  be a jointed distributed random variables such that  $H_{\min}(X|Y) \geq k$ . Let  $\mathcal{H}_{n,m} = \{h : \{0, 1\}^n \rightarrow \{0, 1\}^m\}$  be a family of universal hash function where  $h$  can be described in  $(n+m)$  bits and  $m = k - 2\log(1/\varepsilon) + 2$ . Then

$$d_{\text{TV}}((h(X), Y, h), (U_m, Y, h)) \leq \varepsilon$$

where  $U_m$  is a uniform  $m$  bits string.

Let  $\mathcal{H}_{n', n'-m'-3k} = \{h : \{0, 1\}^n \rightarrow \{0, 1\}^m\}$  be a family of universal hash function where  $h$  can be described in  $d = 2n' - m' - 3k$  bits. Based on the given  $k$ -SDU algorithm  $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$ , we define the algorithm  $B^f : \{0, 1\}^{n'+d} \rightarrow \{0, 1\}^{n'+d-3k}$  as

$$B^f(w, h) \stackrel{\text{def}}{=} (A^f(w), h(w), h).$$

By the chain rule of average min-entropy ([1, Lemma 2.2b])

$$H_{\min}(w|A(w)) \geq H_{\min}(w) - |A(w)| = n' - m',$$

and hence

$$d_{\text{TV}}((A(w), \text{Ext}(w, v)), (A(w), U_{n'-m'+d-2k-O(1)})) \leq 2^{-k}.$$

Therefore, when  $H_{\text{sh}}(f) \geq \tau + 1$

$$\begin{aligned} & d_{\text{TV}}(B^f(U_{n'+d}), U_{n'+d-3k}) \\ &= d_{\text{TV}}((A^f(w), h(w), h), (U_{m'}, U_{n'-m'+d-3k})) \\ &= d_{\text{TV}}(A^f(w), U_{m'}) + d_{\text{TV}}((A^f(w), h(w), h), (A^f(w), U_{n'-m'+d-3k})) \\ &\leq 2^{-k} + 2^{-k} = 2^{-(k-1)}. \end{aligned}$$

## 23:24 A Tight Lower Bound for Entropy Flattening

The last inequality is by the property of  $k$ -SDU algorithm and Lemma 12.

On the other hand, if  $H_{\text{sh}}(f) \leq \tau - 1$ ,

$$|\text{Supp}(B^f(U_{n'+d}))| \leq 2^{m'-k} \cdot 2^{n'-m'+d-3k} \leq 2^{(n'+d-3k)-k}.$$

Therefore,  $B^f$  is an  $(k-1)$ -SDU algorithm with desired parameter.  $\blacktriangleleft$

### B Proof of Lemma A.1

Let  $A^f : [N'] \rightarrow [M']$  be an algorithm, which makes at most  $q$  queries to its oracle  $f : [N] \rightarrow [M]$ . If  $t > q$ , then for all  $z \in \{0, 1\}^{m'}$  and  $i \in [s]$ ,

$$\Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_{i-1}} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] - \Pr_{(f, \vec{\mathbf{X}}) \sim \tilde{\mathcal{D}}_i} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \leq \frac{O(q^3 t^2)}{i^2}.$$

**Proof.** Distributions  $\tilde{\mathcal{D}}_{i-1}$  and  $\tilde{\mathcal{D}}_i$  differ only on the block  $\vec{X}_i$ . So an equivalent way to sample both distributions is that we can first sample the partition  $\vec{\mathbf{X}}$ , and the mapping except on the set  $X_i$ . In particular, we sample  $\vec{Y}_1, \dots, \vec{Y}_{i-1} \sim \mathcal{Y}_0$  and  $\vec{Y}_{i+1}, \dots, \vec{Y}_s \sim \mathcal{Y}_1$ . After that, for fixed  $\vec{\mathbf{X}}$  and  $\vec{Y}_1, \dots, \vec{Y}_{i-1}, \vec{Y}_{i+1}, \dots, \vec{Y}_s$ , we sample  $\vec{Y}_i$  from  $\mathcal{Y}_1$  or  $\mathcal{Y}_0$  for distribution  $\tilde{\mathcal{D}}_i$  or  $\tilde{\mathcal{D}}_{i-1}$ , respectively.

For notational convenience, we define

$$\begin{aligned} \vec{\mathbf{Y}}_{-i} &\stackrel{\text{def}}{=} (\vec{Y}_1, \dots, \vec{Y}_{i-1}, \vec{Y}_{i+1}, \dots, \vec{Y}_s) \\ \vec{\mathbf{X}}_{-i} &\stackrel{\text{def}}{=} (\vec{X}_1, \dots, \vec{X}_{i-1}, \vec{X}_{i+1}, \dots, \vec{X}_s) \end{aligned}$$

Now the difference of the probabilities can be written as

$$\begin{aligned} \Delta_i &= \mathbb{E}_{\vec{\mathbf{Y}}_{-i}, \vec{\mathbf{X}}, z} \left[ \Pr_{\vec{Y}_i \sim \mathcal{Y}_0} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \right] \\ &\quad - \mathbb{E}_{\vec{\mathbf{Y}}_{-i}, \vec{\mathbf{X}}, z} \left[ \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} [\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z] \right]. \end{aligned} \quad (13)$$

If the block  $\mathbf{X}_i$  is not queried, then the distributions are identical to the adversary. To compare two probabilities better, we refine the event  $\exists w \in \text{BC}(f, \mathbf{X}), A^f(w) = z$  based on the block  $\mathbf{X}_i$ . For given  $f, \vec{\mathbf{X}}$  and  $z$ , we define the following events.

$$\begin{aligned} \forall j \in [t], E_{f, \vec{\mathbf{X}}, z}(j) &\stackrel{\text{def}}{=} \exists w \in \text{BC}(f, \mathbf{X}) \text{ s.t. } A^f(w) = z \wedge \vec{X}_i(j) \in \text{Query}_f(w) \\ E_{f, \vec{\mathbf{X}}, z}(\perp) &\stackrel{\text{def}}{=} \exists w \in \text{BC}(f, \mathbf{X}) \text{ s.t. } A^f(w) = z \wedge \text{Query}_f(w) \cap X_i = \emptyset, \end{aligned}$$

where  $\text{Query}_f(w)$  is the set of the queries made by the algorithm  $A^f(w)$  to the  $f$  with input  $w$ .

The main events that we care about is the union of the above events we defined, so for  $\mathcal{Y} \in \{\mathcal{Y}_0, \mathcal{Y}_1\}$

$$\begin{aligned} \Pr_{\vec{Y}_i \sim \mathcal{Y}} [\exists w \in \text{BC}(f, \mathbf{X})] &= \Pr_{\vec{Y}_i \sim \mathcal{Y}} \left[ E_{f, \vec{\mathbf{X}}, z}(\perp) \vee \left( \bigvee_{j=1}^t E_{f, \vec{\mathbf{X}}, z}(j) \right) \right] \\ &= \Pr_{\vec{Y}_i \sim \mathcal{Y}} [E_{f, \vec{\mathbf{X}}, z}(\perp)] + \Pr_{\vec{Y}_i \sim \mathcal{Y}} \left[ \neg E_{f, \vec{\mathbf{X}}, z}(\perp) \wedge \left( \bigvee_{j=1}^t E_{f, \vec{\mathbf{X}}, z}(j) \right) \right]. \end{aligned}$$



An important observation is that the event  $E_{f, \vec{\mathbf{x}}, z}(\perp)$  does not depend on the  $f(X_i)$ , so sampling  $\vec{Y}_i$  from  $\mathcal{Y}_0$  or  $\mathcal{Y}_1$  does not affect the probability of the event. Hence, Equation (13) can be written as

$$\begin{aligned} \Delta_i = & \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \Pr_{\vec{Y}_i \sim \mathcal{Y}_0} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge \left( \bigvee_{j=1}^t E_{f, \vec{\mathbf{x}}, z}(j) \right) \right] \right] \\ & - \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge \left( \bigvee_{j=1}^t E_{f, \vec{\mathbf{x}}, z}(j) \right) \right] \right]. \end{aligned}$$

Now, for the probability over  $\mathcal{Y}_0$  part, we apply the union bound.

$$\begin{aligned} & \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \Pr_{\vec{Y}_i \sim \mathcal{Y}_0} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge \left( \bigvee_{j=1}^t E_{f, \vec{\mathbf{x}}, z}(j) \right) \right] \right] \\ & \leq \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \sum_{j=1}^t \Pr_{\vec{Y}_i \sim \mathcal{Y}_0} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(j) \right] \right] \end{aligned}$$

For the  $\mathcal{Y}_1$  part, we bound the probability via the inclusion-exclusion principle.

$$\begin{aligned} & \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge \left( \bigvee_{j=1}^t E_{f, \vec{\mathbf{x}}, z}(j) \right) \right] \right] \\ & \geq \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \sum_{j=1}^t \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(j) \right] \right. \\ & \quad \left. - \sum_{j \neq j' \in [t]} \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(j) \wedge E_{f, \vec{\mathbf{x}}, z}(j') \right] \right] \end{aligned}$$

Observe that  $A^f(w)$  only queries  $X_i$  at most once for all  $w \in W(f, \mathbf{X})$ , and the marginal distributions of the mapping on  $\vec{X}_i(j)$  for every  $j \in [t]$  are the same in both  $\mathcal{Y}_1$  and  $\mathcal{Y}_0$  cases, so for every  $j \in [t]$

$$\Pr_{\vec{Y}_i \sim \mathcal{Y}_0} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(j) \right] = \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(j) \right]$$

Therefore, the difference between two cases is bounded as

$$\begin{aligned} \Delta_i & \leq \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \sum_{j \neq j' \in [t]} \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(j) \wedge E_{f, \vec{\mathbf{x}}, z}(j') \right] \right] \\ & \leq t^2 \cdot \mathbb{E}_{\vec{Y}_{-i}, \vec{\mathbf{x}}, z} \left[ \Pr_{\vec{Y}_i \sim \mathcal{Y}_1} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(1) \wedge E_{f, \vec{\mathbf{x}}, z}(2) \right] \right] \\ & = t^2 \cdot \Pr_{(f, \vec{\mathbf{x}}) \sim \tilde{\mathcal{D}}_{i, z}} \left[ \neg E_{f, \vec{\mathbf{x}}, z}(\perp) \wedge E_{f, \vec{\mathbf{x}}, z}(1) \wedge E_{f, \vec{\mathbf{x}}, z}(2) \right]. \end{aligned} \tag{14}$$

To bound the term, we consider another way to sample  $(f, \vec{\mathbf{x}})$  from  $\tilde{\mathcal{D}}_i$ .

**Procedure B.1**

1. Sample  $(f, \vec{\mathbf{X}})$  from  $\mathcal{D}_i$  as usual.
2. Undo the partition for the first  $i$  blocks. That is, set  $X_{i'}(j) = *$  and  $Y_{i'}(j) = *$  for all  $(i', j) \in [i] \times [t]$ .
3. For every  $x \in [N] \setminus \mathbf{X}_{>i}$ , randomly sample  $y$  from  $[M]$  and let  $f(x) = y$ .
4. Randomly partition the unassigned part  $[N] \setminus \mathbf{X}_{>i}$  into  $i$  blocks  $\vec{\mathbf{X}}_1, \dots, \vec{\mathbf{X}}_i$ . Specifically, we will use Procedure B.2 for this step.

Since  $f(x)$  for  $x$  in  $\vec{\mathbf{X}}_{\leq i}$  is randomly and independently chosen from  $[M]$ , the partition among the first  $i$  blocks and  $f(x)$  for  $x$  in the first  $i$  blocks are independent. It is equivalent to sample the partition of the first  $i$  blocks after fixing the function  $f$ .

After the first sampling step, since the mapping is fixed, the set  $\{w : A^f(w) = z\}$  and  $\text{Query}_f(w)$  are determined. Let  $\{w : A^f(w) = z\} = \{w_1, \dots, w_u\}$  and define  $Q_\ell \stackrel{\text{def}}{=} \text{Query}_f(w_\ell) \setminus \mathbf{X}_{>i}$  for all  $\ell \in [u]$ . That is, we only look at queries that belong to the blocks that have not been decided.

Recall the definitions of the events  $E_{f, \vec{\mathbf{X}}, z}(\perp)$ . Its negation means that for all  $w$ ,  $A^f(w) \neq z$  or  $w \notin \text{BC}(f, X)$  or  $\text{Query}_f(w)$  intersect with  $X_i$ . By the definition of  $\{w_1, \dots, w_u\}$ , an equivalent way to describe the event  $\neg E_{f, \vec{\mathbf{X}}, z}(\perp)$  is for all  $\ell \in [u]$ , either  $w_\ell \notin \text{BC}(f, X)$  or  $\text{Query}_f(w_\ell) \cap X_i = 1$ . Note that if the first condition fails, then  $w$  is block-compatible, and so the size of the intersection is at most one. Therefore, the probability factor in Equation 14 can be written as

$$\begin{aligned}
& \Pr_{\vec{\mathbf{X}}_{[i]}} \left[ \neg E_{f, \vec{\mathbf{X}}, z}(\perp) \wedge E_{f, \vec{\mathbf{X}}, z}(1) \wedge E_{f, \vec{\mathbf{X}}, z}(2) \right] \\
&= \Pr_{\vec{\mathbf{X}}_{[i]}} \left[ \left( \forall \ell \in [u], |\text{Query}_f(w_\ell) \cap X_i| = 1 \vee w_\ell \notin \text{BC}(f, \mathbf{X}) \right) \right. \\
&\quad \left. \wedge \left( \exists \ell_1 \neq \ell_2 \in [u], w_{\ell_1}, w_{\ell_2} \in \text{BC}(f, \mathbf{X}), \vec{\mathbf{X}}_i(1) \in Q_{\ell_1}, \vec{\mathbf{X}}_i(2) \in Q_{\ell_2} \right) \right] \quad (15)
\end{aligned}$$

In sum up, the event holds when

- Every  $w \in \{w : A^f(w) = z\}$  is either not block-compatible or it intersects with  $X_i$ .
- Exists two distinct inputs in  $\{w : A^f(w) = z\}$  such that the queries made by  $A^f$  using them as input intersect with  $X_i$  at the first and the second positions.

We consider the following procedure to sample  $\vec{\mathbf{X}}_{[i]}$ . Basically, the procedure will decide the partition without assigning the indices first, which is sufficient for deciding the block-compatibilities. Then we assign the indices to the blocks after putting all elements in  $[N]$  in blocks.

The intuition of the probability bounded by  $\Theta(\text{poly}(q)/i^2)$  (omitting the dependency on  $q$ ) is as follows. In the last step, since we assign the indices to the block randomly. The probability of the  $i$ -th block being the one queried by some block-compatible  $w$  in  $\{w_1, \dots, w_u\}$  is at most  $q/i$ . If all blocks are block-compatible, then the probability of two different  $w$  and  $w'$  in  $\{w_1, \dots, w_u\}$  hitting same block with different queries is  $\Theta(q^2/i)$ . On the other hand, the probability of some  $w$  in  $\{w_1, \dots, w_u\}$  being non-block-compatible is bounded by  $\Theta(1/i)$ . Therefore, no matter what, we have two  $\Theta(1/i)$  factors.

**Procedure B.2**

1. Let  $t_{i'}$  represents the remaining slots in the  $i'$ -th block. Initially,  $t_{i'} = t$  for all  $i' \in [i]$ .
2. Let  $\text{block}(x)$  represents which block  $x$  is assigned to. Initially,  $\text{block}(x) = *$  for all  $x \in \mathbf{X}_{\leq i}$ .
3. Let  $\ell_1 = \ell_2 = -1$  and  $\pi : [i] \rightarrow [i]$  be a permutation will be decided later.
4. For  $\ell = 1, \dots, u$ :
  - a. For all  $x \in Q_\ell$ , if  $\text{block}(x) = *$ , set  $\text{block}(x) = i'$  with probability  $t_{i'}/(t_1 + \dots + t_i)$ , and decrease  $t'_{i'}$  by 1.
  - b. If all  $\text{block}(x)$  for  $x \in Q_\ell$  are distinct, namely  $w_\ell \in \text{BC}(f, \mathbf{X})$ , then let  $\ell_1 = \ell$  and **break**
5. If  $\ell_1 = 1$ .
  - a. First we randomly decide the mapping  $\pi(i)$ . (The rest  $i - 1$  mapping will be decided later).
  - b. If  $\pi(i) \notin \{\text{block}(x) \mid x \in Q_1\}$ , **jump** to Step 7.
  - c. Suppose  $x^* \in Q_1$  such that  $\text{block}(x^*) = \pi(i)$ . We check if there is any  $\ell \in [2, u]$  such that  $x^* \notin Q_\ell$ . If there is no such  $\ell$ , then **jump** to Step 7.
  - d. Let  $\ell \in [2, u]$  such that  $x^* \notin Q_\ell$ . For all  $x \in Q_\ell$ , if  $\text{block}(x) = *$ , set  $\text{block}(x) = i'$  with probability  $t_{i'}/(t_1 + \dots + t_i)$ , and decrease  $t'_{i'}$  by 1. If all  $\text{block}(x)$  for  $x \in Q_\ell$  are distinct, namely  $w_\ell \in \text{BC}(f, \mathbf{X})$ , then let  $\ell_2 = \ell$ .
6. For all  $x \in \mathbf{X}_{\leq i}$ , if  $\text{block}(x) = *$ , set  $\text{block}(x) = i'$  with probability  $t_{i'}/(t_1 + \dots + t_i)$ , and decrease  $t'_{i'}$  by 1.
7. Let  $\pi : [i] \rightarrow [i]$  be a random permutation (Except that if  $\pi(i)$  is decided in Step 6(a)).
8. For  $i' \in [i]$ , let  $\pi_{i'} : [t] \rightarrow [t]$  be a random permutation, assign the  $\pi_{i'}(j)$ -th element of  $\{x \mid \text{block}(x) = \pi(i')\}$  to  $\vec{X}_{i'}(j)$ .

To bound Equation (15), we consider a relaxed event. Event  $E$  happens when for all  $\ell \in [u]$ , either is not block-compatible or  $|Q_\ell \cap X_i| = 1$ . And there exists  $\ell_1 \neq \ell_2 \in [u]$ ,  $w_{\ell_1}, w_{\ell_2} \in \text{BC}(f, X)$  and  $Q_{\ell_1} \cap X_i \neq Q_{\ell_2} \cap X_i$ . That is, we do not require the intersection being the first two elements in  $X_i$

Based on Procedure B.2, we also consider the following events. The subscripts are from the step numbers in the procedure.

$$E_5 : \ell_1 = 1 \text{ that is, } w_1 \in \text{BC}(f, \mathbf{X})$$

$$E_{5c} : |Q_1 \cap X_i| = 1 \text{ and } \exists \ell \in [2, u] \text{ such that } x^* \notin Q_\ell.$$

$$E_{5d} : \ell_2 \neq -1 \text{ that is, } w_\ell \in \text{BC}(f, \mathbf{X})$$

First, we consider the probability of  $w_1$  or  $w_\ell$  being non-compatible (in Step 2(b) or Step 4(d)). The probability that the  $(r+1)$ -th choice of value  $i'$  collides to previous  $r$  choices is at most  $r(t-1)/(ti-r-q) \leq (rt+q)/ti \leq (rt+t)/ti \leq q/i$ . By union bound, the probability of  $w_1$  or  $w_\ell$  being non-compatible is at most  $q^2/i$ . That is,

$$\Pr[\neg E_5] \leq \frac{q^2}{i} \quad \text{and} \quad \Pr[\neg E_{5d} \mid E_{5c}] \leq \frac{q^2}{i}.$$

## 23:28 A Tight Lower Bound for Entropy Flattening

By the above inequalities, we can bound the probability in Equation 15 as follows.

$$\begin{aligned}
& \Pr \left[ \neg E_{f, \bar{\mathbf{x}}, z}(\perp) \wedge E_{f, \bar{\mathbf{x}}, z}(1) \wedge E_{f, \bar{\mathbf{x}}, z}(2) \right] \\
& \leq \Pr[E] = \Pr[E \wedge E_5] + \Pr[E \wedge \neg E_5] \\
& \leq \Pr[E \wedge E_{5c} \mid E_5] + \Pr[\neg E_5] \Pr[E \mid \neg E_5] \\
& \leq \Pr[E_{5c} \mid E_5] \cdot \left( \Pr[E \wedge E_{5d} \mid E_{5c}] + \Pr[E \wedge \neg E_{5d} \mid E_{5c}] \right) + \frac{q^2}{i} \cdot \Pr[E \mid \neg E_5] \\
& \leq \Pr[E_{5c} \mid E_5] \cdot \left( \Pr[E \mid E_{5d} \wedge E_{5c}] + \Pr[\neg E_{5d} \mid E_{5c}] \right) + \frac{q^2}{i} \cdot \Pr[E \mid \neg E_5] \\
& \leq \Pr[E_{5c} \mid E_5] \cdot \left( \Pr[E \mid E_{5d} \wedge E_{5c}] + \frac{q^2}{i} \right) + \frac{q^2}{i} \cdot \Pr[E \mid \neg E_5]
\end{aligned}$$

$\Pr[E_{5c} \mid E_5]$  is at most the probability of  $\pi(i) \in \{\text{block}(x) \mid x \in Q_1\}$  which is at most  $q/i$ . Now we consider the event  $E$  happens when  $E_5$  and  $E_{5d}$  happened. Event  $E$  happens only when  $w_\ell$  in Step 5d hit the  $i$ -th block in different locations. For each  $x \in Q_\ell$ , the probability of it hitting the  $i$ -th block is at most  $(t-1)/(ti-2q) \leq (t+2q)/ti \leq 3/i$ . Applying union bound over at most  $q$  elements in  $Q_\ell$ , we get

$$\Pr[E \mid E_{5d} \wedge E_{5c}] \leq \frac{3q}{i}.$$

For  $\Pr[E \mid \neg E_5]$ , if there is no block compatible  $w_\ell$  for  $\ell \in [u]$ , there is no hope for  $E$  to be satisfied. If there is a block compatible  $w_\ell$ , then  $E$  requires that the  $i$ -th block being hit by  $A^f(w_\ell)$ . Since the the indices of the first  $i$  blocks are randomly assigned by the end, the probability is at most  $q/i$ .

Combine the bounds above, we have

$$\Pr \left[ \neg E_{f, \bar{\mathbf{x}}, z}(\perp) \wedge E_{f, \bar{\mathbf{x}}, z}(1) \wedge E_{f, \bar{\mathbf{x}}, z}(2) \right] \leq \frac{q}{i} \cdot \left( \frac{3q}{i} + \frac{q^2}{i} \right) + \frac{q^2}{i} \cdot \frac{q}{i} = O\left(\frac{q^3}{i^2}\right).$$


Insert the above inequality back to Inequality (14), we have  $\Delta_i \leq O\left(\frac{q^3 t^2}{i^2}\right)$ , which concludes the lemma.  $\blacktriangleleft$

# Worst-Case to Average Case Reductions for the Distance to a Code

Eli Ben-Sasson<sup>1</sup>

Technion, Haifa, Israel

eli@cs.technion.ac.il

 <https://orcid.org/0000-0002-0708-0483>

Swastik Kopparty<sup>2</sup>

Rutgers University, New Brunswick, NJ, USA

swastik.kopparty@gmail.com

Shubhangi Saraf<sup>3</sup>

Rutgers University, New Brunswick, NJ, USA

shubhangi.saraf@gmail.com

---

## Abstract

Algebraic proof systems reduce computational problems to problems about estimating the distance of a sequence of functions  $\vec{u} = (u_1, \dots, u_k)$ , given as oracles, from a linear error correcting code  $V$ . The soundness of such systems relies on methods that act “locally” on  $\vec{u}$  and map it to a single function  $u^*$  that is, roughly, as far from  $V$  as are  $u_1, \dots, u_k$ .

Motivated by these applications to efficient proof systems, we study a natural worst-case to average-case reduction of distance for linear spaces, and show several general cases in which the following statement holds: If some member of a linear space  $U = \text{span}(u_1, \dots, u_k)$  is  $\delta$ -far from (all elements) of  $V$  in relative Hamming distance, then nearly all elements of  $U$  are  $(1 - \epsilon)\delta$ -far from  $V$ ; the value of  $\epsilon$  depends only on the distance of the code  $V$  and approaches 0 as that distance approaches 1. Our results improve on the previous state-of-the-art which showed that nearly all elements of  $U$  are  $\frac{1}{2}\delta$ -far from  $V$  [Rothblum, Vadhan and Wigderson, STOC 2013].

When  $V$  is a Reed-Solomon (RS) code, as is often the case for algebraic proof systems, we show how to *boost* distance via a new “local” transformation that may be useful elsewhere. Relying on the affine-invariance of  $V$ , we map a vector  $u$  to a random linear combination of affine transformations of  $u$ , and show this process amplifies distance from  $V$ . Assuming  $V$  is an RS code with sufficiently large distance, this amplification process converts a function  $u$  that is somewhat far from  $V$  to one that is  $(1 - \epsilon)$ -far from  $V$ ; as above,  $\epsilon$  depends only on the distance of  $V$  and approaches 0 as the distance of  $V$  approaches 1.

We give two concrete application of these techniques. First, we revisit the axis-parallel low-degree test for bivariate polynomials of [Polischuk-Spielman, STOC 1994] and prove a “list-decoding” type result for it, when the degree of one axis is extremely small. This result is similar to the recent list-decoding-regime result of [Chiesa, Manohar and Shinkar, RANDOM 2017] but is proved using different techniques, and allows the degree in one axis to be arbitrarily large. Second, we improve the soundness analysis of the recent RS proximity testing protocol of [Ben-Sasson et al., ICALP 2018] and extend it to the “list-decoding” regime, bringing it closer to the Johnson bound.

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

**Keywords and phrases** Proximity testing, Reed-Solomon codes, algebraic coding complexity

---

<sup>1</sup> Supported by the European Research Council under POC grant OMIP – DLV-693423 and Israel Science Foundation grant 1501/14.

<sup>2</sup> Research supported in part by NSF grants CCF-1253886 and CCF-1540634.

<sup>3</sup> Research supported in part by NSF grants CCF-1350572 and CCF-1540634.

Digital Object Identifier 10.4230/LIPIcs.CCC.2018.24

Funding Work supported by the USA–Israel binational science fund, grant # 2014359

## 1 Introduction

Proof systems that involve *interaction* between a *randomized* verifier and a prover have revolutionized computational complexity and cryptography [7, 14]. A question of paramount importance here is *soundness* – the minimal probability of the verifier rejecting a falsity. Transformations that maintain or increase soundness, while improving other aspects of the proof system (like proof length, or query complexity), are few and hard to obtain. Here, we study certain soundness-preserving techniques for the special case of *linear spaces*, improving on the prior state-of-the-art which was due to Rothblum, Vadhan and Wigderson [19]; see Section 1.2. Then, in Section 1.4, we introduce a soundness-amplifying technique for the special case of Reed-Solomon codes; these codes are used in constructions of efficient proof systems. Before presenting the results we explain their relevance to the general study of proof systems.

### 1.1 Motivation – improving concrete soundness and communication complexity of interactive protocols

*Arithmetization* is a technique that was introduced to the construction of interactive proof (IP) systems by [17], and later applied to other systems including multi-prover interactive proof (MIP) [6], probabilistically checkable proof (PCP) [5, 3, 2] and zero knowledge (ZK) systems [14], to name a few notable examples. Arithmetization refers to a family of *reductions* from languages (like 3SAT) to promise problems involving *algebraic codes* like Reed-Solomon (RS), Reed-Muller (RM), or their generalization to algebraic geometry (AG) codes; all are, in particular, *linear* codes.

An arithmetization reduction maps an instance  $x$  (like a 3SAT formula) to a sequence of algebraic codes  $V_1, \dots, V_k$ , along with a set of “local” constraints, meaning that each constraint depends only on a small number of entries from  $k$  purported codewords. The reduction implies that  $x \in L$  if and only if there exists a sequence  $\vec{u} = (u_1, \dots, u_k) \in V_1 \times \dots \times V_k$  that satisfies all local constraints<sup>4</sup>. The locality of the constraints, along with the distance property of the codes  $V_1, \dots, V_k$  also implies that when  $x \notin L$ , *every* sequence  $\vec{u}$  falsifies a large fraction of local constraints, as long as each member  $u_i$  of the sequence is *sufficiently close* to the code  $V_i$  in relative Hamming distance. Therefore, a major problem in the construction of such proof systems is to build protocols that efficiently ensure each  $u_i$  is in close proximity to  $V_i$ , and reject with non-negligible probability  $s = s(\delta)$  a purported codeword  $u_i$  that is  $\delta$ -far in relative Hamming distance from  $V_i$ . This problem is known as *proximity testing*; the study of the reliance of the soundness parameter  $s$  on the query complexity  $q$  and proximity parameter  $\delta$  is referred to as *soundness analysis*.

Suffice it to say that protocols that solve the proximity testing problem are often a bottleneck in the construction of efficient proof systems, and the quality of their soundness analysis determines concrete efficiency and applicability (see, e.g., [1, 8] for recent instances). Therefore, it is desirable to construct transformations that minimize the number of proximity

<sup>4</sup> The exact nature of these constraints is not relevant to our study here. The interested reader is referred, e.g., to [16, Section 3.1] and [11, Section 5] for examples and more information.

testing problems that are needed to be addressed by a proof system, and boost and maintain the distance of  $\vec{u}$  from  $V_1 \times \dots \times V_k$  when  $x \notin L$ .

Certain proof systems use several instances of the same proximity problem, i.e.,  $V_1 = \dots = V_k = V$  for a single linear code  $V$ . In this case, a natural optimization arises: instead of having the prover and verifier interact to solve  $k$  independent proximity problem, let the verifier sample  $r_1, \dots, r_k \in \mathbb{F}$ , send them to the prover, and then interact to solve the *single* proximity problem that refers to  $\sum_i r_i u_i$ . The cost of an extra round of interaction (and extra randomness) are often well-worth the benefit of reducing the number of proximity testing problems. The linearity of  $C$  implies that this transformation does not harm (perfect) completeness, because when  $\vec{u} \in V$  then  $\Pr[(\sum r_i u_i) \in V] = 1$ .

The more interesting question, discussed next, is to understand what happens to the “typical” distance of  $\sum r_i u_i$  as a function of the maximal distance, defined as  $\delta_{\max} = \max_i \Delta(u_i, V)$ .

## 1.2 Soundness transference results for linear spaces and error correcting codes

Our question is a special case of the “worst-case to average-case” problem: Suppose that a member  $u^*$  of a linear space  $U \subseteq \mathbb{F}^n$  is  $\delta_{\max}$ -far in relative Hamming distance from all members of another linear space  $V \subseteq \mathbb{F}^n$  (this is the “worst-case” assumption), what can be said about the *median*<sup>5</sup> distance  $\delta_{\text{med}}$  from  $V$ , where this median is computed among the members of  $U$ ? We address this question first for the case of  $V$  be a general space, then for  $V$  being an error correcting code.

### 1.2.1 General spaces

The basic question above was first raised by Rothblum, Vadhan and Wigderson, as part of their construction of efficient interactive proofs of proximity (IPPs) [19]. They also showed that nearly all members of  $U$  – all but a  $\frac{1}{|\mathbb{F}|-1}$ -fraction of them – are  $\delta/2$ -far from  $V$  (Lemma 4). Thus,  $\delta_{\text{med}} \geq \delta_{\max}/2$ . On the other hand,  $\delta_{\max} \geq \delta_{\text{med}}$  for certain spaces  $U$  (including all 1-dimensional ones). We are interested in closing the gap between these two bounds.

Our first result (Theorem 7) looks at  $\delta_{\text{med}}$  as a function of  $\delta_{\max}$  and says

$$\delta_{\text{med}}(\delta_{\max}) \geq 1 - \sqrt{1 - \delta_{\max}} - o(1)$$

Here and henceforth,  $o(1)$  denotes negligible terms that approach 0 as  $|\mathbb{F}| \rightarrow \infty$ . In words, the median distance scales roughly like the *Johnson list-decoding function* of  $\delta_{\max}$ , denoted  $J(\delta_{\max})$ , where  $J(x) \triangleq 1 - \sqrt{1 - x}$ . Thus, the median distance  $\delta_{\text{med}}$  is strictly greater than  $\delta_{\max}/2$  for all  $\delta_{\max} > 0$ , and approaches 1 as  $\delta_{\max}$  approaches 1; the prior state-of-the-art approached  $1/2$  in this case. For small values of  $\delta_{\max}$ , our bound approaches  $\delta_{\max}/2$ , as in prior works, but for special (and natural) cases we obtain better bounds on  $\delta_{\text{med}}$ , even when it is arbitrarily small, as discussed next.

<sup>5</sup> All our results hold with high probability, i.e., with respect to the average and 99.9th percentile but we stick to using “median” for simplicity.

### 1.2.2 Linear error correcting codes

Most of the applications to interactive proof systems use a space  $V$  that is an *error correcting code*, i.e., the members of  $V$  are pair-wise far. Letting  $\Delta(V)$  denote the relative distance of  $V$ , our second result (Theorem 9) states

$$\forall \delta_{\max} \leq J(J(\Delta(V)) - o(1), \quad \delta_{\text{med}} \geq \delta_{\max} - o(1).$$

In simple words,  $\delta_{\text{med}} \approx \delta_{\max}$  for sufficiently small values of  $\delta_{\max}$ , where “sufficiently small” depends on  $\Delta(V)$  and approaches 1 as  $\Delta(V) \rightarrow 1$ . Combining Theorems 7 and 9, one sees that for any  $\epsilon > 0$  there exists a code-distance parameter  $\delta_\epsilon$ , such that for every  $V$  with  $\Delta(V) > \delta_\epsilon$  and all spaces  $U$ , we have  $\delta_{\text{med}} \geq (1 - \epsilon)\delta_{\max}$ .

## 1.3 Applications to low-degree testing

We now present two different applications of our results. First, we extend the soundness analysis of the ubiquitous bivariate low-degree test of Polischuck and Spielman to the high-error regime for polynomials that have constant degree in one variable. Then we improve the soundness bounds on the recently suggested “fast RS interactive oracle proof of proximity” (FRI) protocol to beyond the unique-decoding radius.

### 1.3.1 High error bivariate testing

The bivariate axis-parallel test theorem of Polischuck and Spielman [18] is a fundamental component in many efficient PCP constructions. Roughly, the theorem says that if a function  $f : \mathbb{F} \times \mathbb{F}$  has the property that its restriction to *most* columns is *very close* to a degree  $d_Y$  polynomial, and the restriction to *most* rows is a function that is *very close* to a degree  $d_X$  polynomial, then  $f$  is *very close* to being the evaluation of a bivariate polynomial of degree  $d_X$  in  $X$  and degree  $d_Y$  in  $Y$ .

As stated there, the result works for degrees  $d_X, d_Y$  as large as  $\approx |\mathbb{F}|/2$  but requires the columns and rows to have *large agreement* with univariate low-degree polynomials, and this setting is known as the *low error regime*. An intriguing question is whether a similar result holds in the high-error regime, when only a non-trivial fraction of rows/columns exhibit non-trivial agreement with degree  $d$  polynomials.

This question has been given a positive answer by Arora and Safra for a richer class of tests that includes the restriction of  $f$  to all lines (not just axis-parallel ones), and when  $d < |\mathbb{F}|^{1/3}$  [4]. Recently, Chiesa, Manohar and Shinkar have proven the high-error case of the axis parallel test for small degree, i.e., when both  $d_X$  and  $d_Y$  are less than  $\log |\mathbb{F}|$  [13].

As the first application of our results, we improve on [18] and present a high-error analysis of the axis-parallel test. Our result, stated in Theorem 14 works when one of the degrees is a constant ( $d_X = O(1)$ ), but the other can be arbitrarily large  $d_Y = \Omega(|\mathbb{F}|)$ . Thus, our result is incomparable to that of [13], because of the different requirements on  $d_X, d_Y$ ; the proof techniques are also quite different.

### 1.3.2 Improved soundness analysis of the Fast Reed-Solomon interactive oracle proof of proximity (IOPP)

The *fast RS IOPP* (FRI) protocol [9] is an interactive oracle proof of proximity (IOPP) for the *RS proximity testing* (RPT) problem (cf. [12, 10] for a definition and discussion of the IOPP model). For RS-codes of message length  $N$  over a field  $\mathbb{F}$ , prover arithmetic



complexity is  $O(N)$  and verifier arithmetic complexity for each test<sup>6</sup> is  $O(\log N)$ ; this also bounds the query complexity of a single test. The efficiency of the FRI protocol is important for proof systems realized in code, like the recent zero knowledge proof system of [8], called a “zk-STARK” there.

The soundness of a proximity testing protocol is described by a *soundness function*  $s(\cdot)$  that takes as input a *proximity parameter*  $\delta$ , and outputs the minimum rejection probability of the verifier, where this minimum is taken over all words that are  $\delta$ -far from the code. In the case of FRI soundness for a single test, an upper bound  $s(\delta) \leq \delta$  is easy to establish. The analysis in [9] showed a nearly matching lower bound for *sufficiently small* values of  $\delta$ . In particular, the bound obtained there gives

$$s(\delta) \geq \min\{\delta, \delta_0\} - o(1) \tag{1}$$

where  $\delta_0$  is a *soundness threshold* constant that depends on the code rate  $\rho$  as follows  $\delta_0 \approx \frac{1-3\rho}{4}$  (see red line in Figure 1). For codes of rate  $\geq 1/3$  this bound is meaningless, and even when  $\rho \rightarrow 0$  it holds that  $\delta_0 \rightarrow 1/4$ ; this rather low soundness means that many tests must be applied in order to reach a target soundness error; for soundness error  $2^{-\lambda}$  and maximal proximity parameter  $1 - \rho$ , the number of tests must still be greater than  $\frac{\lambda}{-\log_2 \frac{3}{4}} \approx 2.4 \cdot \lambda$ .

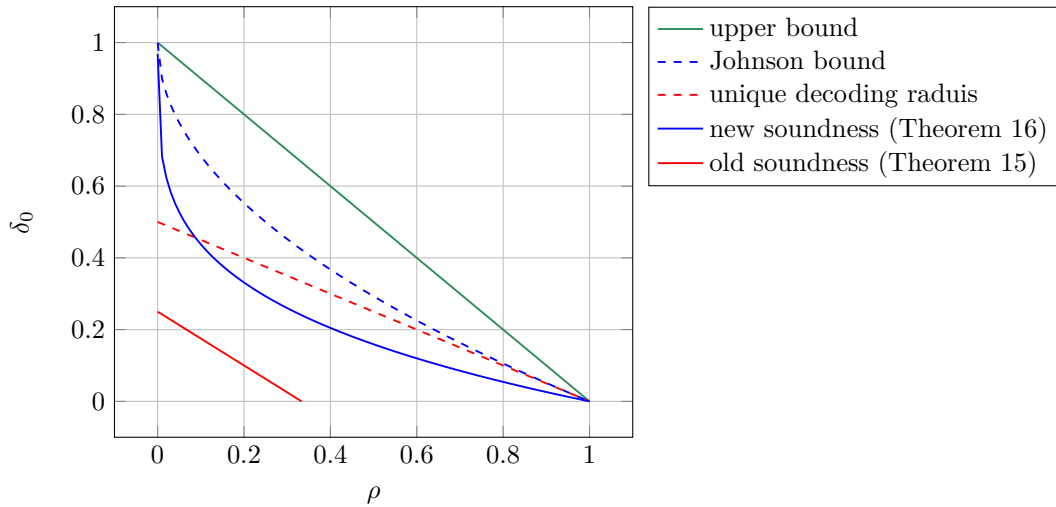
Using the results described in Sections 1.2.1 and 4.2 we improve on this state of affairs, and show that FRI soundness (for a single test) behaves as in Equation (1) but for a larger value of  $\delta_0$ , namely,  $\delta_0 \approx 1 - \sqrt[4]{\rho}$  (see blue line in Figure 1). Consequently, to reach soundness error  $2^{-\lambda}$  as before, the number of tests is reduced to  $\approx \frac{4\lambda}{-\log \rho}$  which is always smaller than  $2.4 \cdot \lambda$  and approaches 0 as  $\rho \rightarrow 0$ . We end by pointing out that [9] conjecture that the trivial soundness upper bound (green line in Figure 1) is nearly tight, i.e., that  $s(\delta) \approx \delta$  for all values of  $\delta$ . Reducing further the gap between soundness upper bounds (green line) and lower bounds (blue line) remains an interesting open problem that is relevant to realized proof systems like the zk-STARK of [8].

## 1.4 Soundness amplification for Reed-Solomon codes

So far we tried to minimize the loss in distance incurred by sampling an element of  $U$ . Next, we suggest a way to boost distance via a family of “locally-computable” transformations acting on a *single* purported codeword  $u$ . A *q*-*locally computable transformation* is a mapping  $M : \mathbb{F}^n \rightarrow \mathbb{F}^n$  for which the *i*th entry of  $M(u)$  can be computed by querying at most *q* entries of  $u$ . To preserve completeness, we require the mappings  $M$  to preserve the space  $V$ , and this leads to a natural suggestion. Let  $\text{Aut}(V)$  be the *automorphism group* of  $V$ . Sample  $M_1 \dots, M_{q-1} \in \text{Aut}(V)$  and  $r_1, \dots, r_{q-1}$  and let  $u^* = M(u) \triangleq u + \sum_{i < q} r_i M_i(u)$ . By definition, this mapping is *q*-local and it preserves (perfect) completeness: if  $u$  belongs to  $V$  then so does each  $M_i(u)$ , so by linearity  $M(u) \in V$ . It now stands to reason that if  $\text{Aut}(V)$  is sufficiently “pseudo-random”, say, a doubly-transitive group, then the median distance of  $M(u)$  should be even greater than  $\Delta(u, V)$  (the distance of  $u$  from  $V$ ).

For example, consider the family of Reed-Solomon codes  $\text{RS}[\mathbb{F}, \rho]$ , which are comprised of all functions  $f : \mathbb{F} \rightarrow \mathbb{F}$  such that  $\deg(f) < \rho|\mathbb{F}|$  where  $\deg(f)$  is the degree of the interpolating polynomial of (the function)  $f$ . It is well known that  $\text{Aut}(\text{RS}[\mathbb{F}, \rho])$  is the 1-dimensional affine group of  $\mathbb{F}$ , denoted  $\text{Aff}_1(\mathbb{F})$ , whose members are all invertible affine transformations  $\text{Aff}_1(\mathbb{F}) = \{M(X) = aX + b \mid a \in \mathbb{F}^*, b \in \mathbb{F}\}$ ; this group is indeed doubly-transitive.

<sup>6</sup> In [9], a single test means a single invocation of the QUERY protocol.,



■ **Figure 1** FRI soundness threshold  $\delta_0$  as a function of RS code rate  $\rho$ , for a single invocation of the FRI QUERY phase (see Equation (1) and explanation in text there for the meaning of the constant  $\delta_0$ ). Higher lines are better. The top line is the trivial upper bound on soundness; the bottom line is the soundness of the original analysis of [9] (cf. Theorem 15). The middle line is the new and improved analysis given by Theorem 16. This analysis presents non-trivial soundness bounds for all code rates, and these bounds are better than the prior state of the art.

Our final set of results studies the effect of taking random linear combinations of random automorphisms for Reed-Solomon codes. Suppose we start with a function  $u$ , and then take random linear combinations of a few random affine shifts of  $u$  to produce a function  $u^*$ . From the discussion above, if  $u$  is in a Reed-Solomon code, then so is  $u^*$ . We show in Theorem 13 that if  $u$  is far from a Reed-Solomon code, then with high probability  $u^*$  is very far from that Reed-Solomon code. The main strength of this result is that this process can then amplify the distance to  $V$  all the way to  $1 - o(1)$  (while more direct analyses, related to the Rothblum-Vadhan-Wigderson [19] lemma, cannot amplify beyond distance  $1/2$ ).

## 2 Preliminaries

We use  $\Delta$  to denote normalized Hamming distance, and  $\mathbf{0} = 0^n$  denotes the identity element of an  $n$ -dimensional vector space, viewed as an additive group.

In what follows  $\Sigma$  is a finite alphabet. For  $S \subset \Sigma^n$  let

$$\Delta(S) = \min \{ \Delta(w, w') \mid w, w' \in S, w \neq w' \}$$

denote the relative Hamming distance of  $S$ . For  $w \in \Sigma^n$  let  $B(w, \delta)$  denote the Hamming ball in  $\Sigma^n$  of normalized radius  $\delta$  centered at  $w$ ,

$$B(w, \delta) = \{ r \in \Sigma^n \mid \Delta(w, r) < \delta \}$$

► **Definition 1** (List decodability). For  $\rho \in [0, 1]$  and  $L \geq 1$ , we say a set  $S \subseteq \Sigma^n$  is  $(\rho, L)$ -list-decodable if for all  $w \in \Sigma^n$ ,

$$|B(w, \rho) \cap S| \leq L.$$

We have the fundamental Johnson bound, which says that sets with large minimum distance have nontrivial list-decodability. See, e.g., [15, Corollary 3.2] for a proof.

► **Theorem 2** (Johnson bound). For every  $\epsilon \in (0, 1]$ , Let  $J_\epsilon : [0, 1] \rightarrow [0, 1]$  be the function

$$J_\epsilon(\delta) = 1 - \sqrt{1 - \delta(1 - \epsilon)}.$$

Let  $\Sigma$  be a finite alphabet,  $n$  an integer and  $S \subseteq \Sigma^n$ . Then  $S$  is  $(J_\epsilon(\Delta(S)), 1/\epsilon)$ -list-decodable for every  $\epsilon \in (0, 1]$ .

An affine space  $U$  is an additive coset of a vector space  $U'$ , i.e., for some fixed  $a \in \mathbb{F}^n$ ,  $U = a + U' \triangleq \{a + u \mid u \in U'\}$ . We introduce the following definition.

► **Definition 3** (Divergence). For  $U, V \subseteq \Sigma^n$ , the *divergence* of  $U$  from  $V$  is  $D(U, V) = \max_{u \in U} \Delta(u, V)$ .

Divergence is not a distance measure because it is not symmetric. This is witnessed by  $U = \{\mathbf{0}, 10^{n-1}\}$ ,  $V = \{\mathbf{0}, 1^n\} \subset \{0, 1\}^n$ , which gives  $D(U, V) = \frac{1}{n} \neq \frac{n-1}{n} = D(V, U)$ .

The next lemma, due to Rothblum-Vadhan-Wigderson, says that if some vector in a linear space  $U$  is  $\delta$ -far from a space  $V$ , then nearly all elements of  $U$  are  $\delta/2$ -far from  $V$ .

► **Lemma 4** ([19, Lemma 1.6]). For any pair of linear spaces  $U, V$  over a finite field  $\mathbb{F}$ ,

$$\Pr_{u \in U} \left[ \Delta(u, V) < \frac{D(U, V)}{2} \right] \leq \frac{1}{|\mathbb{F}| - 1}. \quad (2)$$

### 3 Preserving distances for general subspaces

In this section, we prove our first strengthening of the Rothblum-Vadhan-Wigderson lemma Lemma 4 from above. The main new qualitative feature is that if  $D(U, V) = 1 - o(1)$ , then the lemma concludes that most elements of  $u$  are at distance  $1 - o(1)$  from  $V$ .

► **Theorem 5.** For a pair of affine spaces  $U, V$  over a finite field  $\mathbb{F}$ , and for all  $\epsilon \in (0, 1]$ ,

$$\Pr_{u \in U} [\Delta(u, V) < J_\epsilon(D(U, V))] < \frac{1}{\epsilon(|\mathbb{F}| - 1)}.$$

Theorem 5 is a consequence of the following lemma, which says that if  $u^*$  is  $\delta$ -far from  $V$ , then for any line passing through  $u^*$  in direction  $u$ , most points are  $J_\epsilon(\delta)$  from  $V$ . We state the Lemma, prove Theorem 5 and then prove the lemma.

► **Lemma 6.** Let  $V \subseteq \mathbb{F}^n$  be a linear space over a finite field  $\mathbb{F}$ ; suppose  $u^* \in \mathbb{F}^n$  satisfies  $\Delta(u^*, V) \geq \delta$ . For any  $u \in \mathbb{F}^n$  and  $\epsilon \in (0, 1]$  let

$$A = A_{u, \epsilon} = \{\alpha \in \mathbb{F} \setminus \{0\} \mid \Delta(u^* + \alpha u, V) < J_\epsilon(\delta)\}.$$

Then  $|A| \leq 1/\epsilon$ .

**Proof of Theorem 5.** It suffices to prove the Theorem for the case that  $V$  is a linear space and  $U$  is an affine space (which may be linear as well), because Hamming distance is invariant under shifting both  $U$  and  $V$  by the same vector  $v$ . Let  $u^* \in U$  be some element for which  $\Delta(u^*, V) = D(U, V)$ . We may assume  $u^* \neq \mathbf{0}$ , otherwise  $D(U, V) = 0$  because  $\mathbf{0} \in V$  so the claim trivially holds. If  $\dim(U) = 0$  the claim also trivially holds because  $|U| = 1$ . Therefore, we assume  $U = u^* + U'$  for some linear space  $U'$  of positive dimension  $d$  (which may include  $u^*$ ). There exist  $k = |\mathbb{F}|^{d-1}$  vectors  $u_1, \dots, u_k$  such that  $U \setminus \{u^*\}$  can be partitioned into equi-sized sets, the  $i$ th set being the line  $\{u^* + \alpha u_i \mid \alpha \in \mathbb{F} \setminus \{0\}\}$ . Theorem 5 follows by applying Lemma 6 to each of the sets in this partition. ◀

**Proof of Lemma 6.** For  $\alpha \in A$ , let  $v^\alpha \in V$  be such that  $\Delta(u^* + \alpha u, v^\alpha) < J_\epsilon(\delta)$ . Rewriting, we have that for each  $\alpha \in A$ ,

$$\Delta\left(u, \frac{v^\alpha - u^*}{\alpha}\right) < J_\epsilon(\delta).$$

Assume by way of contradiction that  $|A| > 1/\epsilon$ . Thus, a set (or possibly multi-set) of more than  $1/\epsilon$  vectors are all  $J(\delta, \epsilon)$ -close to  $u$ . By the Johnson bound, two of the vectors must be  $\delta$ -close to one another. Let  $\alpha, \alpha'$  be these distinct members of  $A$  for which

$$\Delta\left(\frac{v^\alpha - u^*}{\alpha}, \frac{v^{\alpha'} - u^*}{\alpha'}\right) < \delta.$$

Recalling  $\Delta(u, v) = \Pr_{i \in [n]}[u_i \neq v_i]$  where  $u_i, v_i$  denote the  $i$ th entry of  $u, v$ , respectively, we have

$$\begin{aligned} \delta &> \Pr_{i \in [n]} \left[ \left( \frac{v^\alpha - u^*}{\alpha} \right)_i \neq \left( \frac{v^{\alpha'} - u^*}{\alpha'} \right)_i \right] = \Pr_{i \in [n]} \left[ \left( \frac{v^\alpha - u^*}{\alpha} - \frac{v^{\alpha'} - u^*}{\alpha'} \right)_i \neq 0 \right] \\ &= \Pr_{i \in [n]} \left[ \left( (\alpha - \alpha')u^* - (\alpha v^{\alpha'} - \alpha' v^\alpha) \right)_i \neq 0 \right] \\ &= \Pr_{i \in [n]} \left[ u_i^* \neq \left( \frac{\alpha v^{\alpha'} - \alpha' v^\alpha}{\alpha - \alpha'} \right)_i \right]. \end{aligned}$$

Setting  $v' = \frac{\alpha v^{\alpha'} - \alpha' v^\alpha}{\alpha - \alpha'}$  and noticing  $v' \in V$  we conclude

$$\Delta(u^*, V) \leq \Delta(u^*, v') < \delta$$

which is false and which contradicts our hypothesis on the size of  $A$ . We conclude  $|A| \leq 1/\epsilon$ , as claimed.  $\blacktriangleleft$

## 4 Preserving distances for good error correcting code

In this section we prove another strengthening of the Rothblum-Vadhan-Wigderson lemma. This strengthening only works when the subspace  $V$  is a code of good distance. Assume for now that  $V$  is a code with minimum distance  $1 - o(1)$ . Then the strengthened theorem gives a stronger guarantee: they show that most elements of  $U$  are at distance  $\min(D(U, V) - o(1))$  from  $V$ . Thus the maximum distance of an element of  $U$  from  $V$  is also the typical distance of an element of  $U$  from  $V$ .

We begin with a warm-up: we show a “unique-decoding” version which only works up to  $1/3$  of the minimum distance of the code  $V$ . The next “list-decoding” version works up to a much larger distance, and in particular for  $V$  having distance  $1 - o(1)$ , it works up to a distance of  $1 - o(1)$ .

### 4.1 Unique-Decoding version

► **Theorem 7.** *Let  $V \subseteq \mathbb{F}^n$  be a linear space over a finite field  $\mathbb{F}$  with  $\Delta(V) = \lambda$ . Let  $U$  be an affine space and suppose  $D(U, V) > \delta$ . For any  $\epsilon > 0$  such that  $\delta - \epsilon < \lambda/3$ ,*

$$\Pr_{u \in U} [\Delta(u, V) < \delta - \epsilon] \leq \frac{1}{\epsilon |\mathbb{F}|}$$

Theorem 7 is a consequence of the following lemma. As in Section 3, we state the lemma, prove Theorem 7 and then prove the lemma.

► **Lemma 8.** *Let  $V \subseteq \mathbb{F}^n$  be a linear space over a finite field  $\mathbb{F}$  with  $\Delta(V) = \lambda$ . Suppose  $u^* \in \mathbb{F}^n$  satisfies  $\Delta(u^*, V) > \delta$  and fix arbitrary  $u \in \mathbb{F}^n$ . For  $\epsilon > 0$  satisfying  $\delta - \epsilon < \lambda/3$  let  $A = \{\alpha \in \mathbb{F} \mid \Delta(u^* + \alpha u, V) < \delta - \epsilon\}$ . If  $|A| > 1/\epsilon$  then there exist  $v, v^* \in V$  such that:*

$$|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}| \geq (1 - \delta) \cdot n.$$

**Proof of Theorem 7.** We prove the contra-positive: If the assumptions on  $\epsilon, \delta, \lambda$  hold and

$$\Pr_{u \in U} [\Delta(u, V) < \delta - \epsilon] > \frac{1}{\epsilon |\mathbb{F}|}, \quad (3)$$

then  $D(U, V) \leq \delta$ .

Let  $u^* \in U$  satisfy  $\Delta(u^*, V) = D(U, V)$ . We may assume  $V$  is a linear space and  $\dim(U) > 0$ , as argued in the proof of Theorem 5. As there, partition  $U \setminus \{u^*\}$  into equi-size sets, each of the form  $\{u^* + \alpha u_i \mid \alpha \in \mathbb{F} \setminus \{0\}\}$  for some set  $u_1, \dots, u_k \in \mathbb{F}^n$  of vectors. By our assumption in Equation (3) there exists  $u_i$  such that the set  $A = \{\alpha \in \mathbb{F} \mid \Delta(u^* + \alpha u_i, V) < \delta - \epsilon\}$  is of size greater than  $1/\epsilon$ . Apply Lemma 8 to this set, and conclude  $\Delta(u^*, v^*) \leq \delta$ , as claimed. ◀

**Proof of Lemma 8.** For  $\alpha \in A$ , let  $v^\alpha \in V$  be such that  $\Delta(u^* + \alpha u, v^\alpha) < \delta - \epsilon$ .

We first show that for all  $\alpha \in A$ , the points  $(\alpha, v^\alpha)$  are all collinear. To see this, let  $\alpha_1, \alpha_2, \alpha_3 \in A$  be distinct. We have  $\Delta(u^* + \alpha_3 u, v^{\alpha_3}) \leq \delta - \epsilon$ . On the other hand, if  $\beta = \frac{\alpha_3 - \alpha_2}{\alpha_1 - \alpha_2}$ , we have:

$$u^* + \alpha_3 u = \beta(u^* + \alpha_1 u) + (1 - \beta)(u^* + \alpha_2 u),$$

and so:

$$\begin{aligned} \Delta(u^* + \alpha_3 u, \beta v^{\alpha_1} + (1 - \beta)v^{\alpha_2}) &\leq \Delta(u^* + \alpha_1 u, v^{\alpha_1}) + \Delta(u^* + \alpha_2 u, v^{\alpha_2}) \\ &\leq (\delta - \epsilon) + (\delta - \epsilon) \\ &= 2(\delta - \epsilon). \end{aligned}$$

Thus  $\Delta(\beta v^{\alpha_1} + (1 - \beta)v^{\alpha_2}, v^{\alpha_3}) \leq 3(\delta - \epsilon) < \lambda$ . By the minimum distance hypothesis on  $V$ , we conclude that

$$\beta v^{\alpha_1} + (1 - \beta)v^{\alpha_2} = v^{\alpha_3},$$

which implies the desired collinearity.

Thus there exist  $v, v^* \in V$  such that for all  $\alpha \in A$ ,

$$v^\alpha = v^* + \alpha v.$$

Taking this information back to the definition of  $v^\alpha$ , we have that for all  $\alpha \in A$ ,

$$\Delta(u^* + \alpha u, v^* + \alpha v) < \delta - \epsilon.$$

Rewriting,

$$\Delta(u^* - v^*, \alpha(v - u)) < \delta - \epsilon.$$

for all  $\alpha \in A$ .

## 24:10 Worst-Case to Average Case Reductions for the Distance to a Code

Now for any coordinate  $i \in [n]$  where  $u_i \neq v_i$  or  $u_i^* \neq v_i^*$ , there can be at most one value of  $\alpha \in \mathbb{F}$  for which  $u_i^* - v_i^* = \alpha(v_i - u_i)$ . Let  $t = |A|$ . Thus there is an  $\alpha \in A$  such that:

$$\delta - \epsilon > \Delta(u^* - v^*, \alpha(v - u)) \geq 1 - \frac{|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}|}{n} - \frac{1}{t}.$$

Putting everything together, we get that:

$$\frac{|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}|}{n} \geq 1 - \delta + \epsilon - \frac{1}{t}.$$

Thus if  $t > \frac{1}{\epsilon}$ , we have:

$$\frac{|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}|}{n} \geq 1 - \delta,$$

as claimed. ◀

### 4.2 List-Decoding version

► **Theorem 9.** *Let  $V \subseteq \mathbb{F}_q^n$  be a subspace with minimum distance  $\lambda$ . Let  $\epsilon, \delta > 0$  with  $\delta < J_\epsilon(J_\epsilon(\lambda))$ .*

*Suppose  $u^* \in \mathbb{F}_q^m$  is such that  $\Delta(u^*, V) > \delta$ . Then for all  $u \in \mathbb{F}_q^n$ , there are at most  $2/\epsilon^3$  values of  $\alpha \in \mathbb{F}_q$  such that  $\Delta(u^* + \alpha u, V) < \delta - \epsilon$ .*

This is a consequence of the following theorem.

► **Theorem 10.** *Let  $V \subseteq \mathbb{F}^n$  be a linear space over a finite field  $\mathbb{F}$  with  $\Delta(V) = \lambda$ . Let  $u^* \in \mathbb{F}^n$  and let  $\epsilon > 0$  satisfy  $\delta < J_\epsilon(J_\epsilon(\lambda))$ . For  $u \in \mathbb{F}^n$  let*

$$A = A_{u, \epsilon} = \{\alpha \in \mathbb{F} \setminus \{0\} \mid \Delta(u^* + \alpha u, V) < \delta - \epsilon\}.$$

*If  $|A| > 2/\epsilon^3$  then there exist  $v^*, v \in V$  such that*

$$|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}| \geq (1 - \delta)n.$$

*In particular,*

$$\Delta(u^*, v^*) \leq \delta.$$

**Proof.** Let  $t = |A|$ . For  $\alpha \in A$ , let  $v^\alpha \in V$  be such that  $\Delta(u^* + \alpha u, v^\alpha) < \delta - \epsilon$ . Thus  $\Delta(u^*, v^\alpha - \alpha u) < \delta - \epsilon$ .

Now consider the following graph with vertex set  $A$ :  $\alpha$  and  $\alpha'$  are adjacent if  $\Delta(v^\alpha - \alpha u, v^{\alpha'} - \alpha' u) < J_\epsilon^{-1}(\delta)$ . The Johnson bound implies that this graph has no independent set of size  $c' = 1/\epsilon$ . Thus by Turan's theorem, there is a vertex  $\alpha_0$  of degree at least  $\epsilon|A| - 1$ .

Concretely, this means that there is a set  $B \subseteq A$ , with  $|B| \geq \epsilon|A| - 1$ , such that for all  $\alpha \in B$ ,

$$\Delta(v^{\alpha_0} - \alpha_0 u, v^\alpha - \alpha u) < J_\epsilon^{-1}(\delta).$$

Rewriting, we have:

$$\Delta(u, \frac{1}{\alpha - \alpha_0} \cdot (v^\alpha - v^{\alpha_0})) < J_\epsilon^{-1}(\delta), \tag{4}$$

for every  $\alpha \in B$ .

Now we apply the Johnson bound again. Since  $V$  has distance  $\lambda$ , and  $J_\epsilon(\lambda) > J_\epsilon^{-1}(\delta)$ , there can be at most  $1/\epsilon$  distinct vectors  $v \in V$  such that  $\Delta(u, v) < J_\epsilon^{-1}(\delta)$ .

The only way this can be consistent with Equation (4) is if many of the  $\frac{1}{\alpha - \alpha_0} \cdot (v^\alpha - v^{\alpha_0})$  are identical. Specifically, by the pigeonhole principle we get that there is a  $v \in V$  and a set  $C \subseteq B$ , with  $|C| \geq \epsilon|B|$ , such that for all  $\alpha \in C$ ,

$$v = \frac{1}{\alpha - \alpha_0} \cdot (v^\alpha - v^{\alpha_0}).$$

So for all  $\alpha \in C$ ,

$$v^\alpha = (v^{\alpha_0} - \alpha_0 v) + \alpha \cdot v.$$

Let us denote this by  $v^\alpha = v^* + \alpha v$ , where  $v, v^* \in V$ .

Taking this information back to the definition of  $v^\alpha$ , we have that for all  $\alpha \in C$ ,

$$\Delta(u^*, v^* + \alpha(v - u)) < \delta - \epsilon.$$

Rewriting,

$$\Delta(u^* - v^*, \alpha(v - u)) < \delta - \epsilon.$$

for all  $\alpha \in C$ .

Now for any coordinate  $i \in [n]$  where  $u_i \neq v_i$  or  $u_i^* \neq v_i^*$ , there can be at most one value of  $\alpha \in \mathbb{F}$  for which  $u_i^* - v_i^* = \alpha(v_i - u_i)$ . Thus there is an  $\alpha \in C$  such that

$$\Delta(u^* - v^*, \alpha(v - u)) \geq 1 - \frac{|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}|}{n} - \frac{1}{|C|}.$$

Combining this with our upper bound on  $\Delta(u^* - v^*, \alpha(v - u))$ , we get that:

$$\frac{|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}|}{n} \geq 1 - \delta + \epsilon - \frac{1}{|C|}. \tag{5}$$

Since  $|C| \geq \epsilon|B| \geq \epsilon(\epsilon|A| - 1)$ , and since  $A \geq 2/\epsilon^3$ , we get that

$$|C| > 1/\epsilon,$$

and the desired conclusion follows from Equation (5).  $\blacktriangleleft$

We now state a variation of Theorem 10. The proof of this theorem follows immediately from the proof of Theorem 10 and hence we omit it. The reason we have this variation is that this precise form of the statement will be useful later in the proof of the low degree Polischuk-Spielman theorem.

**► Theorem 11.** *Let  $V \subseteq \mathbb{F}^n$  be a linear space over a finite field  $\mathbb{F}$  with  $\Delta(V) = \lambda$ . Let  $u^* \in \mathbb{F}^n$  and let  $\epsilon > 0$  satisfy  $\delta < J_\epsilon(J_\epsilon(\lambda))$ . For  $u \in \mathbb{F}^n$  let*

$$A = A_{u, \epsilon} = \{\alpha \in \mathbb{F} \setminus \{0\} \mid \Delta(u^* + \alpha u, V) < \delta\}.$$

*If  $|A| > 2/\epsilon^3$  then the following two statements hold:*

1. *There exist  $v^*, v \in V$  such that*

$$|\{i \in [n] \mid (u_i = v_i) \wedge (u_i^* = v_i^*)\}| \geq (1 - \delta - \epsilon)n.$$

*In particular,*

$$\Delta(u^*, v^*) \leq \delta + \epsilon.$$

24:12 Worst-Case to Average Case Reductions for the Distance to a Code

2. For  $\alpha \in A$ , fix  $v^\alpha \in V$  such that  $\Delta(u^* + \alpha v, v^\alpha) < \delta$ . Then there is a large subset  $C \subseteq A$  such that  $|C| \geq \epsilon^2 |A| - 1$  and such that for all  $\alpha \in C$ ,  $v^* + \alpha v = v^\alpha$ .

We now state a strengthening of the above theorem from lines to higher degree curves. Define  $J_\epsilon^{[k]}(\lambda) = J_\epsilon(J_\epsilon(\dots(J_\epsilon(\lambda))))$ , where there are  $k$  iterations of the function  $J_\epsilon$ .

► **Theorem 12.** Let  $V \subseteq \mathbb{F}^n$  be a linear space over a finite field  $\mathbb{F}$  with  $\Delta(V) = \lambda$ . Let  $u^* \in \mathbb{F}^n$  and let  $\epsilon > 0$  satisfy  $\delta < J_\epsilon^{[\ell+1]}(\lambda)$ . For  $u_1, u_2, \dots, u_\ell \in \mathbb{F}^n$  let  $A = A_{u_1, u_2, \dots, u_\ell, \epsilon} = \{\alpha \in \mathbb{F} \setminus \{0\} \mid \Delta(u^* + \alpha u_1 + \alpha^2 u_2 + \alpha^3 u_3 + \dots + \alpha^\ell u_\ell, V) < \delta\}$ . If  $|A| > K_{\epsilon, \ell}$  for some sufficiently large constant  $K_{\epsilon, \ell}$  that depends only on  $\epsilon$  and  $\ell$ , then the following two statements hold:

1. There exist  $v^*, v_1, v_2, \dots, v_\ell \in V$  such that

$$|\{i \in [n] \mid (u_i^* = v_i^*) \wedge ((u_1)_i = (v_1)_i) \wedge \dots \wedge ((u_\ell)_i = (v_\ell)_i)\}| \geq (1 - \delta - \epsilon)n.$$

In particular,

$$\Delta(u^*, v) \leq \delta + \epsilon.$$

2. For  $\alpha \in A$ , fix  $v^\alpha \in V$  such that  $\Delta(u^* + \alpha u_1 + \alpha^2 u_2 + \alpha^3 u_3 + \dots + \alpha^\ell u_\ell, v^\alpha) < \delta$ . Then there is a large subset  $C \subseteq A$  such that  $|C| \geq c_{\epsilon, \ell} |A|$ , where  $c_{\epsilon, \ell} > 0$  is a constant only depending on  $\epsilon$  and  $\ell$ , and such that for all  $\alpha \in C$ ,  $v^* + \alpha v_1 + \alpha^2 v_2 + \alpha^3 v_3 + \dots + \alpha^\ell v_\ell = v^\alpha$ .

**Proof.** The proof of the above theorem follows from the proof of Theorem 11 and an induction on  $\ell$ . When  $\ell = 1$ , the result follows from Theorem 11. Let us assume the result is true for  $\ell \leq k - 1$ , and now consider  $\ell = k$ .

Let  $t = |A|$ . For  $\alpha \in A$ , let  $v^\alpha \in V$  be such that  $\Delta(u^* + \alpha u_1 + \alpha^2 u_2 + \alpha^3 u_3 + \dots + \alpha^k u_k, v^\alpha) < \delta$ . Thus  $\Delta(u^*, v^\alpha - (\alpha u_1 + \alpha^2 u_2 + \alpha^3 u_3 + \dots + \alpha^k u_k)) < \delta$ .

Now consider the following graph with vertex set  $A$ :  $\alpha$  and  $\alpha'$  are adjacent if  $\Delta(v^\alpha - (\alpha u_1 + \alpha^2 u_2 + \alpha^3 u_3 + \dots + \alpha^k u_k), v^{\alpha'} - (\alpha' u_1 + \alpha'^2 u_2 + \alpha'^3 u_3 + \dots + \alpha'^k u_k)) < J_\epsilon^{-1}(\delta)$ . The Johnson bound implies that this graph has no independent set of size  $c' = 1/\epsilon$ . Thus by Turan's theorem, there is a vertex  $\alpha_0$  of degree at least  $\epsilon |A| - 1$ .

Concretely, this means that there is a set  $B \subseteq A$ , with  $|B| \geq \epsilon |A| - 1$ , such that for all  $\alpha \in B$ ,

$$\Delta(v^{\alpha_0} - (\alpha_0 u_1 + \alpha_0^2 u_2 + \alpha_0^3 u_3 + \dots + \alpha_0^k u_k), v^\alpha - (\alpha u_1 + \alpha^2 u_2 + \alpha^3 u_3 + \dots + \alpha^k u_k)) < J_\epsilon^{-1}(\delta).$$

Rewriting, we have:

$$\Delta\left(\frac{v^{\alpha_0} - v^\alpha}{\alpha_0 - \alpha}, \frac{1}{\alpha_0 - \alpha} \cdot \sum_{i=1}^k (\alpha_0^i - \alpha^i) u_i\right) < J_\epsilon^{-1}(\delta), \quad (6)$$

for every  $\alpha \in B$ .

Now, let  $W^\alpha \in V$  be the vector

$$\frac{v^{\alpha_0} - v^\alpha}{\alpha_0 - \alpha},$$

and let  $u'_1, u'_2, \dots, u'_k \in \mathbb{F}^n$  be such that  $u'_i$  is the coefficient of  $\alpha^{i-1}$  in

$$\frac{1}{\alpha_0 - \alpha} \cdot \sum_{i=1}^k (\alpha_0^i - \alpha^i) u_i.$$



Thus,

$$\frac{1}{\alpha_0 - \alpha} \cdot \sum_{i=1}^k (\alpha_0^i - \alpha^i) u_i = \sum_{i=1}^k u'_i \cdot \alpha^{i-1},$$

and for all  $\alpha \in B$ ,

$$\Delta(W^\alpha, u'_1 + \alpha u'_2 + \cdots + \alpha^{k-1} u'_k) < J_\epsilon^{-1}(\delta).$$

Notice that since we are given that  $\delta < J_\epsilon^{[k+1]}(\lambda)$ , where the function  $J_\epsilon$  is iterated  $k+1$  times, thus for  $\delta' = J_\epsilon^{-1}(\delta)$ ,  $\delta' < J_\epsilon^{[k]}(\lambda)$ , where the function  $J_\epsilon$  is iterated  $k$  times.

Now,  $|B| \geq \epsilon|A| - 1$ . Thus, if  $|A|$  is large enough, then by induction hypothesis, there is a large subset  $C \subseteq B$  such that  $|C| \geq c_{\epsilon,k}|B|$ , where  $c_{\epsilon,k} > 0$  is a constant only depending on  $\epsilon$  and  $k$ , and there exist  $v'_1, v'_2, \dots, v'_k \in V$  such that for all  $\alpha \in C$ ,  $v'_1 + \alpha v'_2 + \alpha^2 v'_3 + \cdots + \alpha^k v'_k = W^\alpha$ .

Thus

$$\frac{v^{\alpha_0} - v^\alpha}{\alpha_0 - \alpha} = v'_1 + \alpha v'_2 + \alpha^2 v'_3 + \cdots + \alpha^k v'_k,$$

where  $v'_1, v'_2, \dots, v'_k \in V$ .

Rearranging, this shows that for all  $\alpha \in C$ , we can express  $v^\alpha$  as  $v^* + \alpha v_1 + \alpha^2 v_2 + \alpha^3 v_3 + \cdots + \alpha^k v_k$ , where  $v^*, v_1, v_2, \dots, v_k \in V$ .

Taking this back to the definition of  $v^\alpha$ , we have that for all  $\alpha \in C$ ,

$$\Delta(u^*, v^* + (\alpha(v_1 - u_1) + \alpha^2(v_2 - u_2) + \alpha^3(v_3 - u_3) + \cdots + \alpha^k(v_k - u_k))) < \delta.$$

Rewriting, we have that for all  $\alpha \in C$ ,

$$\Delta(u^* - v^* + \alpha(u_1 - v_1) + \alpha^2(u_2 - v_2) + \alpha^3(u_3 - v_3) + \cdots + \alpha^k(u_k - v_k), 0) < \delta.$$

Now for any coordinate  $i \in [n]$  where  $u_i^* \neq v_i^*$  or  $(u_j)_i \neq (v_j)_i$  for any  $j \in [k]$ , the restriction to the  $i$ th coordinate gives us a nonzero degree  $k$  polynomial in  $\alpha$ , and so there are at most  $k$  values of  $\alpha \in \mathbb{F}$  for which  $(u^* - v^*)_i + \alpha \cdot (u_1 - v_1)_i + \alpha^2 \cdot (u_2 - v_2)_i + \alpha^3 \cdot (u_3 - v_3)_i + \cdots + \alpha^k \cdot (u_k - v_k)_i = 0$ .

Thus there is an  $\alpha \in C$  such that

$$\Delta(u^* - v^* + \alpha(u_1 - v_1) + \alpha^2(u_2 - v_2) + \alpha^3(u_3 - v_3) + \cdots + \alpha^k(u_k - v_k), 0) \geq 1 - \frac{|\{i \in [n] \mid (u_i^* = v_i^*) \wedge ((u_1)_i = (v_1)_i) \wedge \cdots \wedge ((u_k)_i = (v_k)_i)\}|}{n} - \frac{k}{|C|}.$$

Combining this with our upper bound on

$$\Delta(u^* - v^* + \alpha(u_1 - v_1) + \alpha^2(u_2 - v_2) + \alpha^3(u_3 - v_3) + \cdots + \alpha^k(u_k - v_k), 0),$$

we get that:

$$\frac{|\{i \in [n] \mid (u_i^* = v_i^*) \wedge ((u_1)_i = (v_1)_i) \wedge \cdots \wedge ((u_k)_i = (v_k)_i)\}|}{n} \geq 1 - \delta - \frac{k}{|C|}. \quad (7)$$

Since Now is  $|A|$  is a large enough constant depending on  $\epsilon$  and  $k$ , then the bounds on  $|C|$  imply that

$$|C| > k/\epsilon,$$

and the desired conclusion follows from Equation (7).  $\blacktriangleleft$

## 5 Distance Amplification for Reed-Solomon codes

In this section, we show how to use the results of the previous section to show that some simple transformations *amplify* the distance of a function from the space of low-degree polynomials (i.e., Reed-Solomon codes). In the previous section, we saw results with the flavor: if  $u^*$  is  $\delta$ -far from the subspace  $V$ , then there are many other functions (related to  $u^*$ ) that are also almost  $\delta$ -far from the subspace  $V$ . Now we will get more: we will find many functions related to  $u^*$  that are  $\delta'$ -far from  $V$  for some  $\delta'$  bigger than  $\delta$ . The main strength of this result is that this process can then amplify the distance to  $V$  all the way to  $1 - o(1)$  (while more direct analyses, related to the Rothblum-Vadhan-Wigderson [19] lemma, cannot amplify beyond distance  $1/2$ ).

For a function  $u^*$  we consider taking random linear combinations of a few random affine shifts of  $u^*$ . Notice that if  $u^*$  was actually a low-degree polynomial, then the resulting function would also be a low-degree polynomial (since low-degree polynomials are closed under taking affine shifts and taking linear combinations). We show that if  $u^*$  is far from low-degree polynomials, this operation amplifies distance to low-degree polynomials noticeably. More precisely, suppose  $V$  is the space of polynomials of degree at most  $\rho q$ , let  $\delta > 0$ , and suppose  $\rho > 0$  is small enough as a function of  $\delta$ . We show that if  $u^*$  is  $\delta$ -far from  $V$ , then the function  $u(x) = u^*(x) + c \cdot u^*(ax + b)$  (where  $a, b, c$  are picked uniformly at random from  $\mathbb{F}_q$ ) is with high probability  $\approx (2\delta - \delta^2)$  far from  $V$ . This final distance matches what one would expect if we took the sum of two random functions that were  $\delta$ -far from  $V$  - thus the random affine shift of  $u^*$  behaves nearly independently of  $u^*$  (subject to the trivial constraint that the random affine shift is also  $\delta$ -far from  $V$ ).

To state the theorem, we begin with some notation. For a function  $f : \mathbb{F}_q \rightarrow \mathbb{F}_q$ , we denote by  $T_{a,b}(f)$  the function  $g : \mathbb{F}_q \rightarrow \mathbb{F}_q$  given by:

$$g(\beta) = f(a\beta + b),$$

for each  $\beta \in \mathbb{F}_q$ .

► **Theorem 13.** *Let  $V = \text{RS}(\mathbb{F}_q, (1 - \lambda)q) \subseteq \mathbb{F}_q^q$  be the Reed-Solomon code over  $\mathbb{F}_q$  with minimum distance  $\lambda$ .*

*Let  $u', u'' : \mathbb{F}_q \rightarrow \mathbb{F}_q$  be functions with  $\Delta(u', V) \geq \delta'$  and  $\Delta(u'', V) \geq \delta''$ . Let  $\epsilon > 0$ , and let*

$$\delta = \min(J_\epsilon(J_\epsilon(\lambda)) - \epsilon, \delta' + \delta'' - \delta'\delta'' - 2\epsilon).$$

*Then:*

$$\Pr_{a,b,c \in \mathbb{F}_q} [\Delta(u' + c \cdot T_{a,b}(u''), V) < \delta] \leq \frac{K}{q}, \quad (8)$$

where  $K = 8/\epsilon^4$ .

**Proof.** Set  $\bar{\delta} = \delta + \epsilon$ . Note that

$$\bar{\delta} = \delta + \epsilon < J_\epsilon(J_\epsilon(\lambda)) < J_\epsilon(\lambda).$$

Suppose Equation (8) did not hold. Thus:

$$\Pr_{a,b,c \in \mathbb{F}_q} [\Delta(u' + c \cdot T_{a,b}(u''), V) < \bar{\delta} - \epsilon] > \frac{K}{q}.$$

Then with probability at least  $\frac{K}{2q}$  over the choice of  $(a, b)$ , we have that:

$$\Pr_{c \in \mathbb{F}_q} [\Delta(u' + c \cdot T_{a,b}(u''), V) < \bar{\delta} - \epsilon] > \frac{K}{2q}.$$

Fix such an  $(a, b) \in \mathbb{F}_q^2$ . Since  $K > 4/\epsilon^3$  and  $\bar{\delta} < J_\epsilon(J_\epsilon(\lambda))$ , we may apply Theorem 10 to  $u'$  and  $T_{a,b}(u'')$ . It tells us that there exist  $y, y^* \in V$  such that:

$$|\{\beta \in \mathbb{F}_q \mid u'(\beta) = y(\beta) \wedge u''(a\beta + b) = y^*(\beta)\}| \geq (1 - \bar{\delta})q, \quad (9)$$

which, after letting  $y^{**}(T) = y^*((\beta - b)/a)$ , can be rewritten as:

$$|\{\beta \in \mathbb{F}_q \mid u'(\beta) = y(\beta) \wedge u''(a\beta + b) = y^{**}(a\beta + b)\}| \geq (1 - \bar{\delta})q. \quad (10)$$

It is thus natural to consider the collection of polynomials close to  $u', u''$ :

$$\mathcal{L}' = \{f \in V \mid \Delta(u', f) \leq \bar{\delta}\},$$

$$\mathcal{L}'' = \{f \in V \mid \Delta(u'', f) \leq \bar{\delta}\},$$

as well as the collection of agreement sets:

$$\mathcal{F}' = \{A \subseteq \mathbb{F}_q \mid \text{for some } f \in \mathcal{L}' \text{ we have } A = \{\beta \in \mathbb{F}_q \mid f(\beta) = u'(\beta)\}\}.$$

$$\mathcal{F}'' = \{A \subseteq \mathbb{F}_q \mid \text{for some } f \in \mathcal{L}'' \text{ we have } A = \{\beta \in \mathbb{F}_q \mid f(\beta) = u''(\beta)\}\}.$$

By the Johnson bound, Theorem 2, (and since  $\bar{\delta} < J_\epsilon(\lambda)$ ), we have that

$$|\mathcal{L}'|, |\mathcal{L}''|, |\mathcal{F}'|, |\mathcal{F}''| < 1/\epsilon.$$

Equation (10) and the discussion before it tells us that with probability at least  $\frac{K}{2q}$  over the choice of  $(a, b) \in \mathbb{F}_q^2$ , there exists some  $A' \in \mathcal{F}'$  and some  $A'' \in \mathcal{F}''$  such that

$$|A' \cap \frac{1}{a}(A'' - b)| \geq (1 - \bar{\delta})q.$$

By averaging, this means that there must exist some  $A' \in \mathcal{F}'$  and  $A'' \in \mathcal{F}''$  such that with probability at least  $\frac{\epsilon^2 K}{2q}$  over the choice of  $(a, b) \in \mathbb{F}_q^2$ ,

$$|A' \cap \frac{1}{a}(A'' - b)| \geq (1 - \bar{\delta})q. \quad (11)$$

We will use this to deduce that either  $A'$  or  $A''$  must be big. For each  $r \in A''$ , let  $X_r$  denote the indicator random variable for the event that  $(r - b)/a \in A'$ . Let  $X = \sum_{r \in A''} X_r$ . Note that

$$X = |A' \cap \frac{1}{a}(A'' - b)|.$$

It is easy to see that  $\mathbf{E}[X_r] = |A'|/q$ , and so:

$$\mathbf{E}[X] = \frac{|A'| \cdot |A''|}{q} = \mu.$$

Furthermore, the  $X_r$  are pairwise independent, and thus the variance of  $X$  is bounded by:

$$\mathbf{Var}[X] \leq 4 \frac{|A''||A'|}{q} \leq 4q.$$

Thus:

$$\Pr[X \geq \mu + 2t\sqrt{q}] \leq \frac{1}{t^2}.$$

If  $|A'|, |A''|$  are such that  $|A'| \cdot |A''| \leq (1 - \bar{\delta} - \epsilon) \cdot q^2$ , then  $\mu \leq (1 - \bar{\delta} - \epsilon)q$ , and the above equation with  $t = \frac{\epsilon}{2}\sqrt{q}$  gives us that:

$$\Pr[X \geq (1 - \bar{\delta})q] \leq \frac{4}{\epsilon^2 q}.$$

This is a contradiction to Equation (11), since by the choice of  $K$ ,

$$\frac{\epsilon^2 K}{2q} < \frac{4}{\epsilon^2 q}.$$

Thus we must have that:

$$|A'| \cdot |A''| > (1 - \bar{\delta} - \epsilon)q^2.$$

Recalling that  $A' \in \mathcal{F}'$  and  $A'' \in \mathcal{F}''$ , we conclude that

$$(1 - \delta')(1 - \delta'') > (1 - \delta - 2\epsilon),$$

a contradiction to our assumption on  $\delta', \delta''$ . ◀

## 6 A low-agreement analysis of the Polischuk–Spielman axis-parallel test

In this section, we use the tools we developed above to give improved versions of the Polischuk-Spielman robust low-degree test [18] in certain settings. Their result gives a way to robustly test proximity of a 2-dimensional function  $f : \mathbb{F}_q^2 \rightarrow \mathbb{F}_q$  to bivariate polynomials with individual degrees  $(d, \ell)$ . Our result shows that for  $\ell = O(1)$ , and for  $d = O(q)$ , the Polischuk-Spielman low-degree test works even in the presence of high noise: even if the test passes with some tiny probability  $\eta$ , it means that the underlying bivariate function has nontrivial agreement with some low degree bivariate polynomial.

The original Polischuk-Spielman analysis (improving on Arora-Safra [3]) allows  $d, \ell$  to both be  $\Omega(q)$ , but could only conclude something if the passing probability  $\eta$  was at least  $1/2$ . The very recent analysis of the Polischuk-Spielman test due to Chiesa et al. [13] allows  $\eta$  to be small, as in the result we obtain below, but the two results are incomparable (neither implies the other). The result of [13] works for  $d, \ell$  as large as  $O(\log q)$  whereas ours requires  $\ell = O(1)$  but allows  $d$  to be as large as  $\Omega(q)$ .

► **Theorem 14** (High-error soundness analysis of the Polischuk–Spielman test). *There exists a function  $f : \mathbb{N}^+ \times (0, 1) \times (0, 1)$  that, for each fixed  $\ell \in \mathbb{N}^+$ , satisfies  $f(\ell, \rho, \epsilon) \rightarrow 0$  as  $\rho \rightarrow 0$  and  $\epsilon \rightarrow 0$ , and for which the following holds.*

*Let  $d = \rho q$ . Suppose for each  $x \in \mathbb{F}_q$ , we have a degree  $\ell$  polynomial  $Q_x(Y)$ , and for each  $y \in \mathbb{F}_q$  we have a degree  $d$  polynomial  $P_y(X)$ . Suppose that for some non-trivial agreement parameter  $\eta > f(\ell, \rho, \epsilon)$  all these polynomials meet the following nontrivial agreement condition:*

$$\Pr_{x, y \in \mathbb{F}_q} [Q_x(y) = P_y(x)] \geq \eta. \tag{12}$$

Then there exists a bivariate polynomial  $R(X, Y)$  of individual degree  $(d, \ell)$  such that

$$\Pr_{x, y \in \mathbb{F}_q} [Q_x(y) = R(x, y)] \geq \eta - 2\epsilon,$$

$$\Pr_{x, y \in \mathbb{F}_q} [Q_x(y) = P_y(x) = R(x, y)] \geq C_{\epsilon, \ell} \cdot \eta,$$

where  $C_{\epsilon, \ell} > 0$  is a constant only depending on  $\epsilon$  and  $\ell$ .

**Proof.** Our plan is to use Theorem 12 to deduce some information about  $Q_x$  and  $P_y$ . Let  $V \subseteq \mathbb{F}_q^q$  be the Reed-Solomon code of polynomials of degree at most  $d$ . Let  $\lambda = \Delta(V) = 1 - \rho$ .

Let  $u^*, u_1, \dots, u_\ell$  be functions from  $\mathbb{F}_q$  to  $\mathbb{F}_q$  such that:

$$Q_x(Y) = u^*(x) + u_1(x)Y + u_2(x)Y^2 + \dots + u_\ell(x)Y^\ell.$$

For each  $\alpha \in \mathbb{F}_q$ , define  $v^\alpha(X) = P_\alpha(X)$ .

The non-trivial agreement hypothesis of Equation (12) tells us that:

$$\Pr_{\alpha, x \in \mathbb{F}_q} [u^*(x) + u_1(x)\alpha + \dots + u_\ell(x)\alpha^\ell = v^\alpha(x)] \geq \eta.$$

Equivalently:

$$\mathbb{E}_{\alpha \in \mathbb{F}_q} [\Delta(u^* + \alpha u_1 + \alpha^2 u_2 + \dots + \alpha^\ell u_\ell, v^\alpha)] \leq 1 - \eta.$$

Set  $\delta = 1 - \eta + \epsilon$ . By an averaging argument, we get:

$$\Pr_{\alpha \in \mathbb{F}_q} [\Delta(u^* + \alpha u_1 + \alpha^2 u_2 + \dots + \alpha^\ell u_\ell, v^\alpha) < \delta] \geq \epsilon.$$

Let  $A$  be the set of  $\alpha$  for which the above event happens: thus  $|A| \geq \epsilon \cdot q$ .

We now apply Theorem 12. We need  $\delta < J_\epsilon^{[\ell]}(\lambda)$ , which we may assume by suitably setting  $f(\ell, \rho, \epsilon)$ . We get that there exist  $v^*, v_1, \dots, v_\ell \in V$  and a subset  $G \subseteq \mathbb{F}_q$  with  $|G| \geq (1 - \delta - \epsilon)q$  for all  $x \in G$ ,

$$v^*(x) = u^*(x), v_1(x) = u_1(x), \dots, v_\ell(x) = u_\ell(x).$$

Since  $v^*$  and the  $v_i$  are all in  $V$ , they are polynomials of degree at most  $d$ . Define

$$R(X, Y) = v^*(X) + v_1(X)Y + \dots + v_\ell(X)Y^\ell.$$

Rephrasing what we just concluded in terms of  $R$ , we get that for all  $x \in G$ :

$$R(x, Y) = Q_x(Y),$$

and thus:

$$\Pr_{x \in \mathbb{F}_q, y \in \mathbb{F}_q} [R(x, y) = Q_x(y)] \geq 1 - \delta - \epsilon = \eta - 2\epsilon.$$

Moreover, we conclude from Item 2 of Theorem 12 that for some  $c_{\epsilon, \ell}$  fraction of the  $\alpha \in A$ , we have:

$$v^\alpha = v^* + \alpha v_1 + \dots + \alpha^\ell v_\ell.$$

For any such  $\alpha$  where this identity holds, we get that:

$$R(x, \alpha) = v^\alpha(x) = P_\alpha(x),$$

and thus

$$\Pr_{x \in \mathbb{F}_q} [R(x, \alpha) = P_\alpha(x) = Q_x(\alpha)] \geq 1 - \delta = \eta - \epsilon > \eta/2.$$

This completes the proof of the theorem. ◀

## 7 Improved soundness for the Fast RS IOPP (FRI) protocol

In this section we describe how our prior results lead to a better analysis of the soundness of the FRI protocol of [9]. We start by recalling the notation needed to state our results, referring the reader to [9] for a detailed description of the protocol with its two phases, the COMMIT and QUERY sub-protocols.

### 7.1 Notation

We use the notation introduced in [9, Sections 3.4, 4.2.1]; let us briefly recall it. Our starting point is a function  $f^{(0)} : L^{(0)} \rightarrow \mathbb{F}$  where  $\mathbb{F}$  is a finite field of characteristic 2 and size  $2^n$ , the evaluation domain  $L^{(0)} \subset \mathbb{F}$  is an affine space over the two element field  $\mathbb{F}_2$ , i.e.,  $L^{(0)}$  is a coset of an additive subgroup of  $\mathbb{F}$ , and  $|L^{(0)}| = 2^{k^{(0)}}$  which means that  $k^{(0)} = \dim(L^{(0)})$ . We assume the target rate is  $\rho = 2^{-\mathcal{R}}$  for some positive integer  $\mathcal{R}$ . The COMMIT phase of the FRI protocol involves  $r = k^{(0)} - \mathcal{R}$  rounds. For  $i > 0$ , during the  $i$ th round the verifier sends a uniformly random  $x^{(i)} \in \mathbb{F}$  and the prover responds with a function  $f^{(i+1)} : L^{(i+1)} \rightarrow \mathbb{F}$  where  $L^{(i+1)}$  is an affine space of dimension  $k^{(i+1)} = k^{(i)} - 1$  (and size  $2^{k^{(i+1)}}$ ) defined by  $L^{(i+1)} = q^{(i)}(L^{(i)})$  where  $q^{(i)}$  is a linearized polynomial of degree 2 that is a subspace polynomial of a space  $L_0^{(i)}$  such that  $L^{(i)}$  can be partitioned into additive cosets of  $L_0^{(i)}$ . Let  $\mathcal{S}^{(i)}$  denote the set of cosets of  $L_0^{(i)}$  contained in  $L^{(i)}$ .

For  $f, g : L^{(i)} \rightarrow \mathbb{F}$  let  $\Delta^{(i)}(f, g)$  be the block-wise distance between  $f, g$  (cf. [9, Definition 3.2]), defined as the fraction of cosets of  $L_0^{(i)}$  on which  $f$  and  $g$  differ,

$$\Delta^{(i)}(f, g) \triangleq \Pr_{S \in \mathcal{S}^{(i)}} [f|_S \neq g|_S]$$

where  $f|_S$  is the restriction of  $f$  to  $S$  (and  $g|_S$  is similarly defined) and equality above is in the space  $\mathbb{F}^S$ . Notice  $\Delta^{(i)}(f, g) \geq \Delta(f, g)$ . For a set of functions  $V \subset \mathbb{F}^{L^{(i)}}$  let  $\Delta^{(i)}(f, V) = \min \{ \Delta^{(i)}(f, v) \mid v \in V \}$ .

### 7.2 Statement of results

The following is the main theorem from [9], and we improve its soundness in Theorem 16, stated after it.

► **Theorem 15 (FRI – main properties).** *The following properties hold when the FRI protocol is invoked on oracle  $f^{(0)} : L^{(0)} \rightarrow \mathbb{F}$  with rate  $\rho = 2^{-\mathcal{R}}$  for  $\mathcal{R} \in \mathbb{N}^+$  such that  $\rho|L^{(0)}| > 16$ :*

1. **Completeness** *If  $f^{(0)} \in \text{RS}^{(0)} \triangleq \text{RS}[\mathbb{F}, L^{(0)}, \rho = 2^{-\mathcal{R}}]$  and  $f^{(1)}, \dots, f^{(r)}$  are computed by the prover specified in the COMMIT phase, then the FRI verifier outputs accept with probability 1.*
2. **Soundness** *Suppose  $\delta^{(0)} \triangleq \Delta^{(0)}(f^{(0)}, \text{RS}^{(0)}) > 0$ . Then with probability at least*

$$1 - \frac{3|L^{(0)}|}{|\mathbb{F}|} \tag{13}$$

*over the randomness of the verifier during the COMMIT phase, and for any (adaptively chosen) prover oracles  $f^{(1)}, \dots, f^{(r)}$ , the QUERY protocol with repetition parameter  $\ell$  outputs accept with probability at most*

$$\left( 1 - \min \left\{ \delta^{(0)}, \frac{1 - 3\rho - 2/\sqrt{|L^{(0)}|}}{4} \right\} \right)^\ell \tag{14}$$

Consequently, the soundness of FRI is at least

$$s^-(\delta^{(0)}) \triangleq 1 - \left( \frac{3|L^{(0)}|}{|\mathbb{F}|} + \left( 1 - \min \left\{ \delta^{(0)}, \frac{1 - 3\rho - 2/\sqrt{|L^{(0)}|}}{4} \right\} \right)^\ell \right). \quad (15)$$

3. **Prover complexity** is  $O(|L^{(0)}|)$  arithmetic operations over  $\mathbb{F}$
4. **Verifier complexity** is  $O(\log |L^{(0)}|)$  arithmetic operations over  $\mathbb{F}$  for a single invocation of the QUERY phase; this also bounds communication and query complexity (measured in field elements).

We improve FRI soundness as follows:

► **Theorem 16** (FRI with improved soundness). *The following properties hold when the FRI protocol is invoked on oracle  $f^{(0)} : L^{(0)} \rightarrow \mathbb{F}$ ,  $|L^{(0)}| = k^{(0)}$ , with rate  $\rho = 2^{-\mathcal{R}}$ ,  $\mathcal{R} \in \mathbb{N}^+$  such that  $\rho|L^{(0)}| > 16$ :*

1. **Soundness** Suppose  $\delta^{(0)} \triangleq \Delta^{(0)}(f^{(0)}, \text{RS}^{(0)}) > 0$ . Then for any  $\epsilon > 0$ , with probability at least

$$1 - \frac{2k^{(0)}}{\epsilon^3 |\mathbb{F}|} \quad (16)$$

over the randomness of the verifier during the COMMIT phase, and for any (adaptively chosen) prover oracles  $f^{(1)}, \dots, f^{(r)}$ , the QUERY protocol with repetition parameter  $\ell$  outputs accept with probability at most

$$\left( 1 - \min \left\{ \delta^{(0)}, J_\epsilon(J_\epsilon(1 - \rho)) \right\} + \epsilon k^{(0)} \right)^\ell \quad (17)$$

Consequently, the soundness of FRI is at least

$$s^-(\delta^{(0)}) \triangleq 1 - \left( \frac{2k^{(0)}}{\epsilon^3 |\mathbb{F}|} + \left( 1 - \min \left\{ \delta^{(0)}, J_\epsilon(J_\epsilon(1 - \rho)) \right\} + \epsilon k^{(0)} \right)^\ell \right). \quad (18)$$

## 7.3 Proof of Theorem 16

Before presenting the proof of our main theorem for this section, we require a corollary of our prior results, which we state first. To state the corollary, we need more notation from [9]

### 7.3.1 More notation

Given  $x^{(i)} \in \mathbb{F}$ , the function  $f_{f^{(i)}, x^{(i)}}^{(i+1)} : L^{(i+1)} \rightarrow \mathbb{F}$  is that which is provided by the honest FRI prover upon input  $f^{(i)}$  and verifier randomness  $x^{(i)}$ . For  $s^{(i+1)} \in L^{(i+1)}$  and  $s_0^{(i)}, s_1^{(i)}$  the two roots of  $q^{(i)}(X) - s^{(i+1)}$ , we have

$$f_{f^{(i)}, x^{(i)}}^{(i+1)}(s^{(i+1)}) \triangleq P_{f^{(i)}, s_0^{(i)}, s_1^{(i)}}(s^{(i+1)}) \quad (19)$$

where  $P_{f^{(i)}, s_0^{(i)}, s_1^{(i)}}(X)$  is the polynomial that interpolates the two points  $\left( s_0^{(i)}, f^{(i)}\left( s_0^{(i)} \right) \right)$  and  $\left( s_1^{(i)}, f^{(i)}\left( s_1^{(i)} \right) \right)$ ; notice  $\deg\left( P_{f^{(i)}, s_0^{(i)}, s_1^{(i)}} \right) \leq 1$  and  $\left\{ s_0^{(i)}, s_1^{(i)} \right\}$  is a coset of  $L_0^{(i)}$ , denoted  $S_{s^{(i+1)}}^{(i)}$ .

### 7.3.2 A corollary of Theorem 10

The following statement is a corollary of Theorem 10, as applied to a single round of the FRI protocol involving an honest prover.

► **Corollary 17.** *Suppose  $\delta^{(i)} \triangleq \Delta^{(i)}(f^{(i)}, \text{RS}^{(i)})$  satisfies  $\delta^{(i)} < J_\epsilon(J_\epsilon(1 - \rho))$ . Then*

$$\Pr_{x^{(i)} \in \mathbb{F}} \left[ \Delta \left( f_{f^{(i)}, x^{(i)}}^{(i+1)}, \text{RS}^{(i+1)} \right) \geq \delta^{(i)} - \epsilon \right] \leq \frac{2}{\epsilon^3 |\mathbb{F}|}. \quad (20)$$

**Proof.** Consider the space of functions  $U = U^{(i+1)} = \{f_{f^{(i)}, x^{(i)}}^{(i+1)} \mid x^{(i)} \in \mathbb{F}\} \subset \mathbb{F}^{L^{(i+1)}}$  that are supplied by the honest prover in response to the various verifier messages  $x^{(i)}$ . Let

$$u^* = f_{f^{(i)}, 0}^{(i+1)}, \quad u = f_{f^{(i)}, 1}^{(i+1)} - f_{f^{(i)}, 0}^{(i+1)}.$$

Since  $\deg(P_{f^{(i)}, s_0^{(i)}, s_1^{(i)}}) \leq 1$  for every pair  $\{s_0^{(i)}, s_1^{(i)}\} \in \mathcal{S}^{(i)}$  it follows that every  $u' = f_{f^{(i)}, x^{(i)}}^{(i+1)} \in U$  can be written as a linear combination of  $u^*, u$ ; specifically,  $f_{f^{(i)}, x^{(i)}}^{(i+1)} = u^* + x^{(i)} \cdot u$ . Let  $\bar{U}^{(i+1)} \subseteq U$  be the set of elements in  $U$  that have distance less than  $\delta^{(i)} - \epsilon$  to  $\text{RS}^{(i+1)}$ .

Assume by way of contradiction that  $|\bar{U}^{(i+1)}| > \frac{2}{\epsilon^3}$ . Then Theorem 10 implies the existence of  $v^*, v \in \text{RS}^{(i+1)}$  and a subset  $T \subset L^{(i+1)}$ ,  $\frac{|T|}{|L^{(i+1)}|} \geq 1 - \delta^{(i)}$ , such that  $v^*|_T = u^*|_T$  and  $v|_T = u|_T$ . Let  $Q^*(Y), Q(Y)$  be the polynomials interpolating  $v^*$  and  $v$  respectively. We have  $\deg(Q^*), \deg(Q) < \rho |L^{(i+1)}|$  because  $v^*, v \in \text{RS}^{(i+1)}$ . Let

$$\hat{Q}(X, Y) \triangleq Q^*(Y) + X \cdot Q(Y)$$

and notice that (i)  $\deg_X(\hat{Q}) < 2$ ,  $\deg_Y(\hat{Q}) < \rho |L^{(i+1)}|$  (ii)  $\hat{Q}(0, Y) = Q^*(Y)$ , (iii)  $\hat{Q}(1, Y) = Q(Y)$ .

Consider the polynomial  $R(X) \triangleq \hat{Q}(X, q^{(i)}(X))$ . We have

$$\deg(R) \leq 2 \cdot \deg_Y(\hat{Q}) - 1 < 2|L^{(i+1)}| = \rho |L^{(i)}|.$$

We claim that  $R$  agrees with  $f^{(i)}$  on  $\{S_{s^{(i+1)}}^{(i)} \mid s^{(i+1)} \in T\}$ . Indeed, for each  $s^{(i+1)} \in T$  let  $S_{s^{(i+1)}}^{(i)} = \{s_0^{(i)}, s_1^{(i)}\} \in \mathcal{S}^{(i)}$  be the pair of roots of the polynomial  $q^{(i)}(X) - s^{(i+1)}$ . By our assumption on  $T$ ,

$$\hat{Q}(0, s^{(i+1)}) = f_{f^{(i)}, 0}^{(i+1)}(s^{(i+1)}) \quad \text{and} \quad \hat{Q}(1, s^{(i+1)}) = f_{f^{(i)}, 1}^{(i+1)}(s^{(i+1)}).$$

The polynomials  $\hat{Q}(X, s^{(i+1)})$  and  $P_{f^{(i)}, s_0^{(i)}, s_1^{(i)}}(X)$  are both of degree less than 2 and they agree on the two points  $\{0, 1\}$ , hence they agree everywhere. It follows that

$$f^{(i)}(s_0^{(i)}) = \hat{Q}(s_0^{(i)}, s^{(i+1)}) = \hat{Q}(s_0^{(i)}, q^{(i)}(s_0^{(i)})) = R(s_0^{(i)})$$

and similarly  $f^{(i)}(s_1^{(i)}) = R(s_1^{(i)})$ . Therefore,  $R$  and  $f^{(i)}$  agree on  $T$ , as claimed.

We have established  $\Delta^{(i)}(f^{(i)}, \text{RS}^{(i)}) \leq 1 - \frac{|T|}{|L^{(i+1)}|} \leq \delta^{(i)}$  and this contradicts our assumption. Therefore  $|U^{(i+1)}| \leq \frac{2}{\epsilon^3 |\mathbb{F}|}$ , as claimed. ◀



### 7.3.3 Proof of improved soundness

Armed with Corollary 17 we move on to the proof of the main theorem of this section.

**Proof of Theorem 16.** Let  $E^{(i)}$  be the “bad” event that  $f_{f^{(i)},x^{(i)}}^{(i+1)} \in \bar{U}^{(i+1)}$ ; in words,  $E^{(i)}$  is the event that  $\Delta\left(f_{f^{(i)},x^{(i)}}^{(i+1)}, \text{RS}^{(i+1)}\right) \leq \delta^{(i)} - \epsilon$ . Corollary 17 implies that  $\Pr[E^{(i)}] \leq \frac{2}{\epsilon^3|\mathbb{F}|}$ . (When  $\delta^{(i)} < \frac{1-\rho}{2}$  the stronger bound  $\Pr[E^{(i)}] \leq \frac{1}{\epsilon|\mathbb{F}|}$  holds.) By the union bound

$$\Pr\left[\bigvee_{i=0}^{r-1} E^{(i)}\right] \leq \frac{2r}{\epsilon^3|\mathbb{F}|} \leq \frac{2k^{(0)}}{\epsilon^3|\mathbb{F}|} \quad (21)$$

We continue our analysis assuming no such event holds. Let  $f^{(0)}, \dots, f^{(r)}$  be the sequence of functions sent by the prover, which is not necessarily honest. Recall that during the QUERY phase of the FRI protocol, the verifier selects a random  $s^{(0)} \in L^{(0)}$  and this defines a sequence  $s^{(0)}, \dots, s^{(r)}$  inductively by using the rule  $s^{(i+1)} = q^{(i)}(s^{(i)})$  for  $i \in \{1, \dots, r\}$ ; Recall  $S_{s^{(i+1)}}^{(i)} \in \mathcal{S}^{(i)}$  is the coset containing the two roots of the polynomial  $q^{(i)}(X) - s^{(i+1)}$ , and one of them is  $s^{(i)}$ . The test associated with  $s^{(0)}$  accepts iff Equation (19) holds for all  $i \in \{0, \dots, r-1\}$  and additionally  $f^{(r)}$  is a constant function; we assume it by associating the constant function with the first entry of  $f^{(r)}$ .

For the sake of analysis, consider the directed graph in which an edge appears from  $s^{(i)}$  to  $s^{(i+1)}$  if and only if  $s^{(i)} \in S_{s^{(i+1)}}^{(i)}$ . This graph has  $r+1$  layers, and the vertices in the  $i$ th layer are the elements of  $L^{(i)}$ . For all nodes but for the root and leaves, the in-degree is 2; all non-root nodes have out-degree is 1, making the graph a directed tree (we direct edges from leaves to root). A single invocation of the QUERY phase involves selecting a leaf  $s^{(0)}$  and performing the sequence of tests along the path from  $s^{(0)}$  to the top layer of the graph (which corresponds to  $L^{(r)}$ ).

Call a vertex  $s^{(i)}$  *bad* if Equation (19) fails to hold for  $s^{(i)}$  and  $S_{s^{(i)}}^{(i-1)}$ ; all other vertices are called *good*. A QUERY test rejects if and only if the path examined by it contains a bad vertex. To analyze the rejection probability of the test, it will be simpler to consider only the last such bad vertex along a path. To this end, we shall modify the sequence of functions  $f^{(1)}, \dots, f^{(r-1)}$  (but not  $f^{(0)}$  and  $f^{(r)}$ ) in a way that may change some bad vertices into good ones, but will not make a good vertex bad. We will then analyze the rejection probability of a QUERY test applied to the modified set of functions.

Working top down with  $i = r, \dots, 2$  in decreasing order, for each bad vertex  $s^{(i)} \in L^{(i)}$ , we modify the entries in the sub-tree whose root is  $s^{(i)}$ , as follows. Let  $L_{s^{(i)}}^{(j)}$  be the set of vertices in layer  $j$  that have a path to  $s^{(i)}$ . For  $j \in \{0, \dots, i-2\}$ , in increasing order, set

$$f^{(j+1)}|_{L_{s^{(i)}}^{(j+1)}} \triangleq f_{f^{(j)},x^{(j)}}^{(j+1)}.$$

This modification process may change the entries of  $f^{(1)}, \dots, f^{(r-1)}$  but does not change neither  $f^{(0)}$ , nor  $f^{(r)}$  because  $0 \leq j < r-1$  and we only modify entries in layer  $j+1$ . Crucially, the probability of rejecting during the QUERY phase does not increase as a result of this modification, because the modification does not turn a good vertex into a bad one and hence the set of post-modification bad vertices is a subset of the pre-modification bad vertices.

Consider the sequence of modified functions  $f^{(0)}, \dots, f^{(r)}$ . Let  $\beta^{(i)}$  denote the fraction of bad vertices in  $L^{(i)}$ . As said earlier, the probability of rejection during a single QUERY invocation is precisely the probability that a path originating in a random leaf passes through a bad vertex. After our modification process, the set of leaves that lead to distinct bad vertices, are distinct, and along a path there is at most one bad vertex. Hence, the probability

that the FRI verifier rejects on a single invocation of the QUERY protocol is precisely  $\sum_{i=1}^r \beta^{(i)}$ . All that remains is to bound this sum from below, as done next.

► **Claim 18.** *If  $E^{(i)}$  does not hold, then*

$$\beta^{(i+1)} \geq \delta^{(i)} - \delta^{(i+1)} - \epsilon$$

**Proof.** Assuming  $E^{(i)}$  does not hold, Corollary 17 implies

$$\Delta\left(f_{f^{(i)}, x^{(i)}}^{(i+1)}, \text{RS}^{(i+1)}\right) \geq \delta^{(i)} - \epsilon$$

By the properties of the modification process,  $f^{(i+1)}(s^{(i+1)}) = f_{f^{(i)}, x^{(i)}}^{(i+1)}$  for every  $s^{(i+1)}$  that is not bad. So

$$\Delta\left(f^{(i+1)}, \text{RS}^{(i+1)}\right) \geq \delta^{(i)} - \epsilon - \beta^{(i+1)}$$

or, rearranging,

$$\beta^{(i+1)} \geq \delta^{(i)} - \Delta\left(f_{f^{(i)}, x^{(i)}}^{(i+1)}, \text{RS}^{(i+1)}\right) - \epsilon$$

The claim follows because  $\delta^{(i+1)} \triangleq \Delta\left(f^{(i+1)}, \text{RS}^{(i+1)}\right) \geq \Delta\left(f_{f^{(i)}, x^{(i)}}^{(i+1)}, \text{RS}^{(i+1)}\right)$ . ◀

We continue with the proof. By assumption  $\delta^{(r)} = 0$  and  $f^{(0)}$  is unchanged by the modification process, so

$$\delta^{(0)} = \delta^{(0)} - \delta^{(r)} = \sum_{i=0}^{r-1} \delta^{(i)} - \delta^{(i+1)}$$

Applying Claim 18 to the rightmost term above we conclude that whenever no event  $E^{(i)}$  holds (cf. Equation (21)), then the probability of the verifier rejecting during a single invocation of the QUERY phase is at least  $\sum_{i=1}^r \beta^{(i)} \geq \delta^{(0)} - r\epsilon$ . This completes the proof. ◀

---

## References

- 1 Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligo: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 24th ACM Conference on Computer and Communications Security*, October 2017.
- 2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in FOCS '92.
- 3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in FOCS '92.
- 4 Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003. Preliminary version appeared in STOC '97.
- 5 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, 1991.
- 6 László Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, SFCS '90, pages 16–25, 1990.

- 7 László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988. doi:10.1016/0022-0000(88)90028-1.
- 8 Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046, 2018. Available at <https://eprint.iacr.org/2018/046>.
- 9 Eli Ben-Sasson, Iddo Bentov, Ynon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018. URL: <https://eccc.weizmann.ac.il/report/2017/134>.
- 10 Eli Ben-Sasson, Alessandro Chiesa, Michael A. Forbes, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. On probabilistic checking in perfect zero knowledge. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:156, 2016. URL: <http://eccc.hpi-web.de/report/2016/156>.
- 11 Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, and Madars Virza. Quasilinear-size zero knowledge from linear-algebraic PCPs. In *Proceedings of the 13th Theory of Cryptography Conference, TCC '16*, pages 33–64, 2016.
- 12 Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 31–60, 2016. doi:10.1007/978-3-662-53644-5\_2.
- 13 Alessandro Chiesa, Peter Manohar, and Igor Shinkar. On axis-parallel tests for tensor product codes. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh Srinivas Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 39:1–39:22. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.39.
- 14 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version appeared in STOC '85.
- 15 Venkatesan Guruswami. Algorithmic results in list decoding. *Foundations and Trends in Theoretical Computer Science*, 2(2), 2006. doi:10.1561/0400000007.
- 16 Prahladh Harsha and Madhu Sudan. Small PCPs with low query complexity. *Computational Complexity*, 9(3–4):157–201, Dec 2000. Preliminary version in STACS '01.
- 17 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.
- 18 Alexander Polishchuk and Daniel A. Spielman. Nearly-linear size holographic proofs. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, STOC '94, pages 194–203, 1994.
- 19 Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 793–802. ACM, 2013.



# On the Complexity of the Cayley Semigroup Membership Problem

Lukas Fleischer<sup>1</sup>

FMI, University of Stuttgart  
Universitätsstraße 38, 70569 Stuttgart, Germany  
fleischer@fmi.uni-stuttgart.de

---

## Abstract

We investigate the complexity of deciding, given a multiplication table representing a semigroup  $S$ , a subset  $X$  of  $S$  and an element  $t$  of  $S$ , whether  $t$  can be expressed as a product of elements of  $X$ . It is well-known that this problem is NL-complete and that the more general *Cayley groupoid membership problem*, where the multiplication table is not required to be associative, is P-complete. For groups, the problem can be solved in deterministic log-space which raised the question of determining the exact complexity of this variant. Barrington, Kadau, Lange and McKenzie showed that for Abelian groups and for certain solvable groups, the problem is contained in the complexity class FOLL and they concluded that these variants are not hard for any complexity class containing PARITY. The more general case of arbitrary groups remained open. In this work, we show that for both groups and for commutative semigroups, the problem is solvable in  $\text{qAC}^0$  (quasi-polynomial size circuits of constant depth with unbounded fan-in) and conclude that these variants are also not hard for any class containing PARITY. Moreover, we prove that NL-completeness already holds for the classes of 0-simple semigroups and nilpotent semigroups. Together with our results on groups and commutative semigroups, we prove the existence of a natural class of finite semigroups which generates a variety of finite semigroups with NL-complete Cayley semigroup membership, while the Cayley semigroup membership problem for the class itself is not NL-hard. We also discuss applications of our technique to FOLL.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Problems, reductions and completeness, Theory of computation  $\rightarrow$  Circuit complexity

**Keywords and phrases** subsemigroup, multiplication table, generators, completeness, quasi-polynomial-size circuits, FOLL

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.25

**Acknowledgements** I would like to thank Armin Weiß for several interesting and inspiring discussions, and for pointing out that  $\text{qAC}^0$  is not contained within P/poly. I would also like to thank Samuel Schlesinger for comments which led to an improved presentation of the proof of Proposition 3 and for pointing out that results on the average sensitivity of bounded-depth circuits can be used to show that FOLL is not contained within  $\text{qAC}^0$ . Moreover, I am grateful to the anonymous referees for providing helpful comments that improved the paper.

## 1 Introduction

The *Cayley groupoid membership problem* (sometimes also called the *generation problem*) asks, given a multiplication table representing a groupoid  $G$ , a subset  $X$  of  $G$  and an element  $t$  of  $G$ , whether  $t$  can be expressed as a product of elements of  $X$ . In 1976, Jones and Laaser

---

<sup>1</sup> This work was supported by the DFG grant DI 435/5-2.



showed that this problem is P-complete [20]. Barrington and McKenzie later studied natural subproblems and connected them to standard subclasses of P [10].

When restricting the set of valid inputs to inputs with an associative multiplication table, the problem becomes NL-complete [21]. We will call this variant of the problem the *Cayley semigroup membership problem* and analyze its complexity when further restricting the semigroups encoded by the input. For a class of finite semigroups  $\mathbf{V}$ , the *Cayley semigroup membership problem for  $\mathbf{V}$*  is formally defined as follows.

CSM( $\mathbf{V}$ )	
<b>Input:</b>	The Cayley table of a semigroup $S \in \mathbf{V}$ , a set $X \subseteq S$ and an element $t \in S$
<b>Question:</b>	Is $t$ in the subsemigroup of $S$ generated by $X$ ?

The motivation for investigating this problem is two-fold. Firstly, there is a direct connection between the Cayley semigroup membership problem and decision problems for regular languages: a language  $L \subseteq \Sigma^+$  is regular if and only if there exist a finite semigroup  $S$ , a morphism  $\varphi: \Sigma^+ \rightarrow S$  and a set  $P \subseteq S$  such that  $L = \varphi^{-1}(P)$ . Thus, morphisms to finite semigroups can be seen as a way of encoding regular languages. For encoding such a semigroup, specifying the multiplication table is a natural choice. Deciding emptiness of a regular language represented by a morphism  $\varphi: \Sigma^+ \rightarrow S$  to a finite semigroup  $S$  and a set  $P \subseteq S$  boils down to checking whether any of the elements from the set  $P$  is contained in the subsemigroup of  $S$  generated by the images of the letters of  $\Sigma$  under  $\varphi$ . Conversely, the Cayley semigroup membership problem is a special case of the emptiness problem for regular languages: an element  $t \in S$  is contained in the subsemigroup generated by a set  $X \subseteq S$  if and only if the language  $\varphi^{-1}(P)$  with  $\varphi: X^+ \rightarrow S, x \mapsto x$  and  $P = \{t\}$  is non-empty.

Secondly, we hope to get a better understanding of the connection between algebra and low-level complexity classes included in NL in a fashion similar to the results of [10]. In the past, several intriguing links between so-called *varieties of finite semigroups* and the computational complexity of algebraic problems for such varieties were made. For example, the fixed membership problem for a regular language was shown to be in  $\text{AC}^0$  if its syntactic monoid is aperiodic, in  $\text{ACC}^0$  if the syntactic monoid is solvable and  $\text{NC}^1$ -complete otherwise [8, 11]. It is remarkable that in most results of this type, both the involved complexity classes and the algebraic varieties are natural. On a language-theoretical level, varieties of finite semigroups correspond to subclasses of the regular languages closed under Boolean operations, quotients and inverse morphisms.

**Related Work.** We already mentioned the work of Jones and Laaser on the Cayley groupoid membership problem [20], the work of Jones, Lien and Laaser on the Cayley semigroup membership problem [21] and the work of Barrington and McKenzie on subproblems thereof [10]. The semigroup membership problem and its restrictions to varieties of finite semigroups was also studied for other encodings of the input, such as matrix semigroups [2, 4, 7] or transformation semigroups [5, 6, 12, 13, 14, 15, 18].

The group version of the Cayley semigroup membership problem (CSM( $\mathbf{G}$ ), using our notation) was first investigated by Barrington and McKenzie in 1991 [10]. They observed that the problem is in symmetric log-space, which has been shown to be the same as deterministic log-space by Reingold in 2008 [23], and suggested it might be complete for deterministic log-space. However, all attempts to obtain a hardness proof failed (in fact, their conjecture is shown to be false in this work). There was no progress in a long time until Barrington, Kadau, Lange and McKenzie showed that for Abelian groups and certain solvable groups, the problem lies in the complexity class FOLL and thus, cannot be hard for any complexity class containing PARITY in 2001 [9]. The case of arbitrary groups remained open.

**Our Contributions.** We generalize previous results on Abelian groups to arbitrary commutative semigroups. Then, using novel techniques, we show that the Cayley semigroup membership problem for the variety of finite groups  $\mathbf{G}$  is contained in  $\mathbf{qAC}^0$  and thus, cannot be hard for any class containing  $\mathbf{PARITY}$ . Our approach relies on the existence of succinct representations of group elements by algebraic circuits. More precisely, it uses the fact that every element of a group  $G$  can be computed by an algebraic circuit of size  $\mathcal{O}(\log^3 |G|)$  over any set of generators. Since in the Cayley semigroup membership problem, the algebraic structure is not fixed, we introduce so-called Cayley circuits, which are similar to regular algebraic circuits but expect the finite semigroup to be given as part of the input. We prove that these Cayley circuits can be simulated by sufficiently small unbounded fan-in Boolean circuits. We then use this kind of simulation to evaluate all Cayley circuits, up to a certain size, in parallel.

By means of a closer analysis and an extension of the technique used by Jones, Lien and Laaser in [21], we also show that the Cayley semigroup membership problem remains  $\mathbf{NL}$ -complete when restricting the input to 0-simple semigroups or to nilpotent semigroups.

Combining our results, we obtain that the Cayley semigroup membership problem for the class  $\mathbf{G} \cup \mathbf{Com}$ , which consists of all finite groups and all finite commutative semigroups, is decidable in  $\mathbf{qAC}^0$  (and thus not  $\mathbf{NL}$ -hard) while the Cayley semigroup membership problem for the minimal variety of finite semigroups containing  $\mathbf{G} \cup \mathbf{Com}$  is  $\mathbf{NL}$ -complete.

Finally, we discuss the extent to which our approach can be used to establish membership of Cayley semigroup membership variants to the complexity class  $\mathbf{FOLL}$ . Here, instead of simulating all circuits in parallel, we use an idea based on repeated squaring. This technique generalizes some of the main concepts used in [9].

## 2 Preliminaries

**Algebra.** A semigroup  $T$  is a *subsemigroup* of  $S$  if  $T$  is a subset of  $S$  closed under multiplication. The *direct product* of two semigroups  $S$  and  $T$  is the Cartesian product  $S \times T$  equipped with componentwise multiplication. A subsemigroup of a direct product is also called *subdirect product*. A semigroup  $T$  is a *quotient* of a semigroup  $S$  if there exists a surjective morphism  $\varphi: S \rightarrow T$ .

A *variety of finite semigroups* is a class of finite semigroups which is closed under finite subdirect products and under quotients. Since we are only interested in finite semigroups, we will henceforth use the term *variety* for a variety of finite semigroups. Note that in the literature, such classes of semigroups are often called *pseudovarieties*, as opposed to Birkhoff varieties which are also closed under infinite subdirect products. The following varieties play an important role in this paper:

- $\mathbf{G}$ , the class of all finite groups,
- $\mathbf{Ab}$ , the class of all finite Abelian groups,
- $\mathbf{Com}$ , the class of all finite commutative semigroups,
- $\mathbf{N}$ , the class of all finite nilpotent semigroups, i.e., semigroups where the only idempotent is a zero element.

The *join* of two varieties  $\mathbf{V}$  and  $\mathbf{W}$ , denoted by  $\mathbf{V} \vee \mathbf{W}$ , is the smallest variety containing both  $\mathbf{V}$  and  $\mathbf{W}$ . A semigroup  $S$  is *0-simple* if it contains a zero element  $0$  and if for each  $s \in S \setminus \{0\}$ , one has  $SsS = S$ . The class of finite 0-simple semigroups does not form a variety.

**Complexity.** We assume familiarity with standard definitions from circuit complexity. A function has *quasi-polynomial* growth if it is contained in  $2^{\mathcal{O}(\log^c n)}$  for some fixed  $c \in \mathbb{N}$ .

Throughout the paper, we consider the following unbounded fan-in Boolean circuit families:

- $AC^0$ , languages decidable by circuit families of depth  $\mathcal{O}(1)$  and polynomial size,
- $qAC^0$ , languages decidable by circuit families of depth  $\mathcal{O}(1)$  and quasi-polynomial size,
- FOLL, languages decidable by circuit families of depth  $\mathcal{O}(\log \log n)$  and polynomial size,
- $AC^1$ , languages decidable by circuit families of depth  $\mathcal{O}(\log n)$  and polynomial size,
- P/poly, languages decidable by circuit families of polynomial size (and unbounded depth).

We allow NOT gates but do not count them when measuring the depth or the size of a circuit. We will also briefly refer to the complexity classes  $ACC^0$ ,  $TC^0$ ,  $NC^1$ , L and NL.

It is known that the PARITY function cannot be computed by  $AC^0$ , FOLL or  $qAC^0$  circuits. This follows directly from Håstad's and Yao's famous lower bound results [19, 24], which state that the number of Boolean gates required for a depth- $d$  circuit to compute PARITY is exponential in  $n^{1/(d-1)}$ .

### 3 Hardness Results

Before looking at parallel algorithms for the Cayley semigroup membership problem, we establish two new NL-hardness results. To this end, we first analyze the construction already used by Jones, Lien and Laaser [21]. It turns out that the semigroups used in their reductions are 0-simple which leads to the following result.

► **Theorem 1.** *For a class containing all 0-simple semigroups, the Cayley semigroup membership problem is NL-complete.*

**Proof.** To keep the proof self-contained, we briefly describe the reduction from the connectivity problem for directed graphs (henceforth called STCONN) to the Cayley semigroup membership problem given in [21].

Let  $G = (V, E)$  be a directed graph. We construct a semigroup on the set  $S = V \times V \cup \{0\}$  where 0 is a zero element and the multiplication rule for the remaining elements is

$$(v, w) \cdot (x, y) = \begin{cases} (v, y) & \text{if } w = x, \\ 0 & \text{otherwise.} \end{cases}$$

By construction, the subsemigroup of  $S$  generated by  $E \cup \{(v, v) \mid v \in V\}$  contains an element  $(s, t)$  if and only if  $t$  is reachable from  $s$  in  $G$ . To see that the semigroup  $S$  is 0-simple, note that for pairs of arbitrary elements  $(v, w) \in V \times V$  and  $(x, y) \in V \times V$ , one has  $(x, v)(v, w)(w, y) = (x, y)$ , which implies  $S(v, w)S = S$ . ◀

In order to prove NL-completeness for another common class of semigroups, we need a slightly more advanced construction reminiscent of the “layer technique”, which is usually used to show that STCONN remains NL-complete when the inputs are acyclic graphs.

► **Theorem 2.** *CSM( $\mathbf{N}$ ) is NL-complete (under  $AC^0$  many-one reductions).*

**Proof.** Following the proof of Theorem 1, we describe an  $AC^0$  reduction of STCONN to CSM( $\mathbf{N}$ ).

Let  $G = (V, E)$  be a directed graph with  $n$  vertices. We construct a semigroup on the set  $S = V \times \{1, \dots, n-1\} \times V \cup \{0\}$  where 0 is a zero element and the multiplication rule for the remaining elements is

$$(v, i, w) \cdot (x, j, y) = \begin{cases} (v, i+j, y) & \text{if } w = x \text{ and } i+j < n, \\ 0 & \text{otherwise.} \end{cases}$$



The subsemigroup of  $S$  generated by  $\{(v, 1, w) \mid v = w \text{ or } (v, w) \in E\}$  contains an element  $(s, n - 1, t)$  if and only if  $t$  is reachable (in less than  $n$  steps) from  $s$  in  $G$ . Clearly, the zero element is the only idempotent in  $S$ , so  $S$  is nilpotent. Also, it is readily verified that the reduction can be performed by an  $\text{AC}^0$  circuit family. ◀

## 4 Parallel Algorithms for Cayley Semigroup Membership

Algebraic circuits can be used as a succinct representation of elements in an algebraic structure. This idea will be the basis of the proof that  $\text{CSM}(\mathbf{G})$  is in  $\text{qAC}^0$ . Unlike in usual algebraic circuits, in the context of the Cayley semigroup membership problem, the algebraic structure is not fixed but given as part of the input. We will introduce so-called Cayley circuits to deal with this setting. Since these circuits will be used for the Cayley semigroup membership problem only, we confine ourselves to cases where the algebraic structure is a finite semigroup.

### 4.1 Cayley Circuits

A *Cayley circuit* is a directed acyclic graph with topologically ordered vertices such that each vertex has in-degree 0 or 2. In the following, to avoid technical subtleties when squaring an element, we allow multi-edges. The vertices of a Cayley circuit are called *gates*. The vertices with in-degree 0 are called *input gates* and vertices with in-degree 2 are called *product gates*. Each Cayley circuit also has a designated gate of out-degree 0, called the *output gate*. For simplicity, we assume that the output gate always corresponds to the maximal gate with regard to the vertex order. The *size* of a Cayley circuit  $\mathcal{C}$ , denoted by  $|\mathcal{C}|$ , is the number of gates of  $\mathcal{C}$ . An *input* to a Cayley circuit  $\mathcal{C}$  with  $k$  input gates consists of a finite semigroup  $S$  and elements  $x_1, \dots, x_k$  of  $S$ . Given such an input, the *value* of the  $i$ -th input gate is  $x_i$  and the value of a product gate, whose predecessors have values  $x$  and  $y$ , is the product  $x \cdot y$  in  $S$ . The *value of the circuit*  $\mathcal{C}$  is the value of its output gate. We will denote the value of  $\mathcal{C}$  under a finite semigroup  $S$  and elements  $x_1, \dots, x_k \in S$  by  $\mathcal{C}(S, x_1, \dots, x_k)$ .

A Cayley circuit can be seen as a circuit in the usual sense: the finite semigroup  $S$  and the input gate values are given as part of the input and the functions computed by product gates map a tuple, consisting of semigroup  $S$  and two elements of  $S$ , to another element of  $S$ . We say that a Cayley circuit with  $k$  input gates can be *simulated* by a family of unbounded fan-in Boolean circuits  $(\mathcal{C}_n)_{n \in \mathbb{N}}$  if, given the encodings of a finite semigroup  $S$  and of elements  $x_1, \dots, x_k$  of  $S$  of total length  $n$ , the circuit  $\mathcal{C}_n$  computes the encoding of  $\mathcal{C}(S, x_1, \dots, x_k)$ . For a semigroup  $S$  with  $N$  elements, we assume that the elements of  $S$  are encoded by the integers  $\{0, \dots, N - 1\}$  such that the encoding of a single element uses  $\lceil \log N \rceil$  bits. The semigroup itself is given as a multiplication table with  $N^2$  entries of  $\lceil \log N \rceil$  bits each.

► **Proposition 3.** *Let  $\mathcal{C}$  be a Cayley circuit of size  $m$ . Then,  $\mathcal{C}$  can be simulated by a family of unbounded fan-in constant depth Boolean circuits  $(\mathcal{C}_n)_{n \in \mathbb{N}}$  of size at most  $n^m$ .*

**Proof.** Let  $\mathcal{C}$  be a Cayley circuit with  $k$  input gates and  $m - k$  product gates. We want to construct a Boolean circuit which can be used for all finite semigroups  $S$  with a fixed number of elements  $N$ . The input to such a circuit consists of  $n = (N^2 + k) \lceil \log N \rceil$  bits.

For a fixed vector  $(y_1, \dots, y_m) \in S^m$ , one can check using a single AND gate (and additional NOT gates at some of the incoming wires) whether  $(y_1, \dots, y_m)$  corresponds to the sequence of values occurring at the gates of  $\mathcal{C}$  under the given inputs. To this end, for each gate  $i \in \{1, \dots, m\}$  of  $\mathcal{C}$ , we add  $\lceil \log N \rceil$  incoming wires to this AND gate: if the  $i$ -th gate of  $\mathcal{C}$  is an input gate, we feed the bits of the corresponding input value into the AND gate,

complementing the  $j$ -th bit if the  $j$ -th bit of  $y_i$  is zero. If the  $i$ -th gate is a product gate and has incoming wires from gates  $\ell$  and  $r$ , we connect the entry  $(y_\ell, y_r)$  of the multiplication table to the AND gate, again complementing bits corresponding to 0-bits of  $y_i$ .

To obtain a Boolean circuit simulating  $\mathcal{C}$ , we put such AND gates for all vectors of the form  $(y_1, \dots, y_m) \in S^m$  in parallel. In a second layer, we create  $\lceil \log N \rceil$  OR gates and connect the AND gate for a vector  $(y_1, \dots, y_m)$  to the  $j$ -th OR gate if and only if the  $j$ -th bit of  $y_m$  is one. The idea is that exactly one of the AND gates — the gate corresponding to the vector of correct guesses of the gate values of  $\mathcal{C}$  — evaluates to 1 and the corresponding output value  $y_m$  then occurs as output value of the OR gates.

This circuit has depth 2 and size  $N^m + \lceil \log N \rceil \leq n^m$ . ◀

## 4.2 The Poly-Logarithmic Circuits Property

When analyzing the complexity of CSM(**Ab**), Barrington et al. introduced the so-called *logarithmic power basis property*. A class of semigroups has the logarithmic power basis property if any set of generators  $X$  for a semigroup  $S$  of cardinality  $N$  from the family has the property that every element of  $S$  can be written as a product of at most  $\log(N)$  many powers of elements of  $X$ . In [9], it was shown that the class of Abelian groups has the logarithmic power basis property. Using a different technique, this result can easily be extended to arbitrary commutative semigroups.

► **Lemma 4.** *The variety **Com** has the logarithmic power basis property.*

**Proof.** Suppose that  $S$  is a commutative semigroup of size  $N$  and let  $X$  be a set of generators for  $S$ . Let  $y \in S$  be an arbitrary element. We choose  $k \in \mathbb{N}$  to be the smallest value such that there exist elements  $x_1, \dots, x_k \in X$  and integers  $i_1, \dots, i_k \in \mathbb{N}$  with  $y = x_1^{i_1} \cdots x_k^{i_k}$ . Assume, for the sake of contradiction, that  $k > \log(N)$ .

The power set  $\mathcal{P}(\{1, \dots, k\})$  forms a semigroup when equipped with set union as binary operation. Consider the morphism  $h: \mathcal{P}(\{1, \dots, k\}) \rightarrow S$  defined by  $h(\{j\}) = x_j^{i_j}$  for all  $j \in \{1, \dots, k\}$ . This morphism is well-defined because  $S$  is commutative.

Since  $|\mathcal{P}(\{1, \dots, k\})| = 2^k > 2^{\log(N)} = |S|$ , we know by the pigeon hole principle that there exist two sets  $K_1, K_2 \subseteq \{1, \dots, k\}$  with  $K_1 \neq K_2$  and  $h(K_1) = h(K_2)$ . We may assume, without loss of generality, that there exists some  $j \in K_1 \setminus K_2$ . Now, because

$$y = h(\{1, \dots, k\}) = h(K_1)h(\{1, \dots, k\} \setminus K_1) = h(K_2)h(\{1, \dots, k\} \setminus K_1)$$

and since neither  $K_2$  nor  $\{1, \dots, k\} \setminus K_1$  contain  $j$ , we know that  $y$  can be written as a product of powers of elements  $x_i$  with  $1 \leq i \leq k$  and  $i \neq j$ , contradicting the choice of  $k$ . ◀

For the analysis of arbitrary groups, we introduce a more general concept. It is based on the idea that algebraic circuits (Cayley circuits with fixed inputs) can be used for succinct representations of semigroup elements.

► **Example 5.** Let  $e \in \mathbb{N}$  be a positive integer. Then, one can construct a Cayley circuit of size at most  $2 \lceil \log e \rceil$  which computes, given a finite semigroup  $S$  and an element  $x \in S$  as input, the power  $x^e$  in  $S$ . If  $e = 1$ , the circuit only consists of the input gate. If  $e$  is even, the circuit is obtained by taking the circuit for  $e/2$ , adding a product gate and creating two edges from the output gate of the circuit for  $e/2$  to the new gate. If  $e$  is odd, the circuit is obtained by taking the circuit for  $e - 1$  and connecting it to a new product gate. In this case, the second incoming edge for the new gate comes from the input gate.

A class of semigroups has the *poly-logarithmic circuits property* if there exists a constant  $c \in \mathbb{N}$  such that for each semigroup  $S$  of cardinality  $N$  from the class, for each subset  $X$  of  $S$  and for each  $y$  in the subsemigroup generated by  $X$ , there exists a Cayley circuit  $\mathcal{C}$  of size  $\log^c(N)$  with  $k$  input gates and there exist  $x_1, \dots, x_k \in X$  such that  $\mathcal{C}(S, x_1, \dots, x_k) = y$ .

► **Proposition 6.** *Let  $\mathbf{V}$  be a family of semigroups which is closed under subsemigroups and has the logarithmic power basis property. Then  $\mathbf{V}$  has the poly-logarithmic circuits property.*

**Proof.** Let  $X$  be a subset of a semigroup  $S$  of cardinality  $N$ . Let  $y$  be in the subsemigroup generated by  $X$ . Then, we have  $y = x_1^{i_1} \cdots x_k^{i_k}$  for some  $x_1, \dots, x_k \in X$  with  $k \leq \log(N)$  and  $i_1, \dots, i_k \in \mathbb{N}$ . By the pigeon hole principle, we may assume without loss of generality that  $1 \leq i_1, \dots, i_k \leq N$ . Using the method from Example 5, one can construct Cayley circuits  $\mathcal{C}_1, \dots, \mathcal{C}_k$  of size at most  $2 \lceil \log N \rceil$  such that  $\mathcal{C}_j(S, x) = x^{i_j}$  for all  $j \in \{1, \dots, k\}$  and  $x \in S$ . Using  $k - 1$  additional product gates, these circuits can be combined to a single circuit  $\mathcal{C}$  with  $\mathcal{C}(S, x_1, \dots, x_k) = x_1^{i_1} \cdots x_k^{i_k} = y$ .

In total, the resulting circuit consists of  $k \cdot 2 \lceil \log N \rceil + k - 1 < 5 \log^2(N)$  gates. ◀

Let  $G$  be a finite group and let  $X$  be a subset of  $G$ . A sequence  $(g_1, \dots, g_\ell)$  of elements of  $G$  is a *straight-line program over  $X$*  if for each  $i \in \{1, \dots, \ell\}$ , we have  $g_i \in X$  or  $g_i = g_p^{-1}$  or  $g_i = g_p g_q$  for some  $p, q < i$ . The number  $\ell$  is the *length* of the straight-line program and the elements of the sequence are said to be *generated* by the straight-line program. The following result by Babai and Szemerédi [7] is commonly known as *Reachability Lemma*.

► **Lemma 7 (Reachability Lemma).** *Let  $G$  be a finite group and let  $X$  be a set of generators of  $G$ . Then, for each element  $t \in G$ , there exists a straight-line program over  $X$  generating  $t$  which has length at most  $(\log |G| + 1)^2$ .*

The proof of this lemma is based on a technique called “cube doubling”. For details, we refer to [3]. It is now easy to see that groups admit poly-logarithmic circuits.

► **Lemma 8.** *The variety  $\mathbf{G}$  has the poly-logarithmic circuits property.*

**Proof.** Let  $G$  be a group of order  $N$ , let  $X$  be a subset of  $G$  and let  $y$  be an element in the subgroup of  $G$  generated by  $X$ . By Lemma 7, we know that there exists a straight-line program  $(g_1, \dots, g_\ell)$  over  $X$  with  $\ell \leq (\log(N) + 1)^2$  and  $g_\ell = y$ . We may assume that the elements  $g_1, \dots, g_\ell$  are pairwise distinct. It suffices to describe how to convert this straight-line program into a Cayley circuit  $\mathcal{C}$  and values  $x_1, \dots, x_k \in X$  such that  $\mathcal{C}(S, x_1, \dots, x_k) = y$ .

We start with an empty circuit and with  $k = 0$  and process the elements of the straight-line program left to right. For each element  $g_i$ , we add gates to the circuit. The output gate of the circuit obtained after processing the element  $g_i$  will be called the  *$g_i$ -gate*.

If the current element  $g_i$  is contained in  $X$ , we increment  $k$ , add a new input gate to the circuit and let  $x_k = g_i$ . If the current element  $g_i$  can be written as a product  $g_p g_q$  with  $p, q < i$ , we add a new product gate to the circuit and connect the  $g_p$ -gate as well as the  $g_q$ -gate to this new gate. If the current element  $g_i$  is an inverse  $g_p^{-1}$  with  $p < i$ , we take a circuit  $\mathcal{C}'$  with  $2 \lceil \log N \rceil$  gates and with  $\mathcal{C}'(G, x) = x^{N-1}$  for all  $x \in S$ . Such a circuit can be built by using the powering technique illustrated in Example 5. We add  $\mathcal{C}'$  to  $\mathcal{C}$ , replacing its input gate by an edge coming from the  $g_p$ -gate.

The resulting circuit has size at most  $(\log(N) + 1)^2 \cdot 2 \lceil \log N \rceil \leq 2(\log(N) + 1)^3$ . ◀

We will now show that for classes of semigroups with the poly-logarithmic circuits property, one can solve the Cayley semigroup membership problem in  $\mathbf{qAC}^0$ .

► **Theorem 9.** *Let  $\mathbf{V}$  be a class of semigroups with the poly-logarithmic circuits property. Then  $\text{CSM}(\mathbf{V})$  is in  $\text{qAC}^0$ .*

**Proof.** We construct a family of unbounded fan-in constant-depth Boolean circuits with quasi-polynomial size, deciding, given the multiplication table of a semigroup  $S \in \mathbf{V}$ , a set  $X \subseteq S$  and an element  $t \in S$  as inputs, whether  $t$  is in the subsemigroup generated by  $X$ .

Since  $\mathbf{V}$  has the poly-logarithmic circuits property, we know that, for some constant  $c \in \mathbb{N}$ , the element  $t$  is in the subsemigroup generated by  $X$  if and only if there exist a Cayley circuit  $\mathcal{C}$  of size  $\log^c(n)$  and inputs  $x_1, \dots, x_k \in X$  such that  $\mathcal{C}(S, x_1, \dots, x_k) = t$ . There are at most  $(\log^c(n) \cdot \log^c(n))^{\log^c(n)} = 2^{\log^c(n) \log(2c \log n)}$  different Cayley circuits of this size. Let us consider one of these Cayley circuits  $\mathcal{C}$ . Suppose that  $\mathcal{C}$  has  $k$  input gates. By Proposition 3, there exists a unbounded fan-in constant-depth Boolean circuit of size  $n^{\log^c n} = 2^{\log^{c+1} n}$  deciding on input  $S$  and elements  $x_1, \dots, x_k \in S$  whether  $\mathcal{C}(S, x_1, \dots, x_k) = t$ . There are at most  $n^k \leq n^{\log^c n} = 2^{\log^{c+1} n}$  possibilities of connecting (not necessarily all) input gates corresponding to the elements of  $X$  to this simulation circuit.

Thus, we can check for all Cayley circuits of the given size and all possible input assignments in parallel, whether the value of the corresponding circuit is  $t$ , and feed the results of all these checks into a single OR gate to obtain a quasi-polynomial-size Boolean circuit. ◀

In conjunction with Lemma 4 and Lemma 8, we immediately obtain the following corollary.

► **Corollary 10.** *Both  $\text{CSM}(\mathbf{G})$  and  $\text{CSM}(\mathbf{Com})$  are contained in  $\text{qAC}^0$ .*

As stated in the preliminaries, problems in  $\text{qAC}^0$  cannot be hard for any complexity class containing PARITY. Thus, we also obtain the following statement.

► **Corollary 11.** *Let  $\mathbf{V}$  be a class of semigroups with the poly-logarithmic circuits property, such as the variety of finite groups  $\mathbf{G}$  or the variety of finite commutative semigroups  $\mathbf{Com}$ . Then  $\text{CSM}(\mathbf{V})$  is not hard for any complexity class containing PARITY, such as  $\text{ACC}^0$ ,  $\text{TC}^0$ ,  $\text{NC}^1$ , L or NL.*

### 4.3 The Complexity Landscape of Cayley Semigroup Membership

Our hardness results and  $\text{qAC}^0$ -algorithms have an immediate consequence on algebraic properties of maximal classes of finite semigroups for which the Cayley semigroup membership problem can be decided in  $\text{qAC}^0$ . It relies on the following result, which can be seen as a consequence of [1] and the fact that the zero element in a semigroup is always central. For completeness, we provide a short and self-contained proof.

► **Proposition 12.** *The variety  $\mathbf{N}$  is included in  $\mathbf{G} \vee \mathbf{Com}$ .*

**Proof.** We show that every finite nilpotent semigroup is a quotient of a subdirect product of a finite group and a finite commutative semigroup. Note that in a finite nilpotent semigroup  $S$ , there exists an integer  $e \geq 0$  such that for each  $x \in S$ , the power  $x^e$  is the zero element. Let  $T = \{1, \dots, e\}$  be the commutative semigroup with the product of two elements  $i$  and  $j$  defined as  $\min\{i + j, e\}$ .

Let  $G$  be a finite group generated by the set  $X$  of non-zero elements of  $S$  such that no two products of less than  $e$  elements of  $X$  evaluate to the same element of  $G$ . Such a group exists because the free group over  $X$  is residually finite [22].

Let  $U$  be the subsemigroup of  $G \times T$  generated by  $\{(x, 1) \mid x \in X\}$ . Now, we define a mapping  $\varphi: U \rightarrow S$  as follows. Each element of the form  $(g, e)$  is mapped to zero. For every  $(g, \ell)$  with  $\ell < e$ , there exists, by choice of  $G$  and by the definition of  $U$ , a unique factorization

$g = x_1 \cdots x_\ell$  with  $x_1, \dots, x_\ell \in X$ . We map  $(g, \ell)$  to the product  $x_1 \cdots x_\ell$  evaluated in  $S$ . It is straightforward to verify that  $\varphi$  is a surjective morphism and thus,  $S$  is a quotient of  $U$ . ◀

► **Corollary 13.** *There exist two varieties  $\mathbf{V}$  and  $\mathbf{W}$  such that both  $\text{CSM}(\mathbf{V})$  and  $\text{CSM}(\mathbf{W})$  are contained in  $\text{qAC}^0$  (and thus not hard for any class containing PARITY) but  $\text{CSM}(\mathbf{V} \vee \mathbf{W})$  is NL-complete.*

The corollary is a direct consequence of the previous proposition, Corollary 10 and Theorem 2. As was observed in [9] already, Cayley semigroup problems seem to have “strange complexity”. The previous result makes this intuition more concrete and suggests that it is difficult to find “nice” descriptions of maximal classes of semigroups for which the Cayley semigroup membership problem is easier than any NL-complete problem.

#### 4.4 Connections to FOLL

In a first attempt to solve outstanding complexity questions related to the Cayley semigroup membership problem, Barrington et al. introduced the complexity class FOLL. The approach presented in the present paper is quite different. This raises the question of whether our techniques can be used to design FOLL-algorithms for Cayley semigroup membership. Note that FOLL and  $\text{qAC}^0$  are known to be incomparable, so we cannot use generic results from complexity theory to simulate  $\text{qAC}^0$  circuits using families of FOLL circuits or vice versa. The direction  $\text{FOLL} \not\subseteq \text{qAC}^0$  follows from bounds on the average sensitivity of bounded-depth circuits [16]; using these bounds, one can show that there exists a padded version of the PARITY function which can be computed by a FOLL circuit family and cannot be computed by any  $\text{qAC}^0$  circuit family. Conversely, each subset of  $\{0, 1\}^n$  of cardinality at most  $n^{\log n}$  is decidable by a depth-2 circuit of size  $n^{\log n} + 1$ , but for each fixed  $k \in \mathbb{N}$ , there is some large value  $n \geq 1$  such that the number of such subsets exceeds the number of different circuits of size  $n^k$ . This shows that there exist languages in  $\text{qAC}^0$  which are not contained in  $\text{P/poly} \supseteq \text{FOLL}$ .

Designing an FOLL-algorithm which works for arbitrary classes of semigroups with the poly-logarithmic circuits property seems difficult. However, for certain special cases, there is an interesting approach, based on the repeated squaring technique. In the remainder of this section, we sketch one such special case.

For a Cayley circuit, the *width* of a topological ordering  $(v_1, \dots, v_m)$  of the gates is the smallest number  $w \in \mathbb{N}$  such that for each  $i \in \{1, \dots, m-1\}$ , at most  $w$  product gates from the set  $A_i = \{v_1, \dots, v_i\}$  are connected to gates in  $B_i = \{v_{i+1}, \dots, v_m\}$ . Let  $C_i$  be the set of product gates, which belong to  $A_i$  and are connected to gates in  $B_i$ . The subcircuit induced by  $A_i$  can be interpreted as a Cayley circuit computing multiple output values  $C_i$ . The subcircuit induced by  $B_i$  can be seen as a circuit which, in addition to the input gates of the original circuit, uses the gates from  $C_i$  as input gates. The *width* of a Cayley circuit is the smallest width of a topological ordering of its gates. Let us fix some width  $w \in \mathbb{N}$ .

We introduce a predicate  $P(z_1, \dots, z_w, y_1, \dots, y_w, i)$  which is true if there exists a Cayley circuit of width at most  $w$  and size at most  $2^i$  with  $w$  additional input gates and  $w$  additional *passthrough gates* (which have in-degree 1 and replicate the value of their predecessors), such that the elements  $y_1, \dots, y_w \in S$  occur as values of the passthrough gates when using  $z_1, \dots, z_w \in S$  as values for the additional input gates and using any subset of the original inputs  $X$  as values for the remaining input gates. The additional input gates (resp. passthrough gates) are not counted when measuring the circuit size but are considered as product gates when measuring width and they have to be the first (resp. last) gates in all topological orderings considered for width measurement. For each fixed  $i$ , there are only  $n^{2w}$  such predicates.

The truth value of a predicate with  $i = 0$  can be computed by a constant-depth unbounded fan-in Boolean circuit of polynomial size. This is achieved by computing all binary products of the elements  $z_1, \dots, z_w$  and elements of the input set  $X$ . For  $i \geq 1$ , the predicate  $P(z_1, \dots, z_w, y_1, \dots, y_w, i)$  is true if and only if there exist  $z'_1, \dots, z'_w \in S$  such that both  $P(z_1, \dots, z_w, z'_1, \dots, z'_w, i - 1)$  and  $P(z'_1, \dots, z'_w, y_1, \dots, y_w, i - 1)$  are true. Having the truth values of all tuples for  $i - 1$  at hand, this can be checked with a polynomial number of gates in constant depth because there are only  $n^w$  different vectors  $(z'_1, \dots, z'_w) \in S^w$ .

For a class of semigroups with Cayley circuits of bounded width and poly-logarithmic size, we obtain a circuit family of depth  $\mathcal{O}(\log \log n)$  deciding Cayley semigroup membership: the predicates are computed for increasing values of  $i$ , until  $i$  exceeds the logarithm of an upper bound for the Cayley circuit size and then, we return  $P(x, \dots, x, t, \dots, t, i)$  for the element  $t$  given in the input and for an arbitrary element  $x \in X$ . It is worth noting that the circuits constructed in the proof of Proposition 6 have width at most 2, so our FOLL-algorithm is a generalization of the *Double-Barrelled Recursive Strategy* and the proof that  $\text{CSM}(\mathbf{Ab}) \in \text{FOLL}$  presented in [9]. In particular, the procedure above yields a self-contained proof of the following result.

► **Theorem 14.** *Let  $\mathbf{V}$  be a class of semigroups which is closed under taking subsemigroups and has the logarithmic power basis property. Then  $\text{CSM}(\mathbf{V})$  is in FOLL.*

By Lemma 4, we obtain the following corollary.

► **Corollary 15.**  *$\text{CSM}(\mathbf{Com})$  is contained in FOLL.*

## 5 Summary and Outlook

We provided new insights into the complexity of the Cayley semigroup membership problem for classes of finite semigroups, giving parallel algorithms for the variety of finite commutative semigroups and the variety of finite groups. We also showed that a maximal class of semigroups with Cayley semigroup membership decidable by  $\text{qAC}^0$  circuits does not form a variety. Afterwards, we discussed applicability to FOLL.

It is tempting to ask whether one can find nice connections between algebra and the complexity of the Cayley semigroup membership problem by conducting a more fine-grained analysis. For example, it is easy to see that for the varieties of *rectangular bands* and *semilattices*, the Cayley semigroup membership problem is in  $\text{AC}^0$ . Does the maximal class of finite semigroups, for which the Cayley semigroup membership problem is in  $\text{AC}^0$ , form a variety of finite semigroups? Is it possible to show that  $\text{AC}^0$  does not contain  $\text{CSM}(\mathbf{G})$ ? Potential approaches to tackling the latter question are reducing small distance connectivity for paths of non-constant length [17] to  $\text{CSM}(\mathbf{G})$  or developing a suitable switching lemma. Another related question is whether there exist classes of semigroups for which the Cayley semigroup membership problem cannot be NL-hard but, at the same time, is not contained within  $\text{qAC}^0$ .

Moreover, it would be interesting to see whether the Cayley semigroup membership problem can be shown to be in FOLL for all classes of semigroups with the poly-logarithmic circuits property. More generally, investigating the relation between FOLL and  $\text{qAC}^0$ , as well as their relationships to other complexity classes, remains an interesting subject for future research.



## References

- 1 Jorge Almeida. Some pseudovariety joins involving the pseudovariety of finite groups. *Semi-group Forum*, 37(1):53–57, Dec 1988. doi:10.1007/BF02573123.
- 2 László Babai. Trading group theory for randomness. In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 421–429. ACM, 1985. doi:10.1145/22145.22192.
- 3 László Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 164–174. ACM, 1991. doi:10.1145/103418.103440.
- 4 László Babai, Robert Beals, Jin-Yi Cai, Gábor Ivanyos, and Eugene M. Luks. Multiplicative equations over commuting matrices. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '96*, pages 498–507, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics. URL: <http://dl.acm.org/citation.cfm?id=313852.314109>.
- 5 László Babai, Eugene M. Luks, and Ákos Seress. Permutation groups in NC. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 409–420. ACM, 1987. doi:10.1145/28395.28439.
- 6 László Babai, Eugene M. Luks, and Ákos Seress. Permutation groups in NC. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 409–420. ACM, 1987. doi:10.1145/28395.28439.
- 7 László Babai and Endre Szemerédi. On the complexity of matrix group problems I. In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 229–240. IEEE Computer Society, 1984. doi:10.1109/SFCS.1984.715919.
- 8 David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in  $nc^1$ . In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 1–5. ACM, 1986. doi:10.1145/12130.12131.
- 9 David A. Mix Barrington, Peter Keadu, Klaus-Jörn Lange, and Pierre McKenzie. On the complexity of some problems on groups input as multiplication tables. *J. Comput. Syst. Sci.*, 63(2):186–200, 2001. doi:10.1006/jcss.2001.1764.
- 10 David A. Mix Barrington and Pierre McKenzie. Oracle branching programs and logspace versus P. *Inf. Comput.*, 95(1):96–115, 1991. doi:10.1016/0890-5401(91)90017-V.
- 11 David A. Mix Barrington and Denis Thérien. Finite monoids and the fine structure of  $NC^1$ . *J. ACM*, 35:941–952, 1988.
- 12 Martin Beaudry. Membership testing in commutative transformation semigroups. *Inf. Comput.*, 79(1):84–93, 1988. doi:10.1016/0890-5401(88)90018-1.
- 13 Martin Beaudry. *Membership Testing in Transformation Monoids*. PhD thesis, McGill University, Montreal, Quebec, 1988.
- 14 Martin Beaudry. Membership testing in threshold one transformation monoids. *Inf. Comput.*, 113(1):1–25, 1994. doi:10.1006/inco.1994.1062.
- 15 Martin Beaudry, Pierre McKenzie, and Denis Thérien. The membership problem in aperiodic transformation monoids. *J. ACM*, 39(3):599–616, 1992. doi:10.1145/146637.146661.
- 16 Ravi B. Boppana. The average sensitivity of bounded-depth circuits. *Inf. Process. Lett.*, 63(5):257–261, 1997. doi:10.1016/S0020-0190(97)00131-2.
- 17 Xi Chen, Igor Carboni Oliveira, Rocco A. Servedio, and Li-Yang Tan. Near-optimal small-depth lower bounds for small distance connectivity. 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 612–625. ACM, 2016. doi:10.1145/2897518.2897534.
- 18 Merrick L. Furst, John E. Hopcroft, and Eugene M. Luks. Polynomial-time algorithms for permutation groups. In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 36–41. IEEE Computer Society, 1980. doi:10.1109/SFCS.1980.34.
  - 19 Johan Håstad. Almost optimal lower bounds for small depth circuits. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20. ACM, 1986. doi:10.1145/12130.12132.
  - 20 Neil D. Jones and William T. Laaser. Complete problems for deterministic polynomial time. *Theor. Comput. Sci.*, 3(1):105–117, 1976. doi:10.1016/0304-3975(76)90068-2.
  - 21 Neil D. Jones, Y. Edmund Lien, and William T. Laaser. New problems complete for nondeterministic log space. *Mathematical Systems Theory*, 10:1–17, 1976. doi:10.1007/BF01683259.
  - 22 P. Levi. Über die Untergruppen der freien Gruppen. (2. Mitteilung). *Mathematische Zeitschrift*, 37:90–97, 1933. URL: <http://eudml.org/doc/168437>.
  - 23 Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4):17:1–17:24, 2008. doi:10.1145/1391289.1391291.
  - 24 Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10. IEEE Computer Society, 1985. doi:10.1109/SFCS.1985.49.



# Small Normalized Boolean Circuits for Semi-disjoint Bilinear Forms Require Logarithmic Conjunction-depth

Andrzej Lingas<sup>1</sup>

Department of Computer Science, Lund University  
Box 118, 22100 Lund, Sweden  
Andrzej.Lingas@cs.lth.se

---

## Abstract

We consider *normalized* Boolean circuits that use binary operations of disjunction and conjunction, and unary negation, with the restriction that negation can be only applied to input variables. We derive a lower bound trade-off between the size of normalized Boolean circuits computing Boolean semi-disjoint bilinear forms and their conjunction-depth (i.e., the maximum number of and-gates on a directed path to an output gate). In particular, we show that any normalized Boolean circuit of at most  $\epsilon \log n$  conjunction-depth computing the  $n$ -dimensional Boolean vector convolution has  $\Omega(n^{2-4\epsilon})$  and-gates. Analogously, any normalized Boolean circuit of at most  $\epsilon \log n$  conjunction-depth computing the  $n \times n$  Boolean matrix product has  $\Omega(n^{3-4\epsilon})$  and-gates. We complete our lower-bound trade-offs with upper-bound trade-offs of similar form yielded by the known fast algebraic algorithms.

**2012 ACM Subject Classification** Theory of computation → Circuit complexity

**Keywords and phrases** Boolean circuits, semi-disjoint bilinear form, Boolean vector convolution, Boolean matrix product

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.26

**Acknowledgements** The author is grateful to Mike Paterson and the anonymous conference reviewers for valuable comments/suggestions and to Mia Persson for valuable discussions on different versions of this paper.

## 1 Introduction

### 1.1 Background

A set  $F$  of polynomials over a semi-ring is a *form* (in case of the Boolean semi-ring, just a set of monotone Boolean functions).  $F$  is a *semi-disjoint bilinear form* if it defined on the set of variables  $X \cup Y$  and the following properties hold.

1. For each polynomial  $Q$  in  $F$  and each variable  $z \in X \cup Y$ , there is at most one monomial (in the Boolean case, called a prime implicant [24]) of  $Q$  containing  $z$ .
2. Each monomial of a polynomial in  $F$  consists of exactly one variable in  $X$  and one variable in  $Y$ .
3. The sets of monomials of polynomials in  $F$  are pairwise disjoint.

The  $n$ -dimensional vector convolution and the  $n \times n$  matrix product are important and popular examples of semi-disjoint bilinear forms (for the convolution,  $|X| = |Y| = n$  and

---

<sup>1</sup> Research supported in part by VR grant 2017-03750.



$|F| = 2n - 1$  while for the matrix product,  $|X| = |Y| = |F| = n^2$ ). Both semi-disjoint bilinear forms in the arithmetic and Boolean case have a wide range of fundamental applications, for instance, in stringology (see, e.g., [6]) and graph algorithms (see, e.g., [27]).

Two  $n \times n$  integer matrices can be arithmetically multiplied using  $O(n^3)$  additions and multiplications following the definition of matrix product. This is optimal if neither other operations nor negative constants are allowed [13, 16, 20]. If additionally subtraction or negative constants are allowed then the so-called fast matrix multiplication algorithms can be implemented using  $O(n^\omega)$  operations [7, 22, 26], where  $\omega < 3$ . They rely on algebraic equations following from the possibility of term cancellation (for a study on the power of arithmetic term cancellation see [23]). Le Gall and Vassilevska Williams have recently shown the exponent  $\omega$  of fast matrix multiplication to be smaller than 2.373 in [7, 26]. The fast arithmetic algorithms run on 0–1 matrices yield the same asymptotic upper time bounds for  $n \times n$  Boolean matrix multiplication. On the other hand, Raz proved that if only addition, multiplication and products with constants of absolute value not exceeding one are allowed then  $n \times n$  matrix multiplication requires  $\Omega(n^2 \log n)$  operations [17].

Similarly, the arithmetic convolution of two  $n$ -dimensional vectors can be computed using  $O(n^2)$  additions and multiplications. Next, the convolution of two  $n$ -dimensional vectors over a commutative ring with the so-called principal  $n$ -th root of unity can be computed via Fast Fourier Transform using  $O(n \log n)$  operations of the ring. The  $n$ -dimensional Boolean vector convolution admits an algorithm using  $O(n \log^2 n \log \log n)$  Boolean operations by reduction to the fast integer multiplication algorithm from [21] in turn relying on Fast Fourier Transform [6].

It is well known that for uniform problems, their Boolean circuit complexity corresponds up to logarithmic factors to their Turing complexity [24]. Unfortunately, until today no super-linear lower bounds on the size of circuits using binary and unary Boolean operations forming a complete Boolean basis are known for natural problems [24]. On the other hand, such lower bounds are known in case of monotone Boolean circuits that use only the binary operations of disjunction and conjunction [1, 2, 3, 11, 13, 14, 15, 16, 18, 24, 25]. In particular, Alon and Boppana showed by refining Razborov's breakthrough method [18] that the  $(m, s)$ -clique, i.e., the problem of determining if a graph on  $m$  vertices includes a complete subgraph on  $s$  vertices, requires monotone Boolean circuits of  $2^{\sqrt{m}}$  size [1].

There exist interesting connections between the general Boolean circuit complexity and the monotone one [4]. In particular, any Boolean circuit using disjunctions, conjunctions and negations can be easily transformed into a Boolean circuit using the same operations, where negations are applied solely to input variables. The transformations follows from de Morgan's laws and keep the circuit size within a factor 2. In other words, one can see such Boolean circuits as monotone Boolean circuits with respect to the input literals, i.e., input variables and their negations. We shall term Boolean circuits in the latter form *normalized*.

In case of  $n \times n$  Boolean matrix product, almost tight or even tight lower bounds of the form  $\Omega(n^3)$  for the monotone circuit complexity were presented in a series of papers [13, 14, 16] more than three decades ago. The best known (in the literature) lower bound on monotone Boolean circuit complexity for  $n$ -dimensional Boolean vector convolution is  $\Omega(n^2 / \log^6 n)$  due to Grinchuk and Sergeev [8]. It improves on the previously best  $n^{3/2}$  lower bound due to Weiss [25] and an earlier best  $n^{4/3}$  lower bound due to Blum [3]. The lower bounds of Weiss, Grinchuk and Sergeev are on the number of disjunctions while that of Blum is on the number of conjunctions.

Furthermore, Lingas studied the complexity of monotone Boolean circuits for Boolean semi-disjoint bilinear forms under various monotone circuit restrictions in [12]. In particular, he

■ **Table 1** Lower bounds on the monotone Boolean circuit complexity for  $n$ -dimensional Boolean vector convolution in a historical perspective.

author	year	lower bound
N. Pippinger and L.G. Valiant [15]	1976	$\Omega(n \log n)$
E.A. Lamagna [11]	1979	$\Omega(n \log n)$
N. Blum [3]	1980	$n^{4/3}$ conjunctions
R. Weiss [25]	1981	$n^{3/2}$ disjunctions
M.I. Grinchuk and I.S. Sergeev [8]	2011	$\Omega(n^2 / \log^6 n)$ disjunctions

considered monotone Boolean circuits of bounded conjunction-depth, i.e., bounded maximum number of and-gates on any single directed path to an output gate in the monotone circuit. He showed that any monotone Boolean circuit of conjunction-depth at most  $d$  computing a Boolean semi-disjoint form with  $p$  prime implicants has to have at least  $p/2^{2d}$  and-gates. As a corollary, he obtained the  $\Omega(n^{2-2\epsilon})$  lower bound on the size of any monotone Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth computing the  $n$ -dimensional Boolean vector convolution.

## 1.2 Our contributions

Surprisingly enough, we can derive a lower-bound trade-off between the circuit size and its conjunction-depth for normalized Boolean circuits computing semi-disjoint bilinear forms similar to that for monotone Boolean circuits from [12].

More exactly, we show that any normalized Boolean circuit of conjunction-depth at most  $d$  computing a Boolean semi-disjoint form with  $p$  prime implicants has to have  $\Omega(p/2^{4d})$  and-gates. As a corollary, we obtain the  $\Omega(n^{2-4\epsilon})$  lower bound on the size of any normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth computing the  $n$ -dimensional Boolean vector convolution, and an analogous  $\Omega(n^{3-4\epsilon})$  lower bound for the  $n \times n$  Boolean matrix product.

We complete our lower-bound trade-offs with upper-bounds trade-offs of similar form yielded by the aforementioned fast algebraic algorithms. We observe that there is a positive constant  $c \leq 1$  such that for any  $\epsilon \in (0, \frac{1}{c})$ , the  $n$ -dimensional Boolean vector convolution can be computed by a normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{2-c\epsilon} + n \log^2 n \log \log n)$  size. Similarly, there is a positive constant  $c \leq 1$  such that for any  $\epsilon \in (0, \frac{1}{c})$ , the  $n \times n$  Boolean matrix product can be computed by a normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{3-(3-\omega)c\epsilon})$  size.

## 1.3 Motivations

Our primary motivation is the very weak progress in deriving non-trivial lower bounds on the size of Boolean circuits using disjunctions, conjunctions and negations to compute explicit Boolean functions computable in polynomial time, since the 70s (from  $3n$  [19] to almost  $5n$  [9, 10]). For this reason, trade-offs between structural parameters and the size for the aforementioned circuits computing explicit functions should be of interest.

We believe that the conjunction-depth of a normalized Boolean circuit computing a Boolean form whose prime implicants (see Preliminaries) consist of relatively few literals is an interesting structural characteristic. (For not-necessarily normalized Boolean circuit using disjunctions, conjunctions and negations, the concept of conjunction-depth does not make

sense since conjunctions can be eliminated by composing negations with disjunctions via de Morgan's laws. Also, there are trivial examples of Boolean functions that require a large conjunction-depth in normalized circuits. E.g., the function given by  $\neg \bigvee_{i=1}^n x_i \equiv \bigwedge_{i=1}^n \bar{x}_i$  obviously requires  $\log n$  conjunction-depth. The reason is that it has a prime implicant consisting of  $n$  literals.)

Observe that each prime implicant of the functions occurring in semi-disjoint bilinear forms consists solely of two literals. Hence, any semi-disjoint bilinear form admits a normalized (in fact, monotone) Boolean circuit having conjunction-depth 1 and the number of gates proportional to the total number of prime implicants (see also Fact 1).

Our lower-bound trade-offs showing that in order to decrease the size of normalized Boolean circuits computing a semi-disjoint bilinear form one has to increase their conjunction-depth should be of interest. Our upper-bound trade-offs imply that normalized Boolean circuits of even sub-logarithmic conjunction-depth for Boolean vector convolution or Boolean matrix product have substantially smaller size than their monotone counterparts of unbounded conjunction-depth.

## 1.4 Paper structure

In Preliminaries, we introduce basic definitions and notation. In Section 3, we present three lemmata on restricted normalized circuits computing a Boolean form. In Section 4, we show our lower-bound trade-offs for semi-disjoint bilinear forms which constitute our main results. In Section 5, we present our upper-bound trade-offs. We conclude with final remarks.

## 2 Preliminaries

For two Boolean  $n$ -dimensional vectors  $a = (a_0, \dots, a_{n-1})$  and  $b = (b_0, \dots, b_{n-1})$ , their convolution is a vector  $c = (c_0, \dots, c_{2n-2})$ , where  $c_i = \bigvee_{l=\max\{i-n+1, 0\}}^{\min\{i, n-1\}} a_l \wedge b_{i-l}$  for  $i = 0, \dots, 2n-2$ .

A *literal* is a variable or the negation of a variable.

A (*Boolean*) *circuit* is a finite directed acyclic graph with the following properties:

1. The indegree of each vertex (termed gate) is either 0, 1 or 2.
2. The source vertices (i.e., vertices with indegree 0 called input gates) are labeled by elements in some set of literals, i.e., variables and their negations, and the Boolean constants 0, 1.
3. The vertices of indegree 2 are labeled by elements of the set  $\{and, or\}$  and termed and-gates and or-gates, respectively.
4. The vertices of indegree 1 are labeled by *negation* and termed negation-gates.

A Boolean circuit is *normalized* if it does not use negation-gates. A Boolean circuit is *monotone* if it is normalized and it does not use negated variables.

The *size* of a Boolean circuit  $C$  is the total number of non-input gates in  $C$  while the *depth* of  $C$  is the maximum length of a directed path in  $C$ . Furthermore,  $C$  is of *conjunction-depth*  $d$  if the number of and-gates on any directed path in  $C$  does not exceed  $d$ .

With each gate  $g$  of a normalized Boolean circuit, we associate a set  $T(g)$  of terms in a natural way. Thus, with each input gate, we associate the singleton set consisting of the corresponding variable, negated variable or constant. Next, with an or-gate, we associate the union of the sets associated with its direct predecessors. Finally, with an and-gate  $g$ , we associate the set of concatenations  $t_1 t_2$  of all pairs of terms  $t_1, t_2$ , where  $t_i \in T(g_i)$  and  $g_i$  stands for the  $i$ -th direct predecessor of  $g$  for  $i = 1, 2$ . The function computed at

the gate  $g$  is the disjunction of the functions (called monoms) represented by the terms in  $T(g)$ . The monom represented by a term  $t$  is obtained by replacing concatenations in  $t$  with conjunctions, respectively. A term in  $T(g)$  is a *zero-term* if it contains the Boolean constant 0 or a variable and its negation. Clearly, a zero-term represents the Boolean constant 0.

A form composed of  $k$  Boolean functions is computed by a Boolean circuit if there are  $k$  distinguished gates (called output gates) computing the  $k$  functions.

A term (an output term, respectively) of a circuit  $C$  is a term in  $T(g)$  for some gate (output gate, respectively)  $g$  of  $C$ .

An *implicant* of a Boolean form  $F$  is a conjunction of some variables and/or some negated variables of  $F$  and/or Boolean constants (monom) such that there is a function belonging to  $F$  which is true whenever the conjunction is true. If the conjunction includes the Boolean 0 or a variable  $x$  and its negation  $\bar{x}$  then it is a *trivial implicant* of (any)  $F$ .

A non-trivial implicant of  $F$  that is minimal with respect to included literals is a *prime implicant* of  $F$ .

The following upper bound is straight-forward.

► **Fact 1.** [12] *Each Boolean semi-disjoint bilinear form composed of  $l$  functions on  $x_0, \dots, x_{n-1}$  and  $y_0, \dots, y_{n-1}$  with  $p$  prime implicants in total can be computed by a monotone Boolean circuit of conjunction-depth 1 with  $p \leq n^2$  and-gates and  $p - l$  or-gates.*

**Proof.** First, we use  $p$  and-gates to compute each prime implicant  $x_i y_j$  separately. Then, we form  $l$  disjoint or-unions of the prime implicants corresponding to the  $l$  functions of the bilinear form using  $p - l$  or-gates. ◀

### 3 Lemmata on Normalized Circuits

Recall that the monom represented by a term  $t$  is obtained by replacing concatenations in  $t$  with conjunctions, respectively. We shall say that an implicant (in particular, a prime implicant) of a function  $f_g$  computed at the gate  $g$  is represented by a single term in  $T(g)$  if there is a term  $t \in T(g)$  such that the monom represented by  $t$  is equivalent to the implicant.

In the following two lemmata, we shall show that if the output terms of a normalized circuit computing a form contain a bounded number of different literals, we can obtain a situation somewhat similar to that in monotone circuits, where each prime implicant of an output function has to be represented by a single output term. Namely, we can zero some part of variables such that in the resulting circuit, a large part of the prime implicants of the form is represented by single output terms.

► **Lemma 2.** *Let  $C$  be a normalized Boolean circuit computing a form  $F$ . For each prime implicant of the function  $f_o \in F$  computed at the output gate  $o$  of  $C$ , there is a term in  $T(o)$  representing the (whole) prime implicant or a conjunction of the prime implicant with solely negated variables.*

**Proof.** Consider a prime implicant of  $f_o$ . Assign the Boolean 1 to the variables in the prime implicant and the Boolean 0 to all remaining variables in  $F$ . Under this assignment, the value of  $f_o$  should be 1. Hence, since each term in  $T(o)$  has to represent an implicant of  $f_o$ , there must exist a term in  $T(o)$  representing the whole prime implicant or a conjunction of the prime implicant with solely negated variables. ◀

► **Lemma 3.** *Let  $C$  be a normalized Boolean circuit computing a form  $F$  with  $p$  prime implicants. Suppose that each prime implicant of  $F$  is composed of  $q$  (not negated) variables and each output term of  $C$  contains at most  $k$  distinct literals. Let  $0 < \beta < 1$ . There is a subset of the set of variables of  $F$  such that after setting them to the Boolean 0 there are at least  $p\beta^q(1 - \beta)^{k-q}$  prime implicants of  $F$  represented by single output terms of the circuit  $C'$  resulting from  $C$ . Note that the circuit  $C'$  computes a form  $F'$  whose set of prime implicants is a subset of that of  $F$ .*

**Proof.** Set each variable of  $F$  to the Boolean constant 0 with probability  $1 - \beta$  uniformly at random. Consider any prime implicant  $x_{i_1} \dots x_{i_q}$  of  $F$ . The probability that none of  $x_{i_1}, \dots, x_{i_q}$  is set to 0 is  $\beta^q$ . By Lemma 2, there is a set of  $0 \leq l \leq k - q$  negated variables whose conjunction with  $x_{i_1} \dots x_{i_q}$  is represented by an output term of  $C$ . The probability that each of these negated  $l$  variables is set to 0 is at least  $(1 - \beta)^{k-q}$ . Hence, the expected number of prime implicants of the form computed by the resulting circuit represented by single output terms in this circuit is at least  $p\beta^q(1 - \beta)^{k-q}$ . It follows that there is a subset of the set of variables satisfying the requirements of the lemma. ◀

The final lemma in this section is pretty obvious.

► **Lemma 4.** *Let  $C$  be a normalized Boolean circuit of  $d$ -bounded conjunction-depth computing a form  $F$ . Each term, in particular, each output term of  $C$  includes at most  $2^d$  literals.*

**Proof.** An and-gate can at most double the number of literals in single terms while an or-gate does not increase it. Hence, by induction on the maximum number  $d$  of and-gates on a path from an input gate to a gate  $g$  in  $C$ , any term in  $T(g)$  includes at most  $2^d$  literals. ◀

## 4 Lower-bound Trade-offs (main results)

In monotone circuits, where negation is not used, each prime implicant of a function computed at a gate  $h$  has to be represented by a single term in  $T(h)$  (there might be several such terms and many other terms having subterms representing the prime implicant). This is not the case in normalized circuits generally. There, we can associate to a prime implicant of the function the set of all terms in  $T(g)$  representing a conjunction of the prime implicant with an additional conjunction of literals (e.g.,  $x_i y_j$  could be represented by  $\{x_i y_j x_k, x_i y_j \bar{x}_k\}$ ). Interestingly, the disjunction of the aforementioned additional conjunctions does not have to be always true (e.g.,  $x \vee y$  could be computed by  $x\bar{y} \vee y$  so the prime implicant  $x$  would be represented just by  $\{x\bar{y}\}$ ).

First, we shall show how a restriction on the maximum number of distinct literals which occur in an output term of a normalized Boolean circuit computing a Boolean semi-disjoint form can be used to derive a non-trivial lower bound on the number of and-gates in the circuit.

► **Lemma 5.** *Let  $C$  be a normalized Boolean circuit computing a semi-disjoint bilinear form  $F$  on the variables  $x_0, \dots, x_{n-1}$  and  $y_0, \dots, y_{n-1}$ . Suppose that for each output gate  $o$  in  $C$ , each term in  $T(o)$  contains at most  $k$  different literals. Let  $h$  be a gate connected by directed paths with some output gates in  $C$  such that the function computed at  $h$  has prime implicants  $z_{q_1}, \dots, z_{q_{l(h)}}$  which are single (not negated) variables represented by single terms in  $T(h)$ , and possibly some other prime implicants. The inequality  $l(h) \leq k$  holds or  $h$  can be replaced by the Boolean constant 1.*

**Proof.** Consider a directed path  $P$  connecting  $h$  with some output gate  $o$  in  $C$ . At the output gate  $o$ , for each  $z_{q_r}$ ,  $1 \leq r \leq l(h)$ , any single term  $t(z_{q_r}) \in T(h)$  representing  $z_{q_r}$  has to appear in terms  $t_1 t(z_{q_r}) t_2$  in the associated set  $T(o)$  (see Preliminaries) such that  $t_1 t_2$  is a concatenation (i.e., conjunction) of some terms added by subsequent and-gates on  $P$  and  $t_1 t(z_{q_r}) t_2$  represents an implicant of the function  $f_o$  computed at  $o$ . In general,  $t(z_{q_r})$  may include several occurrences of  $z_{q_r}$  and the Boolean 1, for simplicity we may assume w.l.o.g. that  $t(z_{q_r}) = z_{q_r}$ . (The reason of having  $t_1, t_2$  instead of a single term  $t$  is that syntactically the concatenations can come from both sides.)

Suppose that there is such a  $t_1 t_2$ , where  $t_1 z t_2 \in T(o)$  for some  $z \in \{z_{q_r} | 1 \leq r \leq l(h)\}$ , which does not represent an implicant of  $f_o$ . It follows from the definition of  $t_1 t_2$  that for any  $z \in \{z_{q_r} | 1 \leq r \leq l(h)\}$ , the term  $t_1 z t_2$  also appears in the set  $T(o)$  of terms associated with the output gate  $o$  and consequently it has to represent an implicant of  $f_o$  as well. Therefore, for each such a  $z$ , either  $t_1 t_2$  contains  $\bar{z}$  or  $t_1 t_2$  contains the unique "mate" variable  $z'$  for which  $z z'$  is a prime implicant of  $f_o$ . Note that if  $z$  is an  $x$ -variable then  $z'$  is a  $y$ -variable and *vice versa*. Set  $H$  to  $\{z_{q_1}, \dots, z_{q_{l(h)}}\}$ . E.g., the case that  $t_1 t_2$  contains  $\bar{z}$  could happen if there were some other variables  $z'' \in H$  for which  $t_1 z'' t_2$  are not trivial implicants of  $f_o$  but  $t_1 z t_2$  becomes a trivial implicant because it contains both  $z$  and  $\bar{z}$ .

Consider the mapping of each  $z \in H$  either to the  $z'$  in  $t_1 t_2$  (which must be the unique "mate" among the prime implicants of  $f_o$ ) or to the  $\bar{z} \in t_1 t_2$ . Clearly, all the  $\bar{z}$  for  $z \in H$  are distinct negated variables. Because no two elements of  $H$  have the same mate among the prime implicants of  $f_o$ , no two of the  $z'$  for  $z \in H$  can be the same. Finally, the mates  $z'$  are single not negated variables. It follows that the mapping is one-to-one. We infer that  $l(h) \leq k$ .

On the contrary, if each such term  $t = t_1 t_2$  for each path  $P$  connecting  $h$  with any output gate  $o$ , represents an implicant of  $f_o$  then on each  $P$  we could connect the successor of the start vertex  $h$  with the Boolean constant 1 instead of  $h$  and the output gate  $o$  still would output  $f_o$ . To see this observe that then each  $u \in T(h)$  is a part of the terms of the form  $t_1 u t_2$  in  $T(o)$ , where  $t_1 t_2$  represents an implicant of the function  $f_o$ . Since this holds for each successor of  $h$ , this gate can be replaced by the constant 1. ◀

For an and-gate  $g$  in a normalized Boolean circuit  $C$  computing a semi-disjoint bilinear form  $F$ ,  $S_g$  will denote the set of prime implicants  $s$  of  $F$  such that:

1.  $s$  is a prime implicant of the function computed at  $g$  that is represented by a single term in  $T(g)$ ,
2.  $s$  is not a prime implicant of the function computed at either of the two direct predecessors  $h$  of  $g$  that is represented by a single term in  $T(h)$ , and
3. there is a directed path connecting  $g$  with the output gate computing the function whose prime implicant is  $s$ .

► **Lemma 6.** *Let  $C$  be a normalized Boolean circuit computing a semi-disjoint bilinear form  $F$ . Suppose that for each output gate  $o$  in  $C$ , each term in  $T(o)$  contains at most  $k$  different literals. Next, suppose that  $C$  does not contain any and-gate that could be replaced by the Boolean 1 so the resulting circuit would still compute  $F$ . For any and-gate  $g$  in  $C$ , the inequality  $|S_g| \leq k^2$  holds.*

**Proof.** We may assume w.l.o.g.  $|S_g| \geq 1$ . It follows that at least for one of the direct predecessor gates  $h$  of  $g$ , the function computed at  $h$  has at least  $\sqrt{|S_g|}$  single variable prime implicants represented by single terms in  $T(h)$ . By Lemma 5, we infer that either  $\sqrt{|S_g|} \leq k$  or the gate  $h$  can be replaced by the constant 1. The latter possibility contradicts the lemma assumptions. ◀



► **Theorem 7.** *Let  $C$  be a normalized Boolean circuit computing a semi-disjoint bilinear form  $F$  with  $p$  prime implicants. Suppose that each output term of  $C$  contains at most  $k$  distinct literals. The circuit  $C$  has at least  $\frac{p}{k^4}(1 - \frac{1}{k})^{k-2}$  and-gates.*

**Proof.** We shall apply Lemma 3 with  $\beta = \frac{1}{k}$  and  $q = 2$  to the circuit  $C$ . Let  $C'$  be the circuit resulting from  $C$  by zeroing the subset of variables specified in this lemma. Note that the output terms of  $C'$  still contain at most  $k$  different literals, and that  $C'$  computes a semi-disjoint bilinear form  $F'$  whose prime implicants are prime implicants of  $F$ . Among the prime implicants of  $F'$ , at least  $\frac{p}{k^2}(1 - \frac{1}{k})^{k-2}$  are represented by single output terms by Lemma 3.

Iterate the following steps starting from the circuit  $C'$ . Whenever the current circuit contains an and-gate or an or-gate  $h$  that can be replaced by the Boolean constant 1 without affecting the functions computed at the output gates, replace  $h$  by 1. By induction on the number of iterations, the new circuit still computes the same bilinear form  $F'$ . Also, the number of prime implicants of  $F'$  represented by single output terms does not drop and each output term of the new circuit contains at most  $k$  literals.

Since the circuit  $C'$  is finite and each iteration eliminates at least one gate, after a finite number of iterations, we obtain a circuit  $C''$  sharing the aforementioned properties, not containing any and-gate or or-gate that could be replaced by 1, and still computing  $F'$ . It follows from Lemma 5 that  $C''$  does not have any gate  $h$  such that the function computed at  $h$  contains more than  $k$  single-variable prime implicants represented by single terms in  $T(h)$ .

Let  $S$  be the set of at least  $\frac{p}{k^2}(1 - \frac{1}{k})^{k-2}$  prime implicants of  $F'$  represented by single output terms of  $C''$ . Recall the definition of the set  $S_g$  of prime implicants of a form for an and-gate  $g$  given before Lemma 6. For each  $s \in S$ , there must be at least one and-gate  $g$  of  $C''$  such that  $s \in S_g$ . (To find such a gate  $g$  start from the output gate computing the function of  $F'$  for which  $s$  is a prime implicant represented by a single term and iterate the following steps: check if the current gate  $g$  satisfies  $s \in S_g$ , if not go to the direct predecessor of  $g$  that computes a function having  $s$  as a prime implicant represented by a single term.) By the latter lemma, we have  $|S_g| \leq k^2$ . Hence,  $C''$  has at least  $|S|/k^2 \geq \frac{p}{k^2}(1 - \frac{1}{k})^{k-2}/k^2 \geq \frac{p}{k^4}(1 - \frac{1}{k})^{k-2}$  and-gates since  $|S| \geq \frac{p}{k^2}(1 - \frac{1}{k})^{k-2}$ . ◀

By combining Theorem 7 with Lemma 4, we obtain our main result.

► **Theorem 8.** *Let  $C$  be a normalized Boolean circuit of conjunction-depth at most  $d$  computing a semi-disjoint bilinear form  $F$  with  $p$  prime implicants. The circuit  $C$  has at least  $\frac{p}{2^{4d}}(1 - \frac{1}{2^d})^{2^d-2}$  and-gates.*

Observe that the  $n$ -dimensional Boolean vector convolution has  $\Theta(n^2)$  prime implicants while the  $n \times n$  Boolean matrix product has  $\Theta(n^3)$  prime implicants.

► **Corollary 9.** *For  $\epsilon > 0$ , any normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth that computes the  $n$ -dimensional Boolean vector convolution has  $\Omega(n^{2-4\epsilon})$  and-gates.*

► **Corollary 10.** *For  $\epsilon > 0$ , any normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth that computes the  $n \times n$  Boolean matrix product has  $\Omega(n^{3-4\epsilon})$  and-gates.*



## 5 Upper-bound Trade-offs

The fast algebraic algorithms for arithmetic matrix multiplication [7, 22, 26] yield normalized Boolean circuits for the  $n \times n$  Boolean matrix product of  $O(n^\omega)$  size and  $O(\log n)$  depth (see [5]). Similarly, the fast algorithm for integer multiplication [21] yields normalized Boolean circuits for the  $n$ -dimensional Boolean vector convolution of  $O(n \log^2 n \log \log n)$  size and  $O(\log n)$  depth [6, 5]. We can use these facts to derive the following upper-bound trade-offs analogous to our lower-bound trade-offs for these two problems.

► **Proposition 11.** *There is a positive constant  $c \leq 1$  such that for any  $\epsilon \in (0, \frac{1}{c})$ , the  $n$ -dimensional Boolean vector convolution can be computed by a normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{2-c\epsilon} n \log^2 n \log \log n)$  size.*

**Proof.** By the aforementioned facts, for some positive constant  $c \leq 1$ , an  $n^{c\epsilon}$ -dimensional Boolean vector convolution can be computed by a normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{c\epsilon} \log^2 n \log \log n)$  size. On the other hand, since  $c\epsilon < 1$ , the  $n$ -dimensional Boolean vector convolution can be easily reduced to  $n^{2-2c\epsilon}$   $n^{c\epsilon}$ -dimensional Boolean vector convolutions using just disjunctions. The resulting normalized Boolean circuit has still  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{2-c\epsilon} \log^2 n \log \log n)$  size. ◀

► **Proposition 12.** *There is a positive constant  $c \leq 1$  such that for any  $\epsilon \in (0, \frac{1}{c})$ , the  $n \times n$  Boolean matrix product can be computed by a normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{3-(3-\omega)c\epsilon})$  size.*

**Proof.** By the aforementioned facts, there is a positive constant  $c \leq 1$  such that an  $n^{c\epsilon} \times n^{c\epsilon}$  Boolean matrix product can be computed by a normalized Boolean circuit of  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{\omega c\epsilon})$  size. On the other hand, since  $c\epsilon < 1$ , the  $n \times n$  Boolean matrix product can be easily reduced to  $n^{3-3c\epsilon}$   $n^{c\epsilon} \times n^{c\epsilon}$  Boolean matrix products using just disjunctions. The resulting normalized Boolean circuit has still  $\epsilon \log n$ -bounded conjunction-depth and  $O(n^{3-(3-\omega)c\epsilon})$  size. ◀

## 6 Final Remarks

The disjointness of the sets of prime implicants of the Boolean functions forming a bilinear form is not essential in the proofs of Theorems 7, 8. Hence, these theorems hold even for Boolean bilinear forms satisfying only the two remaining conditions (see Introduction) provided that  $p$  denotes the number of distinct prime implicants of the form.

Our main results are the lower-bound trade-offs between the number of and-gates and conjunction-depth in normalized Boolean circuits computing semi-disjoint bilinear forms (Section 4). They rely on the analysis of output terms containing bounded numbers of literals because of the assumed bound on the conjunction-depth (Lemma 4, note that this lemma wouldn't hold if the fan-in of and-gates wasn't bounded).

---

### References

- 1 N. Alon and R. B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.
- 2 A. E. Andreev. On one method of obtaining constructive lower bounds for the monotone circuit size. *Algebra and Logics*, 26(1):3–26, 1987.
- 3 N. Blum. An  $\omega(n^{4/3})$  lower bound on the monotone network complexity of the  $n$ -th degree convolution. *Theoretical Computer Science*, 36:59–69, 1985.
- 4 N. Blum. On negations in boolean networks. In *Efficient Algorithms*, volume 5760 of *Lecture Notes in Computer Science*, pages 18–29. Springer-Verlag, 2009.

- 5 J. H. Reif (editor). *Synthesis of Parallel Algorithms*. Morgan Kaufmann Publishers, San Mateo, 1993.
- 6 M. J. Fisher and M. S. Paterson. String-matching and other products. In *Proceedings of the 7th SIAM-AMS Complexity of Computation*, pages 113–125, 1974.
- 7 F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, Lecture Notes in Computer Science, pages 296–303. Springer-Verlag, 2014.
- 8 M. I. Grinchuk and I. S. Sergeev. Thin circulant matrices and lower bounds on the complexity of some boolean operations. *Diskretn. Anal. Issled. Oper.*, 18:35–53, 2011.
- 9 K. Iwama and H. Morizumi. An explicit lower bound of  $5n - o(n)$  for boolean circuits. In *Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, pages 353–364. Springer-Verlag, 2002.
- 10 O. Lachish and R. Raz. Explicit lower bound of  $4.5n - o(n)$  for boolean circuits. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 399–408. ACM, 2001.
- 11 E. A. Lamagna. The complexity of monotone networks for certain bilinear forms, routing problems, sorting, and merging. *IEEE Transactions on Computers*, c-28(10), 1979.
- 12 A. Lingas. Towards an almost quadratic lower bound on the monotone circuit complexity of the boolean convolution. In *Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, pages 401–411. Springer-Verlag, 2017.
- 13 K. Mehlhorn and Z. Galil. Monotone switching circuits and boolean matrix product. *Computing*, 16:99–111, 1976.
- 14 M. Paterson. Complexity of monotone networks for boolean matrix product. *Theoretical Computer Science*, 1(1):13–20, 1975.
- 15 N. Pippenger and L.G. Valiant. Shifting graphs and their applications. *Journal of the ACM*, 23(3):423–432, 1976.
- 16 R. Pratt. The power of negative thinking in multiplying boolean matrices. *SIAM J. Comput.*, 4(3):326–330, 1975.
- 17 R. Raz. On the complexity of matrix product. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 144–151. ACM, 2002.
- 18 A. A. Razborov. Lower bounds on the monotone complexity of some boolean functions. *Doklady Akademii Nauk*, 281(4):798–801, 1985.
- 19 C. P. Schnorr. Zwei lineare untere schranken für die komplexität boolescher funktionen. *Computing*, 13(2):155–171, 1974.
- 20 C. P. Schnorr. A lower bound on the number of additions in monotone computations. *Theoretical Computer Science*, 2(3):305–315, 1976.
- 21 A. Schönhage and V. Strassen. Schnelle multiplikation grober zahlen. *Computing*, 7:281–292, 1971.
- 22 V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.
- 23 L.G. Valiant. Negation can be exponentially powerfull. *Theoretical Computer Science*, 12:303–314, 1980.
- 24 I. Wegener. *The Complexity of Boolean Functions*. Wiley-Teubner Series in Computer Science, New York, Stuttgart, 1987.
- 25 J. Weiss. An  $n^{3/2}$  lower bound on the monotone network complexity of the boolean convolution. *Information and Control*, 59:184–188, 1983.
- 26 V. Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 807–898. ACM, 2012.
- 27 U. Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication. *Journal of the ACM*, 49(3):289–317, 2002.

# Lower Bounds on Non-Adaptive Data Structures Maintaining Sets of Numbers, from Sunflowers

Sivaramakrishnan Natarajan Ramamoorthy<sup>1</sup>

Paul G. Allen School for Computer Science & Engineering, University of Washington, Seattle, USA

sivanr@cs.washington.edu

Anup Rao<sup>2</sup>

Paul G. Allen School for Computer Science & Engineering, University of Washington, Seattle, USA

anuprao@cs.washington.edu

---

## Abstract

---

We prove new cell-probe lower bounds for dynamic data structures that maintain a subset of  $\{1, 2, \dots, n\}$ , and compute various statistics of the set. The data structure is said to handle insertions *non-adaptively* if the locations of memory accessed depend only on the element being inserted, and not on the contents of the memory. For any such data structure that can compute the median of the set, we prove that:

$$t_{\text{med}} \geq \Omega\left(\frac{n^{\frac{1}{t_{\text{ins}}+1}}}{w^2 \cdot t_{\text{ins}}^2}\right),$$

where  $t_{\text{ins}}$  is the number of memory locations accessed during insertions,  $t_{\text{med}}$  is the number of memory locations accessed to compute the median, and  $w$  is the number of bits stored in each memory location. When the data structure is able to perform deletions non-adaptively and compute the minimum non-adaptively, we prove

$$t_{\text{min}} + t_{\text{del}} \geq \Omega\left(\frac{\log n}{\log w + \log \log n}\right),$$

where  $t_{\text{min}}$  is the number of locations accessed to compute the minimum, and  $t_{\text{del}}$  is the number of locations accessed to perform deletions. For the predecessor search problem, where the data structure is required to compute the predecessor of any element in the set, we prove that if computing the predecessors can be done non-adaptively, then

$$\text{either } t_{\text{pred}} \geq \Omega\left(\frac{\log n}{\log \log n + \log w}\right), \text{ or } t_{\text{ins}} \geq \Omega\left(n^{\frac{1}{2(t_{\text{pred}}+1)}}\right),$$

where  $t_{\text{pred}}$  is the number of locations accessed to compute predecessors.

These bounds are nearly matched by Binary Search Trees in some range of parameters. Our results follow from using the Sunflower Lemma of Erdős and Rado [11] together with several kinds of encoding arguments.

**2012 ACM Subject Classification** Theory of computation → Cell probe models and lower bounds

**Keywords and phrases** Non-adaptive data structures, Sunflower lemma

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.27

---

<sup>1</sup> Supported by the National Science Foundation under agreement CCF-1420268 and CCF-1524251

<sup>2</sup> Supported by the National Science Foundation under agreement CCF-1420268 and CCF-1524251



© Sivaramakrishnan Natarajan Ramamoorthy and Anup Rao;

licensed under Creative Commons License CC-BY

33rd Computational Complexity Conference (CCC 2018).

Editor: Rocco A. Servedio; Article No. 27; pp. 27:1–27:16



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



**Acknowledgements** We thank Paul Beame for useful discussions and bringing to our attention a variant of the Sunflower lemma from [2]. We thank Pavel Hrubeš for observing that one could insert a set using deletions, to prove Theorem 1. We thank Kasper Green Larsen and Mikkel Thorup for helpful conversations. We thank an anonymous reviewer for a suggestion that helped improve a bound in Theorem 3.

## 1 Introduction

Data structures are algorithmic primitives to efficiently manage data. They are used widely in computer systems, and not just to maintain large data sets; these primitives play a fundamental role in many algorithmic tasks. For example, the *heap* data structure is a crucial component of the best algorithms for computing shortest paths in weighted graphs, and the *union-find* data structure is vital to algorithms for computing minimum spanning trees in graphs. In both of these examples, the running times of these algorithms depend on the performance of the underlying data structures. In this paper, we study data structures that maintain a set of numbers  $S$  and allow for quickly computing the *minimum*, *median* or *predecessors* of the set. The median is the middle number of the set in sorted order, and the predecessor of a number  $x$  is the largest element in  $S$  that is at most  $x$ . We give new lower bounds on data structures computing these statistics.

The performance of data structures is usually measured with Yao’s *cell-probe* model [32]. A *dynamic data structure* in this model is a collection of *cells* that stores the data, along with an algorithm that makes changes to the data or retrieves information about it by reading from and writing to some of the cells. The *word-size* of the data structure, denoted  $w$  throughout this paper, is the number of bits stored in each cell of the data structure. The time complexity for performing a particular operation is the number of cells that are accessed when the operation is carried out. Usually, there is a trade-off between the time for performing different operations. For example, if we maintain a set  $S \subseteq \{1, 2, \dots, n\}$  by storing its indicator vector (with  $w = 1$ ), then elements can be inserted and deleted from the set in time 1, but computing the median of the set could take time  $\Omega(n)$  in the worst case. However, if we maintained the set by storing its elements in sorted order (with  $w = \log n$ ), and the size of the set, then the median can be computed in time 2, but inserting elements into the set would take time  $\Omega(n)$ . Binary search trees are a well-known data structure that maintain sets and allow one to compute the median and predecessors in time  $O(\log n)$ , when  $w = \log n$ . One can also use a very clever data structure due to van Emde Boas [29] that brings down the time required for all operations to  $O(\log \log n)$ , when  $w = \log n$ . The Fusion trees data structure of Fredman and Willard [14] takes  $O(\log n / \log w)$  time for all operations.

Proving lower bounds on the performance of dynamic data structures is usually challenging. In their landmark paper, Fredman and Saks [13] were the first to establish tight lower bounds for several dynamic data structure problems. They invented the *chronogram technique* and leveraged it to prove several lower bounds. Since then, researchers have built on their techniques to prove lower bounds on many other dynamic data structure problems [24, 23, 26, 18, 33, 30]. Notably, Pătraşcu and Thorup [26] proved lower bounds on data structures that can compute the  $k$ ’th smallest number of the set for every  $k$  via a reduction from Parity Sum for which [13] used the chronogram technique to prove a lower bound. This shows that computing the  $k$ ’th smallest element takes strictly more time than just computing the median. Some of our own results also use the chronogram technique of Fredman-Saks.

Lower bounds on data structures for computing single statistics like the median or minimum have been particularly elusive. Computing statistics like the median and the minimum are very fundamental in algorithm design. The best known upper bounds require  $O(\log \log n)$  time for insertions, and median and minimum computations. It is surprising that no previous lower bounds were known in the cell-probe model. We prove the first lower bounds on the performance of data structures computing the median and minimum. Brodal, Chaudhuri and Radhakrishnan [7] showed that if the data structure is only allowed to compare the contents of cells, and perform no other computation with the cells, then we must have  $t_{\min} \geq \Omega(n/4^{t_{\text{ins}}})$ , where  $t_{\min}$  is the number of comparisons used to compute the minimum, and  $t_{\text{ins}}$  is the number of comparisons used to insert numbers into the set. Moreover, [7] gave a data structure matching these bounds. The same bounds apply for computing the median as well. It remains an interesting open problem to prove a lower bound of  $t_{\text{ins}} + t_{\text{med}} \geq \Omega(\log \log n)$  in the cell-probe model when  $w = O(\log n)$ , where  $t_{\text{ins}}$  is the time for insertions and  $t_{\text{med}}$  is the time to compute the median. We note here that there is a long sequence of works proving lower bounds on computing the median in the context of branching programs [10, 21, 5, 9].

Past work had found more success with understanding the complexity of the predecessor search problem. A long sequence of works has proved lower bounds here [1, 20, 19, 4, 28, 25]. In particular, [4, 28] showed that some operation must take time  $\Omega(\log \log n / \log \log \log n)$ , when  $w = \log n$ , and this was improved to  $\Omega(\log \log n)$  by [25]. Still, it remains open to understand the full trade-off between the time complexity of inserting elements and the time complexity of computing predecessors<sup>3</sup>.

In our work, we prove new lower bounds on *non-adaptive* data structures that allow for computing the median, minimum, and predecessors of elements. A data structure is said to perform an operation non-adaptively if the locations of memory accessed depend only on the operation being performed, and not on the contents of the memory that are read while the operation is executing.

Perhaps the most widely known and basic dynamic data structure for maintaining sets of numbers is the binary search tree (see Appendix A for a description). Both insertions and deletions into a binary search tree are non-adaptive operations. Indeed, all of the assumptions regarding non-adaptivity made in our lower bounds are satisfied by binary search trees—so the models we consider here are both well motivated and quite natural. Non-adaptive data structures tend to be simple, and faster in practice. This is because a practical implementation can load all of the cells required to perform the operation into a local cache in a single step, rather than having to fetch cells from the memory multiple times.

Several past works have proved lower bounds on various computational models under the assumption of non-adaptivity (see for example [17]). In the context of data structures, Brody and Larsen [8] showed polynomial lower bounds for various dynamic problems in the non-adaptive setting. Among other results, they showed that any data structure for reachability in directed graphs that non-adaptively checks for reachability between pairs of vertices must take time  $\Omega(n/w)$ , where  $n$  is the size of the underlying graph. [3, 16] proved non-adaptive lower bounds on static data structures for the dictionary problem in the bit probe model.

---

<sup>3</sup> We thank Mikkel Thorup for bringing this question to our attention.

## 1.1 Our Results

We prove new lower bounds on non-adaptive data structures computing the minimum, median and predecessors. Our results are obtained via an application of the famous Sunflower Lemma of Erdős and Rado [11]. The Sunflower Lemma was used in the past to prove lower bounds on dynamic data structures by Frandsen and Miltersen [12] and then again for static data structures by Gal and Miltersen [15], and our use of it is similar. However, in the setting of non-adaptive data structures, we are able to leverage the lemma to get results even when the word size is large.

Our first result proves a lower bound when both deletions and minimum computations are non-adaptive<sup>4</sup>. Similar results hold for computing the median and predecessors as well, but they are subsumed by the theorems to follow.

► **Theorem 1.** *Any data structure that computes the minimum of a subset of  $\{1, 2, \dots, n\}$  while supporting non-adaptive delete operations and non-adaptive minimum computations must take time  $\Omega\left(\frac{\log n}{\log \log n + \log w}\right)$  for some operation, where  $w$  is the word size of the cells.*

Our second result concerns non-adaptive data structures for computing the median. Here the lower bound holds even if the median computation is adaptive and the insertion operation is non-adaptive:

► **Theorem 2.** *Any data structure that computes the median of a subset of  $\{1, 2, \dots, n\}$  while supporting non-adaptive insert operations must satisfy*

$$t_{\text{med}} \geq \Omega\left(\frac{n^{\frac{1}{t_{\text{ins}}+1}}}{w^2 \cdot t_{\text{ins}}^2}\right),$$

where  $t_{\text{med}}$  is the time required to compute the median,  $t_{\text{ins}}$  is the time required to insert elements, and  $w$  is the word size of the cells.

Our last result concerns the predecessor search problem. Here the lower bound holds even if the insertion operation is adaptive, as long as the predecessor computations are non-adaptive:

► **Theorem 3.** *Any data structure that maintains a subset of  $\{1, 2, \dots, n\}$  while supporting non-adaptive predecessor operations must satisfy*

$$t_{\text{pred}} \geq \Omega\left(\frac{\log n}{\log \log n + \log w}\right) \quad \text{or} \quad t_{\text{ins}} \geq \Omega\left(n^{\frac{1}{2(t_{\text{pred}}+1)}}\right),$$

where  $t_{\text{ins}}$  is the time required for inserts,  $t_{\text{pred}}$  is the time required for computing predecessors and  $w$  is the word-size of the cells.

Very recently, Boninger, Brody and Kephart [6] independently obtained some lower bounds on non-adaptive data structures computing predecessors. Among other results, they showed that any data structure with non-adaptive insertions and non-adaptive predecessor computations must have<sup>5</sup>  $t_{\text{ins}} \geq \Omega(\log n)$ , or  $t_{\text{pred}} \geq \frac{\log n}{\log w + \log t_{\text{ins}}}$ . Our bounds do not require

<sup>4</sup> The analogous result for computing the maximum also holds. Its proof is nearly identical to the proof for theorem about the minimum.

<sup>5</sup> [6] consider the tradeoff with the size of the set being added, which allows them to prove lower bounds even when the data structure is only required to maintain small sets. The bound stated here is what they obtain when the size of the set is allowed to be arbitrary.

non-adaptivity for the insertion operations, and are quantitatively better when  $t_{\text{pred}} = o(\log n / \log \log n)$ . We also note that the *cell sampling* technique ([22, 18]) does not give any meaningful lower bounds for these problems.

Our theorems are complemented by the observation that a variant of Binary Search trees gives a data structure that can insert and delete elements non-adaptively, compute predecessors non-adaptively, and perform all operations in time  $O(\log n)$ , with  $w = \log n$ . Theorem 2 and Theorem 3 show that there is a gap between adaptive and non-adaptive data structures computing the median and predecessors, since we know that the van Emde Boas data structure can compute both in time  $O(\log \log n)$  with  $w = \log n$ .

The rest of the paper is organized as follows. After the preliminaries, we begin proving lower bounds in Section 3, where we give an introduction to our techniques by proving lower bounds for several problems when all operations are assumed to be non-adaptive. We prove Theorem 1 there. We then prove Theorem 2 in Section 4, and Theorem 3 in Section 5. We discuss a simple data structure based on binary search trees for these problems in Appendix A.

## 2 Preliminaries

Unless otherwise stated, logarithms in this article are computed base two. Given  $a = a_1, a_2, \dots, a_n$ , we write  $a_{\leq i}$  to denote  $a_1, \dots, a_i$ . We define  $a_{> i}$  and  $a_{\leq i}$  similarly. Similarly, we write  $a_{-i}$  to denote  $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n$ .  $[\ell]$  denotes the set  $\{1, 2, \dots, \ell\}$ , for  $\ell \in \mathbb{N}$ .

The *entropy* of a discrete random variable  $A$ , is defined to be

$$H(A) = \sum_a \Pr[A = a] \cdot \log \frac{1}{\Pr[A = a]}.$$

For two random variables  $A, B$ , the entropy of  $A$  conditioned on  $B$  is defined as

$$H(A|B) = \sum_{a,b} \Pr[A = a, B = b] \cdot \log \frac{1}{\Pr[A = a|B = b]}.$$

The entropy satisfies some useful properties:

► **Proposition 4** (Chain Rule).  $H(A_1 A_2 | B) = H(A_1 | B) + H(A_2 | B A_1)$ .

► **Lemma 5** (Subadditivity).  $H(A_1 A_2 | B) \leq H(A_1 | B) + H(A_2 | B)$ .

► **Proposition 6**. For every  $a, b \geq 1$  and  $c > 2$ , if  $a \log ab \geq c$ , then  $a \geq \frac{c}{\log c + \log b}$ .

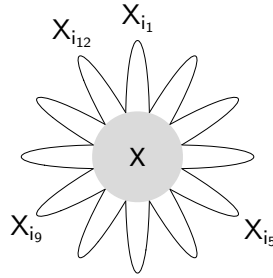
**Proof.** Suppose that  $a < \frac{c}{\log c + \log b}$ . We then have,

$$\begin{aligned} a \log ab &< \frac{c}{\log c + \log b} \cdot (\log b + \log c - \log(\log c + \log b)) \\ &< c, \end{aligned}$$

where the last inequality follows from the fact that  $c > 2$ . This contradicts  $a \log ab \geq c$ , and therefore,  $a \geq \frac{c}{\log c + \log b}$ . ◀

► **Proposition 7**. For  $1 \leq k \leq n$ ,  $\log \binom{n}{k} \leq k \cdot \log \frac{en}{k}$ .





■ **Figure 1** A Flower with 12 petals.  $X$  denotes the core of the Flower.

## 2.1 Sunflowers

Our proof relies on a variant<sup>6</sup> of the Sunflower lemma [11]. The lemma we need is almost identical to a lemma proved by [2], and we use their ideas to prove it.

► **Definition 8.** A sequence of sets  $X_1, \dots, X_p$  is called a  $t$ -flower with  $p$  petals if each set in the sequence is of size  $t$ , and there is a set  $X$  of size at most  $t$  such that for every  $i, j$ ,  $X_i \cap X_j \subseteq X$ .  $X$  is called the *core* of the flower.

See Figure 1 for an illustration of a flower. Next, following [2], we show that a long enough sequence of sets must contain a flower.

► **Lemma 9 (Flower Lemma).** *Let  $X_1, \dots, X_n$  be a sequence of sets each of size  $t$ . If  $n > (p-1)^{t+1}$ , then there is a subsequence that is a  $t$ -flower with  $p$  petals.*

**Proof.** We prove the bound by induction on  $t, p$ . When  $t = 1$ , if  $n > (p-1)^2$ , either there are  $p$  sets that are the same or  $p$  sets that are distinct. Either way, we obtain a 1-flower with  $p$  petals. When  $p = 1$  the statement is trivially true.

Suppose that  $t \geq 2$ , and the sequence does not contain a  $t$ -flower with  $p$  petals. For each set  $X \subseteq X_1$ , we get a subsequence by restricting our attention to the sets  $X_i$  such that  $X_i \cap X_1 = X$  and  $i > 1$ . By induction, the length of this subsequence can be at most  $(p-2)^{t+1-|X|}$  since all of these sets have  $X$  in common, and any  $(t-|X|)$ -flower with  $p-1$  petals yields a  $t$ -flower with  $p$  petals in our original sequence, by adding  $X_1$  to the list of petals. Thus we get,

$$\begin{aligned} n &\leq 1 + \sum_{X \subseteq X_1} (p-2)^{t+1-|X|} \\ &= 1 + (p-2) \cdot \sum_{X \subseteq X_1} (p-2)^{t-|X|} \\ &\leq 1 + (p-2) \cdot (p-2+1)^t \leq (p-1)^{t+1}, \end{aligned}$$

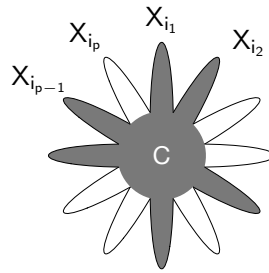
as desired. ◀

## 3 Lower Bounds when All Operations are Non-Adaptive

As a warm up, we prove some loose lower bounds when all operations in the data structure are non-adaptive. In the next section, we prove our final theorems where we only assume that some of the operations are non-adaptive.

<sup>6</sup> Using the Sunflower lemma would give us bounds with the same asymptotics, but the Flower Lemma (Lemma 9) gives cleaner bounds.





■ **Figure 2**  $C$  denotes the core of the Flower, and the shaded cells are the only cells accessed when deleting  $\{i_1, i_2, \dots, i_p\} \setminus S$ .

We start by proving Theorem 1, which gives a lower bound on the time for any data structure that computes minimum and deletions non-adaptively.

**Proof of Theorem 1.** Consider the sequence of sets  $\mathcal{X} = X_1, \dots, X_n$  where

$$X_i = \{j \mid \text{cell } j \text{ is accessed while deleting } i, \text{ or when computing the minimum}\}.$$

If  $t$  is the time required for the operations of the data structure, then each set  $X_i$  is of size at most  $2t$ . Without loss of generality, we can assume that each  $X_i$  is of size *exactly*  $2t$ . The key observation is that there cannot be a large  $2t$ -flower in  $\mathcal{X}$ :

► **Claim 10.** *If  $\mathcal{X}$  has a  $2t$ -flower with  $p$  petals, then  $p \leq 2wt$ .*

**Proof.** Suppose for the sake of contradiction that the sequence  $X_{i_1}, \dots, X_{i_p}$  is a  $2t$ -flower with  $i_1 < i_2 < \dots < i_p$ , and  $p = 2wt + 1$ . Then let  $S$  be any subset of  $\{i_1, i_2, \dots, i_p\}$  and  $C$  denote the the contents of the core of the  $2t$ -flower after inserting the set  $\{i_1, \dots, i_p\}$  and then deleting the elements of  $\{i_1, i_2, \dots, i_p\} \setminus S$ .

We show that  $C$  serves as an encoding of  $S$ . This is because  $C$  is all we need to reconstruct the execution of the following sequence of deletion and minimum operations: compute the minimum, delete the minimum, compute the minimum, delete the minimum, and so on. The answers to these computations determine the elements in  $S$ . The answer to the first minimum computation can be reconstructed from  $C$ , since  $C$  contains all cells used in this computation. If we attempt to delete  $i_j$ , then the only cells of  $X_{i_j}$  that were modified by a previous deletion operation are contained in  $C$ . Thus, every such deletion operation can be simulated with access to  $C$  (See Figure 2).

$C$  can be described using at most  $2t \cdot w$  bits, yet  $C$  encodes an arbitrary subset of  $p$  elements. This proves the claim. ◀

By the Flower-Lemma (Lemma 9), the sequence  $\mathcal{X}$  has a  $2t$ -flower with  $n^{\frac{1}{2t+1}}$  petals. So, we get

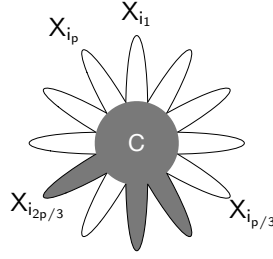
$$t \geq \frac{p}{2w} \geq \frac{n^{\frac{1}{2t+1}}}{2w},$$

where the last inequality follows from the choice of  $p$ . After rearranging, we get

$$t \cdot \log wt \geq \Omega(\log n).$$

Proposition 6 implies the desired bound on  $t$ . ◀

Next we prove a similar result for computing the median.



■ **Figure 3**  $C$  denotes the core of the Flower, and the shaded cells are the only cells accessed when inserting  $S$ .

► **Theorem 11.** *Any data structure with non-adaptive insertions and median computations must take time  $\Omega\left(\frac{\log n}{\log \log n + \log w}\right)$  for some operation.*

**Proof.** Consider the sequence of sets  $\mathcal{X} = X_1, \dots, X_n$  where

$$X_i = \{j \mid \text{cell } j \text{ is accessed while inserting } i, \text{ or when computing the median}\}.$$

If  $t$  is the time required for the operations of the data structure, then each set  $X_i$  is of size at most  $2t$ . Without loss of generality, we can assume that each  $X_i$  is of size *exactly*  $2t$ . The key observation is that there cannot be a large  $2t$ -flower in  $\mathcal{X}$ :

► **Claim 12.** *If  $\mathcal{X}$  has a  $2t$ -flower with  $p$  petals, then  $p \leq 6wt + 2$ .*

**Proof.** Suppose for the sake of contradiction that the sequence  $X_{i_1}, \dots, X_{i_p}$  is a  $2t$ -flower with  $i_1 < i_2 < \dots < i_p$ , and  $p = 6wt + 3$ . Then let  $S$  be any subset of  $\{i_{p/3+1}, i_{p/3+2}, \dots, i_{2p/3}\}$  and  $C$  denote the contents of the core of the  $2t$ -flower after inserting elements of  $S$  into the data structure (see Figure 3).

We show that  $C$  serves as an encoding of  $S$ . This is because  $C$  is all we need to reconstruct the execution of the following sequence of insert and median operations: insert  $i_1$ , compute the median, insert  $i_2$ , compute the median,  $\dots$ , insert  $i_{p/3}$ , compute the median. These operations determine the elements in  $S$  between its smallest element and median. By the definition of the flower, the only cells of  $X_{i_1}, \dots, X_{i_{p/3}}$  that were accessed when  $S$  was inserted are contained in  $C$ . Therefore, the sequence of operations can be simulated using  $C$  (see Figure 3). Similarly, executing the following operations helps retrieve elements in  $S$  between its median and largest element: insert  $i_{2p/3+1}$ , compute the median, insert  $i_{2p/3+2}$ , compute the median,  $\dots$ , insert  $i_p$ , compute the median.

$C$  can be described using at most  $2t \cdot w$  bits, yet  $C$  encodes a subset of  $p/3 = (2tw + 1)$  elements. This proves the claim. ◀

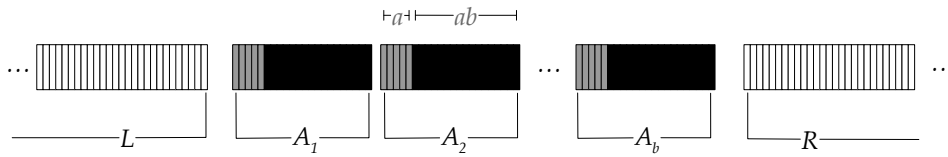
By the Flower-Lemma (Lemma 9), the sequence  $\mathcal{X}$  has a  $2t$ -flower with  $n^{\frac{1}{2t+1}}$  petals. Then we get

$$t \geq \frac{p-2}{6w} \geq \frac{n^{\frac{1}{2t+1}} - 2}{6w},$$

where the last inequality follows from the choice of  $p$ . After rearranging, we get

$$t \cdot \log wt \geq \Omega(\log n).$$

Proposition 6 implies the desired bound on  $t$ . ◀



■ **Figure 4** The elements corresponding to petals are partitioned into disjoint intervals  $L, A_1, \dots, A_q, R$ .  $T$  is the set of black elements.  $S_i$  is a random subset of the  $i$ 'th gray elements from each interval  $A_j$ .

Next we prove a lower bound for the predecessor search problem.

► **Theorem 13.** *Any data structure for the predecessor problem with non-adaptive insert operations and non-adaptive predecessor operations must have time  $\Omega\left(\frac{\log n}{\log \log n + \log w}\right)$ .*

**Proof.** Let  $\mathcal{X} = X_1, \dots, X_n$ , where

$$X_i = \{j \mid \text{cell } j \text{ is accessed while inserting } i \text{ or computing the predecessor of } i\}.$$

If  $t$  is the time required for the operations of the data structure, then each set  $X_i$  is of size at most  $2t$ . Without loss of generality, we can assume that each  $X_i$  is of size *exactly*  $2t$ . We first show that the time complexity can be lower bounded in terms of the number of petals in a  $2t$ -flower belonging to  $\mathcal{X}$ .

► **Claim 14.** *If  $\mathcal{X}$  has a  $2t$ -flower with  $p$  petals, then  $p \leq 4tw + 1$ .*

**Proof.** Supposed for the sake of contradiction that the sequence  $X_{i_1}, \dots, X_{i_p}$  is a  $2t$ -flower and  $i_1 < i_2 < \dots < i_p$ , and  $p = 4tw + 2$ . Let  $S$  be any subset of  $\{i_1, i_3, \dots, i_{p-1}\}$  and  $C$  denote the contents of the cells in the core after inserting the elements of  $S$ .

We show that  $C$  serves as an encoding of  $S$ . To reconstruct  $S$ , it suffices to compute the predecessors of the following elements:  $i_2, i_4, \dots, i_p$ . By the definition of the  $2t$ -flower, the only cells accessed in  $X_{i_2}, X_{i_4}, \dots, X_{i_p}$  during the insertion operations are contained in the core of the  $2t$ -flower. Therefore, the sequence of predecessor operations can be simulated by access only to the cells in the core.

Hence  $C$  encodes  $S$ . Since there are  $2^{2tw+1}$  possible sets  $S$ , and  $C$  can be described using  $2tw$  bits, we must have  $2tw \geq p/2$ . This proves the claim. ◀

By the Flower Lemma 9, the sequence  $\mathcal{X}$  has a  $2t$ -flower with  $n^{\frac{1}{2t+1}}$  petals. So  $t \geq \frac{p-1}{4w} \geq \frac{n^{\frac{1}{2t+1}} - 1}{4w}$ , which follows from the choice of  $p$ . After rearranging, we get

$$t \cdot \log wt \geq \Omega(\log n).$$

Proposition 6 implies the desired bound on  $t$ . ◀

#### 4 Lower Bounds for Median when Insertions are Non-Adaptive

In this section, we prove Theorem 2. We start by giving an outline of the proof. As before, we first associate every element in  $\{1, 2, \dots, n\}$  with the set of cells that are accessed while inserting the element. We then identify a flower among these sets. Proving a lower bound on the time to compute the median is challenging as the computation is adaptive. We shall have to use the flower found above in a subtle way. We come up with a carefully chosen sequence of insertions, followed by a median computation that recovers the  $k$ 'th smallest element of

the set. The sequence of insertions are performed in batches, and every cell that is not in the core of the flower is associated with the batch number of the insertion operation that last accessed it. Ignoring the cells that belong to the core of the flower, we show that at least one cell associated with every batch is accessed with constant probability. Since these cells are disjoint, this will prove that the time to compute the median is at least a constant fraction of the number of batches. To make the above argument work, we use Shannon entropy to quantify the amount of information that the median computation must recover from the cells associated with each batch of insertions.

We now proceed with the formal proof. Define the sequence of sets  $X = X_1, \dots, X_n$ , where

$$X_i = \{j \mid \text{cell } j \text{ is accessed while inserting } i\}.$$

By the flower lemma (Lemma 9), this sequence of sets must contain a  $t_{\text{ins}}$ -flower with  $p = n^{1/(t_{\text{ins}}+1)}$  petals, and without loss of generality, we assume that the petals are  $X_1, \dots, X_p$ . Let  $C$  denote the core of the  $t_{\text{ins}}$ -flower.

To carry out the proof, we need to carefully define a sequence of operations that insert a subset of the elements  $\{1, 2, \dots, p\}$ <sup>7</sup>. For parameters  $a, b$ , let  $L, A_1, \dots, A_b, R \subseteq \{1, 2, \dots, p\}$  be consecutive disjoint intervals in ascending order, such that  $L$  is of size  $p/3$ ,  $R$  is of size  $p/3$  and for each  $i$ ,  $A_i$  is of size  $a + ab$ , and  $b(a + ab) \leq p/3$ . See Figure 4. Let  $S_1, \dots, S_a$  be independently sampled sets, such that  $S_i$  is a uniformly random subset of  $\{j : j \text{ is the } i\text{'th element of } A_r \text{ for some } r\}$ . So each  $S_i$  is a subset of the gray elements in Figure 4. Finally, let  $T$  be the set

$$T = \{j : \text{for some } i \in [b], j \in A_i \text{ and } j \text{ is not one of the first } a \text{ elements of } A_i\},$$

so  $T$  is the set of black elements in Figure 4. Let  $k$  be a uniformly random element of  $\{a, a + (a + ab), a + 2(a + ab), \dots, a + (b - 1)(a + ab)\}$ .

Consider the following sequence of operations with the data structure:

1. Phase 1:
  - a. Insert the elements of  $T$ .
  - b. Insert the elements of  $S_1$ , then the elements of  $S_2$ , and so on, until  $S_a$  has been inserted.
2. Phase 2:
  - a. Insert an appropriate number of elements into  $L$  or  $R$  so that the median of all the elements inserted is the  $k$ 'th smallest element of  $T \cup S_1 \cup S_2 \dots \cup S_a$ .
  - b. Compute the median of the inserted set.

We shall prove that the expected number of cells accessed to compute the median must be close to  $a$ . In order to prove this, we use ideas inspired by the chronogram approach. Consider the cells accessed during Phase 1. We say that a cell *belongs to*  $S_i$  if it is in the set

$$\bigcup_{j: j \text{ is the } i\text{'th element of } A_r \text{ for some } r} X_j \setminus C$$

So, every cell of the data structure can belong to at most one of the sets  $S_1, \dots, S_a$ . Moreover, every cell that is accessed when inserting  $S_i$  either belongs to  $S_i$  or is in the core of the  $t_{\text{ins}}$ -flower.

---

<sup>7</sup> This sequence of operations is inspired by an argument in [27]

Define

$$E_i = \begin{cases} 1 & \text{if a cell that belongs to } S_i \text{ is accessed in Phase 2,} \\ 0 & \text{otherwise.} \end{cases}$$

Observe that the insertions in Phase 2(a) never access a cell that belongs to  $S_i$  for any  $i$ . Since  $E_i = 1$  whenever a cell that belongs to  $S_i$  is accessed, all such accesses must come from the median computation in Phase 2. Thus,  $t_{\text{med}} \geq \sum_{i=1}^a \mathbb{E}[E_i]$ . We now proceed to lower bound  $\sum_{i=1}^a \mathbb{E}[E_i]$ .

Let  $C_i$  denote the contents of the core immediately after  $S_i$  was inserted. Let  $S_i^j$  denote the set  $S_i \cap A_j$  and  $S_i^{<j}$  denote the set  $S_i^1 \cup S_i^2 \cup \dots \cup S_i^{j-1}$ . Recall that  $S_{-i}$  denotes  $S_1, \dots, S_{i-1}, S_{i+1}, \dots, S_a$ .

► **Claim 15.** *The variables  $S_{-i}, C_i$  determine the contents, after Phase 1, of all cells that do not belong to  $S_i$ .*

**Proof.** If a cell does not belong to  $S_i$ , then there are three possibilities. If it belongs to a set  $S_{i'}$  for  $i' < i$ , then its value can be reconstructed from  $S_1, \dots, S_{i'}$ . If it belongs to  $S_{i'}$  for  $i' > i$ , its value can be reconstructed from  $C_i$  and  $S_{i+1}, \dots, S_a$ . If it does not belong to any set, then if it is in the core, it is determined by  $C_i$  and  $S_{i+1}, \dots, S_a$ , and if it is not in the core, its value is fixed. ◀

Let  $k = a + (j - 1)(a + ab)$ , so  $j$  is a uniformly random number from the set  $\{1, 2, \dots, b\}$ .

► **Claim 16.** *The  $k$ 'th smallest element of  $T \cup S_1 \cup S_2 \dots \cup S_a$  computed in Phase 2 and  $S_{-i}$  together determine  $\sum_{\ell=1}^j |S_i^\ell|$ .*

**Proof.** The  $k$ 'th smallest element of  $T \cup S_1 \cup S_2 \dots \cup S_a$  is  $e$  if and only if the number of elements in  $A_1 \cup A_2 \dots \cup A_b$  that are less than  $e$  and missing in  $T \cup S_1 \cup S_2 \dots \cup S_a$  is exactly  $e - k$ . In other words, the  $k$ 'th smallest element of  $T \cup S_1 \cup S_2 \dots \cup S_a$  is  $e$  if and only if

$$|\{j : j < e, j \in (A_1 \cup A_2 \dots \cup A_b) \setminus (T \cup S_1 \cup S_2 \dots \cup S_a)\}| = e - k.$$

Let  $\alpha$  be the number of elements missing from the intervals  $A_1, A_2, \dots, A_b$ , and  $e$  be the  $k$ 'th smallest element of  $T \cup S_1 \cup S_2 \dots \cup S_a$ . We know that  $0 \leq \alpha \leq ab$ , and hence  $k \leq e \leq k + ab$ . Therefore, the  $k$ 'th smallest element must be the  $a$ 'th smallest element in  $A_j$  or belong to  $T \cap A_j$ , and must determine the total number of elements missing before this point. This proves the claim. ◀

► **Claim 17.**

$$\mathbb{E}[E_i] \geq \mathbb{E}_j \left[ \mathbb{H} \left( S_i^j | S_i^{<j}, S_{-i}, C_i, |S_i| \right) \right].$$

**Proof.** The intuition behind the proof is that in Phase 2, the algorithm starts out knowing only the size of the sets, and learns the  $k$ 'th smallest element of the sets after computing the median. The contents of all cells needed to insert elements in Phase 2 are determined by  $S_{-i}, C_i$ , since these variables determine the cells in the core. By Claim 15, after fixing  $S_i^{<j}, S_{-i}, C_i, |S_i|$ , all the cells that do not belong to  $S_i$  are determined. Thus, after fixing  $S_i^{<j}, S_{-i}, C_i, |S_i|$ , the value of  $E_i$  is determined. Now if  $E_i = 0$ , then the  $k$ 'th smallest element is determined, which means that  $\mathbb{H} \left( S_i^j | S_i^{<j}, S_{-i}, C_i, |S_i| \right) = 0$ . If  $E_i = 1$ , the inequality holds trivially. ◀



(a)  $\sum_{i=1}^3 Z_i = 2.$

(b)  $\sum_{i=1}^3 Z_i = 0.$

■ **Figure 5**  $S \subseteq \{1, 5, 9\}$ . Cells in petal  $X_i$  are shaded black when  $\text{Pred}'(i) \neq \text{Pred}(i)$ .

Recall that  $t_{\text{med}} \geq \sum_{i=1}^a \mathbb{E}[E_i]$ . Then by the above claim, linearity of expectation and the chain rule for entropy, we have:

$$\begin{aligned} t_{\text{med}} &\geq \sum_{i=1}^a \mathbb{E}[E_i] \geq \sum_{i=1}^a \mathbb{E}_j \left[ \mathbb{H}(S_i^j | S_i^{<j}, C_i, S_{-i}, |S_i|) \right] = (1/b) \sum_{i=1}^a \mathbb{H}(S_i | C_i, S_{-i}, |S_i|) \\ &\geq (1/b) \sum_{i=1}^a \mathbb{H}(S_i | S_{-i}) - \mathbb{H}(C_i, |S_i|) \\ &\geq a \cdot \left( 1 - \frac{t_{\text{ins}} w + \log b}{b} \right), \end{aligned} \quad (1)$$

where the last inequality follows from the facts that

$$\mathbb{H}(S_i | S_{-i}) = \mathbb{H}(S_i) = b, \text{ and } \mathbb{H}(C_i, |S_i|) \leq \mathbb{H}(C_i) + \mathbb{H}(|S_i|) \leq w t_{\text{ins}} + \log b.$$

Set  $b = 4w t_{\text{ins}}$  and  $a$  to be the largest integer such that  $a \leq \frac{p}{3b(b+1)}$ . Since  $b \geq 4$ ,  $\frac{\log b}{b} \leq \frac{1}{2}$ . Now, (1) implies that

$$t_{\text{med}} \geq a/4 \geq \Omega \left( \frac{n^{1/(t_{\text{ins}}+1)}}{w^2 \cdot t_{\text{ins}}^2} \right),$$

where the last inequality follows from the fact that  $a \geq \frac{p}{3b(b+1)} - 1$ .

## 5 Lower Bounds for Predecessor Search when Predecessors are Non-Adaptive

In this section we prove Theorem 3. Consider the sequence  $\mathcal{X} = X_1, \dots, X_n$ , where

$$X_i = \{j | \text{cell } j \text{ is accessed while computing the predecessor of } i\}.$$

By the Flower Lemma (Lemma 9),  $\mathcal{X}$  contains a  $t_{\text{pred}}$ -flower with  $n^{\frac{1}{t_{\text{pred}}+1}}$  petals. Let  $a$  be the largest even integer such that  $a(a+1) \leq n^{\frac{1}{t_{\text{pred}}+1}}$ . Note that  $a \geq \frac{n^{\frac{1}{2(t_{\text{pred}}+1)}}}{2}$ . For ease of notation, we assume that  $X_1, X_2, \dots, X_{a(a+1)}$  are the promised  $t_{\text{pred}}$ -flower.

Let  $S$  be any subset of  $\{i | i = (j-1)(a+1) + 1 \text{ for some } j \in [a]\}$ . Insert all elements of  $S$ . For  $j \in [a(a+1)]$ , let  $\text{Pred}'(j)$  be the value obtained by simulating the predecessor computation assuming that the cells outside the core were never accessed when  $S$  was inserted. Note that  $\text{Pred}'(j)$  can be computed from the cells in the core. Let  $\text{Pred}(j)$  be the predecessor

of  $j$ . For every  $i \in [a]$ , define

$$Z_i = \begin{cases} 1, & |\{j \in \{i(a+1) - a + 1, \dots, i(a+1)\} \mid \text{Pred}(j) \neq \text{Pred}'(j)\}| > a/2 \\ 0, & \text{otherwise.} \end{cases}$$

Figure 5 shows an example with  $a = 3$ . Since  $|S| \leq a$  and the total number of cells accessed while inserting  $S$  is at least  $\frac{a}{2} \cdot \sum_{i=1}^a Z_i$ ,

$$\sum_{i=1}^a Z_i \cdot (a/2) \leq t_{\text{ins}} \cdot a. \quad (2)$$

Let  $C$  denote the contents of the core after inserting elements of  $S$ , the names of the elements  $i$  with  $Z_i = 1$ , and whether or not  $i \in S$  for every element with  $Z_i = 1$ . In other words,  $C$  encodes the core, the set  $\{i : Z_i = 1\}$  and the set  $S \cap \{i : Z_i = 1\}$ .

► **Lemma 18.**  $C$  encodes  $S$ .

**Proof.** It suffices to come up with a decoding procedure that given  $C$  recovers  $S$ . The decoding algorithm first recovers elements of  $S$  in  $\{i \mid Z_i = 1\}$  from the description of  $C$ . By definition, if  $i \in S$  and  $i \notin \{i \mid Z_i = 1\}$ , then  $\text{Pred}'(j) = i$  for the majority values of  $j \in \{i(a+1) - a + 1, \dots, i(a+1)\}$ . If  $i \notin \{i \mid Z_i = 1\}$ , then the decoding algorithm computes  $\text{Pred}'(j)$  for every  $j \in \{i(a+1) - a + 1, \dots, i(a+1)\}$ . If the majority of the answers equal  $i$ , then the decoding algorithm infers that  $i \in S$ . Otherwise, it infers that  $i \notin S$ . This determines whether or not  $i \in S$ . ◀

We now analyze the length of the encoding of  $C$ . The contents of the core can be described with  $wt_{\text{pred}}$  bits. It takes at most  $2 \log a$  bits to encode  $|\{i \mid Z_i = 1\}|$  and  $|S \cap \{i \mid Z_i = 1\}|$ . Given their sizes, it takes  $\log \left( \sum_{i=1}^a Z_i \right)$  bits to encode  $\{i \mid Z_i = 1\}$ , and  $\log \left( |S \cap \{i \mid Z_i = 1\}| \right)$  bits to encode  $S \cap \{i \mid Z_i = 1\}$ . Therefore, the length of the encoding is at most

$$wt_{\text{pred}} + \log \left( \sum_{i=1}^a Z_i \right) + \log \left( |S \cap \{i \mid Z_i = 1\}| \right) + 2 \log a.$$

Since there are  $2^a$  possible sets  $S$ , we must have

$$a \leq wt_{\text{pred}} + \log \left( \sum_{i=1}^a Z_i \right) + \log \left( |S \cap \{i \mid Z_i = 1\}| \right) + 2 \log a. \quad (3)$$

Observe that either  $t_{\text{ins}} \geq \frac{a}{64}$  or not. In the former case, since  $a \geq \frac{n^{\frac{1}{2(t_{\text{pred}}+1)}}}{2}$ , we can conclude that  $t_{\text{ins}} \geq \Omega \left( n^{\frac{1}{2(t_{\text{pred}}+1)}} \right)$ . In the latter case, Equation 2 implies that  $\sum_{i=1}^a Z_i \leq a/32$ . Note that  $\left( |S \cap \{i \mid Z_i = 1\}| \right) \leq \left( \sum_{i=1}^a Z_i \right)$  when  $\sum_{i=1}^a Z_i \leq a/32$ . Using Proposition 7, we get

$$\log \left( \sum_{i=1}^a Z_i \right) + \log \left( |S \cap \{i \mid Z_i = 1\}| \right) \leq 2 \left( \sum_{i=1}^a Z_i \right) \cdot \log \frac{ea}{\sum_{i=1}^a Z_i} \leq \frac{a}{2},$$

where the last inequality follows from the fact that  $\sum_{i=1}^a Z_i \leq a/32$ . After rearranging (3), the previous inequality implies that  $t_{\text{pred}} \geq \frac{a}{2w} - \frac{2 \log a}{w}$ . Since  $a \geq \frac{n^{\frac{1}{2(t_{\text{pred}}+1)}}}{2}$ , we can conclude that  $t_{\text{pred}} \cdot \log(wt_{\text{pred}}) \geq \Omega(\log n)$ . Using Proposition 6, we obtain the desired lower bound of  $t_{\text{pred}} \geq \Omega \left( \frac{\log n}{\log \log n + \log w} \right)$ .

## References

- 1 Miklós Ajtai. A lower bound for finding predecessors in yao’s call probe model. *Combinatorica*, 8(3), 1988.
- 2 Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.
- 3 Noga Alon and Uriel Feige. On the power of two, three and four probes. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*. SIAM, 2009.
- 4 Paul Beame and Faith E. Fich. Optimal bounds for the predecessor problem and related problems. *JCSS: Journal of Computer and System Sciences*, 65, 2002.
- 5 Paul Beame, Vincent Liew, and Mihai Patrascu. Finding the median (obliviously) with bounded space. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 103–115, 2015.
- 6 Joseph Boninger, Joshua Brody, and Owen Kephart. Non-adaptive data structure bounds for dynamic predecessor. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India*, pages 20:1–20:12, 2017.
- 7 Gerth Stølting Brodal, Shiva Chaudhuri, and Jaikumar Radhakrishnan. The randomized complexity of maintaining the minimum. In *SWAT: Scandinavian Workshop on Algorithm Theory*, 1996.
- 8 Joshua Brody and Kasper Green Larsen. Adapt or die: Polynomial lower bounds for non-adaptive dynamic data structures. *Theory of Computing*, 11:471–489, 2015.
- 9 Amit Chakrabarti, T. S. Jayram, and Mihai Patrascu. Tight lower bounds for selection in randomly ordered streams. In *Proc. 19th Symp. on Discrete Algorithms (SODA)*, pages 720–729. ACM/SIAM, 2008.
- 10 Timothy M. Chan. Comparison-based time-space lower bounds for selection. *ACM Trans. Algorithms*, 6(2), 2010.
- 11 Paul Erdős and Richard Rado. Intersection theorems for systems of sets. *Journal of London Mathematical Society*, 35:85–90, 1960.
- 12 Gudmund Skovbjerg Frandsen, Peter Bro Miltersen, and Sven Skyum. Dynamic word problems. *J. ACM*, 44(2):257–271, 1997. doi:10.1145/256303.256309.
- 13 Michael Fredman and Michael Saks. The cell probe complexity of dynamic data structures. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1989.
- 14 Michael L. Fredman and Dan E. Willard. Surpassing the information theoretic bound with fusion trees. *JCSS: Journal of Computer and System Sciences*, 47, 1993.
- 15 Anna Gál and Peter Bro Miltersen. The cell probe complexity of succinct data structures. *Theor. Comput. Sci.*, 379(3):405–417, 2007. doi:10.1016/j.tcs.2007.02.047.
- 16 Mohit Garg and Jaikumar Radhakrishnan. Set membership with non-adaptive bit probes. In *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, pages 38:1–38:13, 2017.
- 17 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2000.
- 18 Kasper Green Larsen. The cell probe complexity of dynamic range counting. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 85–94, 2012.
- 19 Peter Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. 57:37–49, 1 1998.

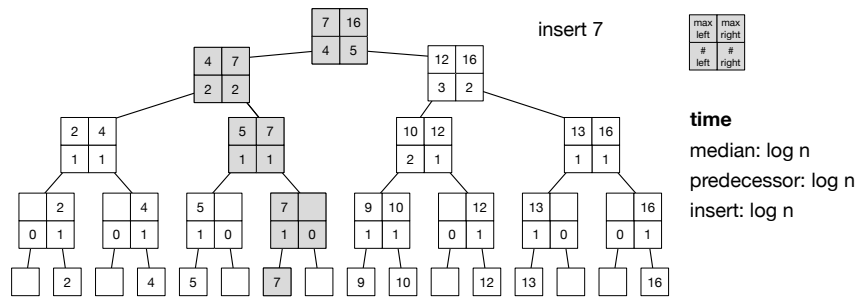


- 20 Peter Bro Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proceedings of the 26th Annual Symposium on the Theory of Computing*, pages 625–634, New York, 1994. ACM Press.
- 21 J. Ian Munro and Venkatesh Raman. Selection from read-only memory and sorting with minimum data movement. *TCS: Theoretical Computer Science*, 165, 1996.
- 22 Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower bounds on near neighbor search via metric expansion. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, FOCS '10*, pages 805–814, Washington, DC, USA, 2010. IEEE Computer Society.
- 23 Mihai Pătraşcu. Lower bounds for 2-dimensional range counting. In *Proc. 39th ACM Symposium on Theory of Computing (STOC)*, pages 40–46, 2007.
- 24 Mihai Pătraşcu and Erik D. Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM Journal on Computing*, 35(4):932–963, 2006. See also STOC'04, SODA'04.
- 25 Mihai Patrascu and Mikkel Thorup. Time-space trade-offs for predecessor search. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 232–240, 2006.
- 26 Mihai Pătraşcu and Mikkel Thorup. Don't rush into a union: Take time to find your roots. In *Proc. 43rd ACM Symposium on Theory of Computing (STOC)*, pages 559–568, 2011. See also arXiv:1102.1783.
- 27 Mihai Patrascu and Mikkel Thorup. Dynamic integer sets with optimal rank, select, and predecessor search. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 166–175, 2014.
- 28 Pranab Sen and Srinivasan Venkatesh. Lower bounds for predecessor searching in the cell probe model. *J. Comput. Syst. Sci.*, 74(3):364–385, 2008.
- 29 Peter van Emde Boas. Preserving order in a forest in less than logarithmic time and linear space. *Information Processing Letters*, 6(3):80–82, 1977.
- 30 Omri Weinstein and Huacheng Yu. Amortized dynamic cell-probe lower bounds from four-party communication. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 305–314, 2016.
- 31 Dan E. Willard. Log-logarithmic worst-case range queries are possible in space  $\Theta(N)$ . *Information Processing Letters*, pages 81–84, 1983.
- 32 Andrew Yao. Should tables be sorted? *JACM: Journal of the ACM*, 28, 1981.
- 33 Huacheng Yu. Cell-probe lower bounds for dynamic problems via a new communication model. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 362–374, 2016.

## **A** A Data Structure based on Binary Search Trees

Here we describe a data structure that maintains a subset of  $\{1, \dots, n\}$  allowing non-adaptive inserts, non-adaptive predecessor computations and adaptive median computations. The data structure builds on the well known binary search tree on  $\{1, \dots, n\}$  and is very close to the *x-fast trie* (see [31]). This data structure matches many of the lower bounds in our proofs.

► **Theorem 19.** *There is a data structure that maintains a subset of  $\{1, 2, \dots, n\}$  and supports insertions, deletions and computing the median, minimum, and predecessors. All operations take time  $O(\log n)$ , the word size is  $\log n$ , and all operations except for the median operation are non-adaptive.*



■ **Figure 6** A data structure based on binary search trees storing the set  $\{2, 4, 5, 7, 9, 10, 12, 13, 16\}$ .

**Proof.** Without loss of generality, we may assume that  $n$  is a power of 2. We maintain a balanced binary tree of height  $\log n$ . Every leaf is assigned an element from the universe.

There is a memory cell associated with every leaf and four memory cells associated with every internal node of the tree. The cells corresponding to each internal node store the number of elements in the left subtree rooted at that node, the number of elements stored in the right subtree, the maximum element of the left subtree and the maximum element of the right subtree. Figure 6 shows an example of the data structure.

To insert an element into the set, we only need to access the cells associated with each node on the path from the root to the corresponding leaf. These are the only cells that need to be modified to make the data structure consistent with the new set. Deletions can be performed in the same way. The time required for these operations is  $O(\log n)$ , and they are non-adaptive.

To compute the median or minimum, we read the cells associated with the root to determine if the desired value belongs to the left or the right sub tree. Accordingly, we read the cells associated with either the left or the right child and recurse to find the median or minimum. The time required for this operation is  $O(\log n)$ , but it is adaptive.

To compute the predecessor of an element, we only need to access the cells associated with every node on the path from the root to the corresponding leaf in the tree. The predecessor is the maximum of last non-empty left-subtree seen on this path. Again, we see that this operation takes  $O(\log n)$  time, and is non-adaptive. ◀

# Dimension Reduction for Polynomials over Gaussian Space and Applications

Badih Ghazi<sup>1</sup>

Google Research, 1600 Amphitheatre Parkway Mountain View, CA 94043, USA  
badihghazi@gmail.com

Pritish Kamath<sup>2</sup>

Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA  
pritish@mit.edu

Prasad Raghavendra<sup>3</sup>

University of California Berkeley, Berkeley, CA, USA  
raghavendra@berkeley.edu

---

## Abstract

We introduce a new technique for reducing the dimension of the ambient space of low-degree polynomials in the Gaussian space while preserving their relative correlation structure. As an application, we obtain an explicit upper bound on the dimension of an  $\varepsilon$ -optimal noise-stable Gaussian partition. In fact, we address the more general problem of upper bounding the number of samples needed to  $\varepsilon$ -approximate any joint distribution that can be *non-interactively simulated* from a correlated Gaussian source. Our results significantly improve (from Ackermann-like to “merely” exponential) the upper bounds recently proved on the above problems by De, Mossel & Neeman [CCC 2017, SODA 2018 resp.] and imply decidability of the larger alphabet case of the *gap non-interactive simulation* problem posed by Ghazi, Kamath & Sudan [FOCS 2016].

Our technique of dimension reduction for low-degree polynomials is simple and can be seen as a generalization of the Johnson-Lindenstrauss lemma and could be of independent interest.

**2012 ACM Subject Classification** Theory of computation → Complexity theory and logic

**Keywords and phrases** Dimension reduction, Low-degree Polynomials, Noise Stability, Non-Interactive Simulation

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2018.28

**Acknowledgements** The authors are extremely grateful to Madhu Sudan for helpful guidance throughout all the stages of this project. The authors would also like to thank Anindya De, Elchanan Mossel and Joe Neeman for clarifying explanations of their papers and helpful discussions. PK would like to thank Irit Dinur, Mika Göös, Raghu Meka and Li-Yang Tan for helpful discussions.

## 1 Introduction

### 1.1 Gaussian Isoperimetry & Noise Stability

Isoperimetric problems over the Gaussian space have become central in various areas of theoretical computer science such as hardness of approximation and learning theory. In

---

<sup>1</sup> The work was done while the author was a student at MIT. Supported in parts by NSF CCF-1650733 and CCF-1420692.

<sup>2</sup> Supported in parts by NSF CCF-1420956, CCF-1420692, CCF-1218547 and CCF-1650733.

<sup>3</sup> Research supported by Okawa Research Grant and NSF CCF-1408643.



its simplest and classic form, the central question in isoperimetry is to determine what is the smallest possible surface area for a body of a given volume. Alternately, isoperimetric problems can also be formulated in terms of the notion of *Noise stability*.

Fix a real number  $\rho \in [0, 1]$  and let  $f : \mathbb{R}^n \rightarrow \{0, 1\}$  denote the indicator function of a subset (say  $\mathcal{A}_f$ ) of the  $n$ -dimensional Gaussian space ( $\mathbb{R}^n$  with the Gaussian measure  $\gamma_n$  given by the density function  $d\gamma_n/d\mathbf{X} = \exp(-\|\mathbf{X}\|_2^2/2)/(2\pi)^{n/2}$ ). The noise stability  $\text{Stab}_\rho(f)$  is the probability that two  $\rho$ -correlated Gaussians  $\mathbf{X}, \mathbf{Y}$  both fall inside or outside  $\mathcal{A}_f$ . More generally, the Gaussian “noise operator”  $U_\rho$  (also known as the Ornstein-Uhlenbeck operator), defined for each  $\rho \in [0, 1]$ , acts on any  $f : \mathbb{R}^n \rightarrow [0, 1]$  as

$$(U_\rho f)(\mathbf{X}) := \mathbb{E}_{\mathbf{Z} \sim \gamma_n} \left[ f \left( \rho \mathbf{X} + \sqrt{1 - \rho^2} \cdot \mathbf{Z} \right) \right].$$

The noise stability is then defined as

$$\text{Stab}_\rho(f) := \mathbb{E}_{\mathbf{X} \sim \gamma_n} [f(\mathbf{X}) \cdot U_\rho f(\mathbf{X}) + (1 - f(\mathbf{X})) \cdot (1 - U_\rho f(\mathbf{X}))]$$

Reformulated in terms of noise stability, the isoperimetric problem is to determine the largest possible value of  $\text{Stab}_\rho(f)$  for a function  $f : \mathbb{R}^n \rightarrow [0, 1]$  with a given expectation  $\mathbb{E}[f] = \alpha$ . The seminal isoperimetric theorem of Borell [10] shows that indicator functions of halfspaces are the most noise-stable among all functions  $f : \mathbb{R}^n \rightarrow [0, 1]$  with a given expectation over the Gaussian measure. Borell’s theorem (along with the invariance principle [44, 42]) has had fundamental applications in theoretical computer science, e.g., in the hardness of approximation for Max-Cut under the Unique Games conjecture [39] and in voting theory [42].

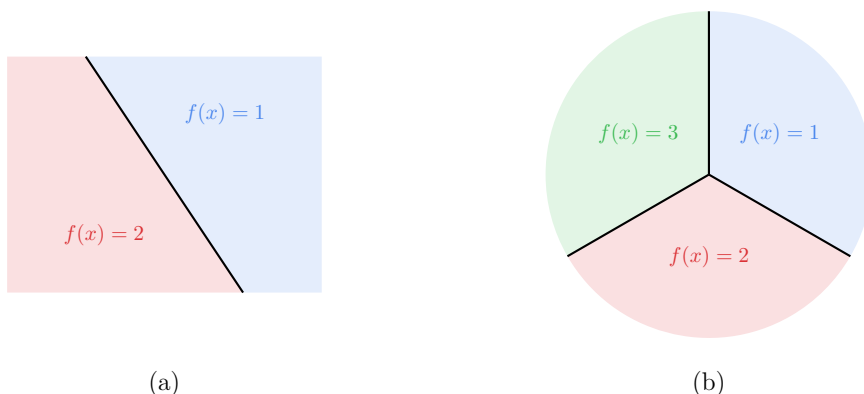
In this work, we are interested in analogues of Borell’s theorem for partitions of the Gaussian space *into more than two subsets*, or equivalently noise stability of functions  $f$  taking values over  $[k] := \{1, \dots, k\}$ . Towards stating these analogues, let’s state Borell’s theorem formally in a more general notation. Let  $\Delta_k$  be the probability simplex in  $\mathbb{R}^k$  (i.e., convex hull of the basis vectors  $\{e_1, \dots, e_k\}$ ). The Ornstein-Uhlenbeck operator naturally extends to vector valued functions  $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$  as  $U_\rho f := (U_\rho f_1, \dots, U_\rho f_k)$  where  $f = (f_1, \dots, f_k)$ . The noise stability of functions  $f : \mathbb{R}^n \rightarrow \Delta_k$  is now defined as  $\text{Stab}_\rho(f) := \mathbb{E}_{\mathbf{X} \sim \gamma_n} [\langle f(\mathbf{X}), U_\rho f(\mathbf{X}) \rangle]$  where  $\langle \cdot, \cdot \rangle$  denotes the inner product over  $\mathbb{R}^k$ . We can similarly define the noise stability of a function  $f : \mathbb{R}^n \rightarrow [k]$  by embedding  $[k]$  in  $\Delta_k$ , i.e., identifying coordinate  $i \in [k]$  with the standard basis vector  $e_i \in \Delta_k$ . Borell’s theorem can be formally stated in this notation as follows:

**Borell’s Theorem [10].** *For any  $f : \mathbb{R}^n \rightarrow \Delta_2$ , consider the halfspace function  $h = (h_1, h_2) : \mathbb{R}^n \rightarrow \Delta_2$  given by  $h_1(\mathbf{X}) = 1\{\langle a, \mathbf{X} \rangle \geq b\}$  and  $h_2(\mathbf{X}) = 1 - h_1(\mathbf{X})$ , for suitable  $a \in \mathbb{R}^n$ ,  $b \in \mathbb{R}$  such that  $\mathbb{E}[f] = \mathbb{E}[h]$ . Then,  $\text{Stab}_\rho(f) \leq \text{Stab}_\rho(h)$ .*

While Borell’s theorem deals with the case of  $k = 2$ , it is natural to consider the question of maximal noise stability for  $k > 2$ , stated as follows.

**Question 1.** [Maximum Noise Stability (MNS)] Given a positive integer  $k \geq 2$  and  $\alpha \in \Delta_k$ , what is the maximum noise stability of a function  $f : \mathbb{R}^n \rightarrow \Delta_k$  satisfying  $\mathbb{E}[f] = \alpha$ ?

Question 1 remains open even for  $k = 3$ . In the particular case where  $\alpha = (\frac{1}{k}, \dots, \frac{1}{k})$ , the *Standard Simplex Conjecture* posits that the maximum noise stability is achieved by a “standard simplex partition” (this is equivalent to the *Plurality is Stablest* conjecture) [39, 35]. Even in the special case when  $k = 3$  and  $\alpha = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ , the answer is still tantalizingly open.



■ **Figure 1** (a) Borell’s Theorem: Halfspaces are most noise stable (b) Standard Simplex Partition for  $k = 3$  conjectured to be most noise stable (also known as the “Peace Sign Conjecture”).

In fact, a surprising result of [32] shows that when the  $\alpha_i$ ’s are not all equal, the standard simplex partition (and any variant thereof) *does not* achieve the maximum noise stability. This indicates that the case  $k \geq 3$  is fundamentally different than the case of  $k = 2$ . On the positive side, if we consider  $0 < \rho < \rho_0(k, n)$  (for some  $\rho_0(k, n)$  that goes to 0 for large  $n$ ), then the Standard Simplex Conjecture has been shown to hold [31]. However, this result is not applicable in the setting where  $\rho$  is fixed and  $n \rightarrow \infty$ .

The fact that we do not understand optimal partitions for  $k \geq 3$ , led De, Mossel & Neeman [19] to ask whether the optimal partition is realized in any finite dimension. More formally:

**Question 2.** Given  $k \geq 2$ ,  $\rho \in (0, 1)$ , and  $\alpha \in \Delta_k$ , let  $S_n(\alpha)$  be the optimal noise stability of a function  $f : \mathbb{R}^n \rightarrow \Delta_k$  subject to  $\mathbb{E}[f] = \alpha$ . Is there an  $n_0$  such that  $S_n(\alpha) = S_{n_0}(\alpha)$  for all  $n \geq n_0$ ?

Even Question 2 is open as of now! In this light, De, Mossel & Neeman [19] ask whether one can obtain an *explicitly computable*  $n_0 = n_0(k, \rho, \varepsilon)$  such that  $S_{n_0}(\alpha) \geq S_n(\alpha) - \varepsilon$  for all  $n \in \mathbb{N}$  (in other words, there exists a function  $f : \mathbb{R}^{n_0} \rightarrow \Delta_k$  that comes  $\varepsilon$ -close to achieving the optimal noise stability). Note that the challenge is really about  $n_0$  being *explicit*, since some  $n_0(k, \rho, \varepsilon)$  always exists, as  $S_n(\alpha)$  is a converging sequence as  $n \rightarrow \infty$ .

A natural approach to proving such an explicit bound is the idea of *dimension reduction*. Basically, it suffices to obtain an  $n_0 = n_0(k, \rho, \varepsilon)$  such that for any  $n$  and any given function  $f : \mathbb{R}^n \rightarrow \Delta_k$ , there exists a function  $\tilde{f} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  with  $\mathbb{E}[\tilde{f}] = \mathbb{E}[f]$  and  $\text{Stab}_\rho(\tilde{f}) \geq \text{Stab}_\rho(f) - \varepsilon$ . Instantiating  $f$  with an optimal (or near-optimal) partition in  $\mathbb{R}^n$ , for arbitrarily large  $n$ , then gives an  $\varepsilon$ -optimal partition  $\tilde{f}$  in  $\mathbb{R}^{n_0}$ .

Indeed, De, Mossel and Neeman follow such an approach and obtain an *explicitly computable* bound on  $n_0$ . To do so, they use and build on the theory of *eigenregular polynomials* that were previously studied in [21], which in turn uses other tools such as Malliavin calculus.

In this work, we introduce fundamentally different, but more elementary techniques (elaborated on shortly), thereby significantly improving the bound in [19]. In particular, we show the following.

► **Theorem 1** (Dimension Bound on Approximately Optimal Noise-Stable Function). *Given parameters  $k \geq 2$ ,  $\rho \in [0, 1]$  and  $\varepsilon > 0$ , there exists an explicitly computable  $n_0 = n_0(k, \rho, \varepsilon)$*

such that the following holds:

For any  $n \in \mathbb{N}$  and  $f : \mathbb{R}^n \rightarrow \Delta_k$ , there exists  $\tilde{f} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  such that,

1.  $\|\mathbb{E}[f] - \mathbb{E}[\tilde{f}]\|_1 \leq \varepsilon$ .
2.  $\text{Stab}_\rho(\tilde{f}) \geq \text{Stab}_\rho(f) - \varepsilon$ .

In particular, the explicit choice of  $n_0$  can be upper bounded by  $\exp\left(\text{poly}\left(k, \frac{1}{1-\rho}, \frac{1}{\varepsilon}\right)\right)$ .

### Remarks

- (i) In contrast to our theorem, the bound on  $n_0$  in [19] has an Ackermann-type growth.
- (ii) It is a slight technicality that we get  $\|\mathbb{E}[f] - \mathbb{E}[\tilde{f}]\|_1 \leq \varepsilon$  instead of  $\mathbb{E}[f] = \mathbb{E}[\tilde{f}]$  as was required. However, it is possible to slightly modify  $\tilde{f}$  to make  $\mathbb{E}[f] = \mathbb{E}[\tilde{f}]$ , if we allow  $n_0$  to depend on  $\alpha = \mathbb{E}[f]$  (which is the case in Question 2).

Theorem 1 has an immediate application to showing that approximately most-stable voting schemes (among all low-influential voting schemes) can be computed efficiently. We refer the reader to [19] for the details of this application. In order to prove Theorem 1, we in fact turn to the more general setting of *non-interactive simulation*.

## 1.2 Non-Interactive Simulation from Correlated Gaussian Sources

Consider a more general setting where instead of a single function  $f$ , we have two players, Alice and Bob, with corresponding functions  $A : \mathbb{R}^n \rightarrow \Delta_k$  and  $B : \mathbb{R}^n \rightarrow \Delta_k$ . They apply  $A$  and  $B$  on the sequence of random variables  $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)$  and  $\mathbf{Y} = (\mathbf{Y}_1, \dots, \mathbf{Y}_n)$  respectively, where  $(\mathbf{X}, \mathbf{Y}) \sim \mathcal{G}_\rho^{\otimes n}$ , which is the distribution of  $\rho$ -correlated Gaussians in  $n$  dimensions,

i.e. each coordinate  $(\mathbf{X}_i, \mathbf{Y}_i)$  is independently sampled from  $\mathcal{G}_\rho := \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)$ . The goal is to choose  $A$  and  $B$  such that  $\mathbb{E}[A] = \mathbb{E}[B] = \alpha$ , which is a pre-specified vector in  $\Delta_k$ , while maximizing  $\mathbb{E}_{(\mathbf{X}, \mathbf{Y})}[\langle A(\mathbf{X}), B(\mathbf{Y}) \rangle]$ . Note that, this quantity is same as  $\mathbb{E}_{\mathbf{X} \sim \gamma_n}[\langle A(\mathbf{X}), U_\rho B(\mathbf{X}) \rangle]$ , and hence in the restricted setting of  $A = B = f$  this quantity is exactly the noise stability of  $f$ .

We can interpret the above as: Alice observes  $\mathbf{X}$  and outputs  $i \in [k]$  with probability  $A_i(\mathbf{X})$ , similarly Bob observes  $\mathbf{Y}$  and outputs  $j \in [k]$  with probability  $B_j(\mathbf{Y})$ . In this sense, Alice and Bob wish to maximize their ‘‘agreement probability’’, i.e., their probability of outputting the same symbol. The dimension reduction mentioned in Theorem 1 generalized to this setup would require obtaining an  $n_0(k, \rho, \varepsilon)$  and a dimension reduction of  $A$  and  $B$  that approximately preserves the marginals and does not decrease the agreement probability by more than  $\varepsilon$ .

However, in this language, it is more natural to ask for a much stronger dimension reduction that preserves the entire joint distribution of symbols that Alice and Bob output, up to  $\varepsilon$  in total variation distance. We denote the joint distribution of Alice and Bob’s outputs as  $(A(\mathbf{X}), B(\mathbf{Y}))_{\mathcal{G}_\rho^{\otimes n}}$ , which is the distribution over  $(i, j) \in [k] \times [k]$  given as  $\Pr[\text{Alice outputs } i \text{ and Bob outputs } j] = \mathbb{E}_{(\mathbf{X}, \mathbf{Y})}[A_i(\mathbf{X})B_j(\mathbf{Y})]$ . In the case of  $k = 2$ , such a dimension reduction follows from (a more general version of) Borell’s theorem with in fact  $n_0 = 1!$  Our main result is indeed such a dimension reduction for all  $k \geq 2$ .

► **Theorem 2 (NIS from correlated Gaussian source).** *Given parameters  $k \geq 2$ ,  $\rho \in (0, 1)$  and  $\varepsilon > 0$ , there exists an explicitly computable  $n_0 = n_0(k, \rho, \varepsilon)$  such that the following holds: For any  $n$  and  $A : \mathbb{R}^n \rightarrow \Delta_k$  and  $B : \mathbb{R}^n \rightarrow \Delta_k$ , there exist  $\tilde{A} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  and  $\tilde{B} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  such that,*

$$d_{\text{TV}}\left((A(\mathbf{X}), B(\mathbf{Y}))_{\mathcal{G}_\rho^{\otimes n}}, (\tilde{A}(\mathbf{a}), \tilde{B}(\mathbf{b}))_{\mathcal{G}_\rho^{\otimes n_0}}\right) \leq \varepsilon.$$

In particular, the explicit choice of  $n_0$  is upper bounded as  $\exp\left(\text{poly}\left(k, \frac{1}{1-\rho}, \frac{1}{\varepsilon}\right)\right)$ .

The transformation satisfies a stronger property that there exists an “oblivious” randomized transformation (with a shared random seed) to go from  $A$  to  $\tilde{A}$  and from  $B$  to  $\tilde{B}$ , which works with probability at least  $1 - \varepsilon$ . Since the same transformation is applied on  $A$  and  $B$  with the same random seed, if  $A = B$ , then  $\tilde{A} = \tilde{B}$  as well.

Theorem 1 follows immediately from Theorem 2, by simply setting  $A = B = f$  to obtain  $\tilde{f} = \tilde{A} = \tilde{B}$ . In fact, following up on [19], De, Mossel & Neeman were able to extend their techniques to prove Theorem 2 [20] (again with Ackerman-type bounds on  $n_0$ ). To do so, they build on the tools developed in [19] along with a new smoothing argument inspired by boosting procedures in learning theory and potential function arguments in complexity theory and additive combinatorics. As we shall present shortly, our approach gets directly to Theorem 2 in a much more elementary fashion.

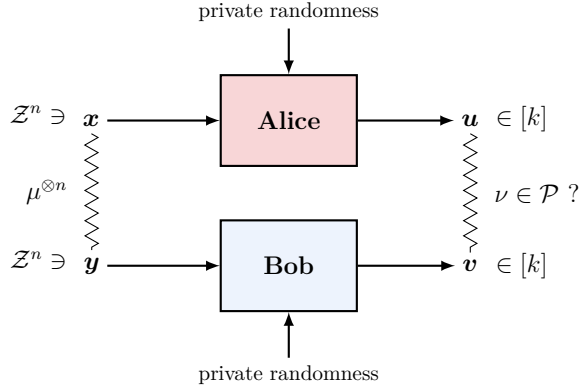
### 1.3 Extension: Non-Interactive Simulation from General Discrete Sources

The *Non-Interactive Simulation of Joint Distributions* is quite well studied in Information Theory and more recently in Theoretical Computer Science. Two players, Alice and Bob, observe the sequences of random variables  $(\mathbf{x}_1, \dots, \mathbf{x}_n)$  and  $(\mathbf{y}_1, \dots, \mathbf{y}_n)$  respectively, where each pair  $(\mathbf{x}_i, \mathbf{y}_i)$  is independently drawn from a known *source* distribution  $\mu$ . The fundamental question here is to understand which other *target* joint distributions  $\nu$  can Alice and Bob simulate, without communicating with each other? How many samples from  $\mu$  are needed to do so, or in other words, what is the *simulation rate*?

The history of this problem goes back to the classical works of Gács and Körner [24] and Wyner [56]. Specifically, consider the distribution Eq over  $\{0, 1\} \times \{0, 1\}$  where both marginals are  $\text{Ber}(1/2)$  and the bits are identical with probability 1. Gács and Körner studied the special case of this problem corresponding to the target distribution  $\nu = \text{Eq}$ . They characterized the simulation rate in this case, showing that it is equal to what is now known as the *Gács-Körner common information* of  $\mu$ . On the other hand, Wyner studied the special case corresponding to the source distribution  $\mu = \text{Eq}$ . He characterized the simulation rate in this case, showing that it is equal to what is now known as *Wyner common information* of  $\nu$ . Another particularly important work was by Witsenhausen [54] who studied the case where the target distribution  $\nu = \mathcal{G}_\rho$ . In this case, he showed that the largest correlation  $\rho$  that can be simulated is exactly the well-known “maximal correlation coefficient”<sup>4</sup>  $\rho(\mu)$  which was first introduced by Hirschfeld [33] and Gebelein [25] and then studied by Rényi [52]. Witsenhausen also considered the case where the target distribution  $\nu = \text{DSBS}_\rho$  is a “doubly symmetric binary source”, which is a pair of  $\rho$ -correlated bits (i.e., a pair of  $\pm 1$  random variables with correlation  $\rho$ ), and gave an approach to simulate correlated bits by first simulating  $\mathcal{G}_\rho$  starting with samples from  $\mu$ , and then applying half-space functions to get outputs in  $\{\pm 1\}$ . Starting with  $\mu$ , such a approach simulates  $\text{DSBS}_{\rho'}$  where  $\rho' = 1 - \frac{2 \arccos \rho(\mu)}{\pi}$ . Indeed, this calculation is identical to one that arises in the rounding technique employed in Goemans-Williamson’s approximation algorithm [30] for MAXCUT 20 years later!

We will consider the modern formulation of the NIS question as defined in [37]. This formulation ignores the simulation rate, and only focuses on whether simulation is even possible or not, given infinitely many samples from  $\mu$  – that is, whether the simulation rate is non-zero or not.

<sup>4</sup> We skip this definition as it is not central to our paper. The definition can be found in e.g. [29].



■ **Figure 2** Non-Interactive Simulation, e.g., as studied in [37]

► **Definition 3** (Non-interactive Simulation of Joint Distributions [37]). Let  $(\mathcal{Z} \times \mathcal{Z}, \mu)$  and  $([k] \times [k], \nu)$  be two joint probability spaces. The distribution  $\nu$  can be *non-interactively simulated* from distribution  $\mu$  if there exists a sequence of functions  $\{A^{(n)} : \mathcal{Z}^n \rightarrow \Delta_k\}_{n \in \mathbb{N}}$  and  $\{B^{(n)} : \mathcal{Z}^n \rightarrow \Delta_k\}_{n \in \mathbb{N}}$  such that the joint distribution  $\nu_n = (A^{(n)}(\mathbf{x}), B^{(n)}(\mathbf{y}))_{\mu^{\otimes n}}$  over  $[k] \times [k]$  is such that  $\lim_{n \rightarrow \infty} d_{\text{TV}}(\nu_n, \nu) = 0$ .

A central question that was left open following the work of Witsenhausen is: given distributions  $\mu$  and  $\nu$ , can  $\nu$  be non-interactively simulated from  $\mu$ ? Can this be even decided algorithmically? Even when  $\mu$  and  $\nu$  are extremely simple, e.g.,  $\mu$  is uniform on the triples  $\{(0, 0), (0, 1), (1, 0)\}$  and  $\nu$  is the doubly symmetric binary source DSBS<sub>0.49</sub>, it is open if  $\mu$  can simulate  $\nu$ ! This problem was formalized as a natural gap-problem in a work by a subset of the authors along with Sudan [29]. Here we state a slightly more general version.

► **Problem 4** (GAP-NIS $((\mathcal{Z} \times \mathcal{Z}, \mu), \mathcal{P}, k, \varepsilon)$ , cf. [29]). Given a joint probability space  $(\mathcal{Z} \times \mathcal{Z}, \mu)$ , a family of joint probability spaces  $\mathcal{P}$  supported over  $[k] \times [k]$ , and an error parameter  $\varepsilon > 0$ , distinguish between the following cases:

- (i) there exists  $n$  and  $A : \mathcal{Z}^n \rightarrow \Delta_k$  and  $B : \mathcal{Z}^n \rightarrow \Delta_k$ , s.t. the distribution  $\nu' = (A(\mathbf{x}), B(\mathbf{y}))_{\mu^{\otimes n}}$  satisfies  $d_{\text{TV}}(\nu', \nu) \leq \varepsilon$  for some  $\nu \in \mathcal{P}$ .
- (ii) for all  $n$  and all  $A : \mathcal{Z}^n \rightarrow \Delta_k$  and  $B : \mathcal{Z}^n \rightarrow \Delta_k$ , the distribution  $\nu' = (A(\mathbf{x}), B(\mathbf{y}))_{\mu^{\otimes n}}$  satisfies  $d_{\text{TV}}(\nu', \nu) > 2\varepsilon$  for all  $\nu \in \mathcal{P}$ .<sup>5</sup>

In prior work [29], it was shown that GAP-NIS for discrete distributions  $\mu$  and  $\nu$  is decidable, in the special case where  $k = 2$ . This was done by introducing a framework, which reduced the problem to understanding GAP-NIS for the special case where  $\mu = \mathcal{G}_\rho$ . Indeed, the reason why the case of  $k = 2$  was easier was precisely because Borell's theorem [10] gives an exact characterization of the distributions over  $[2] \times [2]$  that can be simulated from  $\mathcal{G}_\rho$ . The lack of understanding of the distributions over  $[k] \times [k]$  that can be simulated from  $\mathcal{G}_\rho$  was suggested in [29] as a barrier for extending their result to  $k > 2$ . With Theorem 2 in hand, it is possible to extend the framework in [29] of using a Regularity Lemma and Invariance principle, to yield the following theorem (as also done in [20], but with Ackerman-type bounds).

<sup>5</sup> the choice of constant 2 is arbitrary. Indeed, we could replace it by any constant greater than 1.



► **Theorem 5** (NIS from Discrete Sources). *Let  $(\mathcal{Z} \times \mathcal{Z}, \mu)$  be a joint probability space. Given parameters  $k \geq 2$  and  $\varepsilon > 0$ , there exists an explicitly computable  $n_0 = n_0(k, \mu, \varepsilon)$  such that the following holds:*

*For any  $n$  and  $A : \mathcal{Z}^n \rightarrow \Delta_k$  and  $B : \mathcal{Z}^n \rightarrow \Delta_k$ , there exist  $\tilde{A} : \mathcal{Z}^{n_0} \rightarrow \Delta_k$  and  $\tilde{B} : \mathcal{Z}^{n_0} \rightarrow \Delta_k$  such that,*

$$d_{\text{TV}} \left( (A(\mathbf{x}), B(\mathbf{y}))_{\mu^{\otimes n}}, (\tilde{A}(\mathbf{a}), \tilde{B}(\mathbf{b}))_{\mu^{\otimes n_0}} \right) \leq \varepsilon.$$

*In particular, the explicit choice of  $n_0$  is upper bounded as  $\exp \left( \text{poly} \left( k, \frac{1}{\varepsilon}, \frac{1}{1-\rho}, \log \left( \frac{1}{\alpha} \right) \right) \right)$ , where  $\alpha = \alpha(\mu)$  is the smallest atom in  $\mu$  and  $\rho = \rho(\mu)$  is the maximal correlation coefficient of  $\mu$ .*

The above theorem immediately suggests a brute force algorithm to decide  $\text{GAP-NIS}((\mathcal{Z} \times \mathcal{Z}, \mu), \mathcal{P}, k, \varepsilon)$ . We do not provide details of the proof of the above theorem in this extended abstract. The interested reader is referred to the full version of this paper [27] (available online) for details.

## 1.4 Dimension Reduction for Polynomials over Gaussian Space

We now describe the main technique of “*dimension reduction for low-degree polynomials*” that we introduce in this work, which could be of independent interest. We highlight that this technique is the main contribution of this paper.

Let’s start with Theorem 2, and explain the main ideas behind its proof. We are given two vector-valued functions  $A : \mathbb{R}^n \rightarrow \Delta_k$  and  $B : \mathbb{R}^n \rightarrow \Delta_k$ . We wish to reduce the dimension  $n$  of the Gaussian space on which  $A$  and  $B$  act while preserving the joint distribution  $(A(\mathbf{X}), B(\mathbf{Y}))_{\mathcal{G}_\rho^{\otimes n}}$  over  $[k] \times [k]$ . Recall that  $\mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{G}_\rho^{\otimes n}} [A_i(\mathbf{X}) \cdot B_j(\mathbf{Y})]$  is the probability of the event [*Alice outputs  $i$  and Bob outputs  $j$* ]. For succinctness, we write this expectation as  $\langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}}$ . In order to approximately preserve the joint distribution  $(A(\mathbf{X}), B(\mathbf{Y}))_{\mathcal{G}_\rho^{\otimes n}}$ , it suffices to approximately preserve  $\langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}}$  for each  $(i, j) \in [k] \times [k]$  upto an additive  $\varepsilon/k^2$ . Thus, to prove Theorem 2, we wish to find an explicit constant  $n_0 = n_0(\rho, k, \varepsilon)$ , along with functions  $\tilde{A} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  and  $\tilde{B} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  such that

$$\left| \langle \tilde{A}_i, \tilde{B}_j \rangle_{\mathcal{G}_\rho^{\otimes n_0}} - \langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \frac{\varepsilon}{k^2}.$$

Achieving this directly is highly unclear, since a priori, we have no structural information about  $A$  and  $B$ ! To get around this, we show that it is possible to first apply a structural transformation on  $A$  and  $B$  to convert them to low-degree and multilinear polynomials (see subsection 2.2 for formal definitions). Such transformations are described in section 4. This however creates a new problem that the transformed  $A$  and  $B$  no longer map to  $\Delta_k$ . Nevertheless, we will show that after the said transformation, we still have that the outputs of  $A$  and  $B$  are close to  $\Delta_k$  in expected  $\ell_2^2$  distance, that is,  $\text{dist}(A, \Delta_k) := (\mathbb{E}_{\mathbf{X}} \|\mathcal{R}(A(\mathbf{X})) - A(\mathbf{X})\|_2^2)^{1/2}$  is small (where  $\mathcal{R} : \mathbb{R}^k \rightarrow \Delta_k$  denotes the *rounding operator* that maps any  $v \in \mathbb{R}^k$  to its closest point in  $\Delta_k$ ). This will ensure that rounding the outputs of  $A$  and  $B$  to  $\Delta_k$  will approximately preserve the correlations  $\langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}}$ .

We are now able to revise our objective as follows: Given two (vector-valued) degree- $d$  polynomials  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$ , does there exist an explicit function  $n_0 = n_0(k, d, \delta)$ , along with polynomials  $\tilde{A} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^k$  and  $\tilde{B} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^k$  that  $\delta$ -approximately preserve (i) the correlation  $\langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}}$  for all  $(i, j) \in [k] \times [k]$  and (ii) closeness of the outputs of  $A$  and  $B$  to  $\Delta_k$ , that is,  $\text{dist}(A, \Delta_k)$  and  $\text{dist}(B, \Delta_k)$ ?

We introduce a very simple and natural dimension-reduction procedure for low-degree multilinear polynomials over Gaussian space. Specifically, for  $M$  that is a randomly sampled  $n \times n_0$  matrix with i.i.d. standard Gaussian entries, we set

$$\tilde{A}(a) := A \left( \frac{Ma}{\|a\|_2} \right) \quad \text{and} \quad \tilde{B}(b) := B \left( \frac{Mb}{\|b\|_2} \right) \quad \text{for } a, b \in \mathbb{R}^{n_0} \setminus \{0\}. \quad (1)$$

We leave  $\tilde{A}$  and  $\tilde{B}$  undefined on  $0 \in \mathbb{R}^{n_0}$ . This is inconsequential as  $\{0\}$  is a measure zero set under  $\gamma_n$ . Our main dimension-reduction theorem for polynomials is stated as follows.

► **Theorem 6** (Dimension Reduction Over Gaussian Space). *Given parameters  $k \geq 2$ ,  $d \in \mathbb{Z}_{\geq 0}$ ,  $\rho \in (0, 1)$  and  $\delta > 0$ , there exists an explicitly computable  $n_0 = n_0(d, k, \delta)$  such that the following holds:*

*Let  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$  be degree- $d$  multilinear polynomials. Additionally, suppose that  $\text{dist}(A, \Delta_k), \text{dist}(B, \Delta_k) \leq \delta$ . Consider the functions  $\tilde{A} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^k$  and  $\tilde{B} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^k$  as defined in Equation 1. With probability at least  $1 - O(\delta)$  over the choice of  $M \sim \gamma_1^{\otimes(n \times n_0)}$ , the following holds:*

1. For every  $i, j \in [k]$  :  $\left| \langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}} - \langle \tilde{A}_i, \tilde{B}_j \rangle_{\mathcal{G}_\rho^{\otimes n_0}} \right| \leq \delta$ .
2.  $\text{dist}(\tilde{A}, \Delta_k) \leq \sqrt{\delta}$  and  $\text{dist}(\tilde{B}, \Delta_k) \leq \sqrt{\delta}$ .

*In particular, the explicit choice of  $n_0$  is upper bounded as  $\exp(\text{poly}(d, \log k, \log(\frac{1}{\delta})))$ .*

It is clear from the construction of  $\tilde{A}$  and  $\tilde{B}$  that this theorem is giving us an “oblivious” randomized transformation, as also remarked in Theorem 2. The proof of Theorem 6 is obtained by combining Theorem 8 and Proposition 9 in section 3.

### Proof outline and analogy with the Johnson-Lindenstrauss lemma.

We will now highlight a few parallels between our proof of Theorem 6 and the proof of the Johnson-Lindenstrauss (JL) lemma (cf. [36, 18]), which has been extremely influential in computer science with numerous applications including compressed sensing, manifold learning, unsupervised learning and graph embedding.

Suppose that we have two unit vectors  $u, v \in \mathbb{R}^n$ . We wish to obtain a randomized transformation  $\Psi_{\mathbf{s}} : \mathbb{R}^n \rightarrow \mathbb{R}^{n_0}$  (for some random seed  $\mathbf{s}$ ) that approximately preserves the inner product, that is,  $\langle \Psi_{\mathbf{s}}(u), \Psi_{\mathbf{s}}(v) \rangle \approx_\delta \langle u, v \rangle$  holds with probability  $1 - \delta$ , over the randomness of seed  $\mathbf{s}$ ; note that here  $\langle \cdot, \cdot \rangle$  denotes the inner product over  $\mathbb{R}^n$  and  $\mathbb{R}^{n_0}$  respectively. Indeed, there is such a transformation, namely,  $\Psi_M(u) = \frac{M \cdot u}{\sqrt{n_0}}$  where  $M \sim \gamma_1^{\otimes n_0 \times n}$ . Let  $F(M) = \langle \Psi_M(u), \Psi_M(v) \rangle$ . Such a transformation satisfies,

$$\mathbb{E}_M[F(M)] = \langle u, v \rangle \quad \text{and} \quad \text{Var}_M(F(M)) = \frac{\langle u, v \rangle^2 + \|u\|_2^2 \|v\|_2^2}{n_0} \leq \frac{2}{n_0},$$

where we use that  $u$  and  $v$  are unit vectors. Thus, if we choose  $n_0 = 2/\delta^3$ , then we can make the variance smaller than  $\delta^3$ . Thereby, using Chebyshev’s inequality, we get that with probability at least  $1 - \delta$ , the inner product  $\langle u, v \rangle$  is preserved, that is,  $|\langle \Psi_M(u), \Psi_M(v) \rangle - \langle u, v \rangle| \leq \delta$ . Thus, we have a *oblivious* randomized dimension reduction that reduces the dimension of any pair of unit vectors to  $O(1/\delta^3)$ , independent of  $n$ . Note that, instead of using Chebyshev’s inequality, we could use a much sharper concentration bound to show that  $n_0 = O(1/\varepsilon^2 \log(1/\delta))$  suffices to preserve the inner product up to an additive  $\varepsilon$ , with probability  $1 - \delta$ . However, we described the Chebyshev’s inequality version as this is similar to our proof of Theorem 6.

The problem we are facing, although morally similar, is technically entirely different. For simplicity, let's first consider the task of reducing the dimension of the domain of a *single* pair of polynomials  $A : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}$ . And for the moment, consider the transformation such that  $\Psi_M A : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$  is given by  $A(\mathbf{M}a/\sqrt{n_0})$ . Similarly,  $\Psi_M B(b) = B(\mathbf{M}b/\sqrt{n_0})$ . Our proof of Theorem 6 proceeds along similar lines as the above proof of the JL Lemma, that is, by considering  $F(\mathbf{M}) = \langle \Psi_M A, \Psi_M B \rangle_{\mathcal{G}_\rho^{\otimes n_0}}$ , and proving bounds on  $\mathbb{E}_M[F(\mathbf{M})]$  and  $\text{Var}(F(\mathbf{M}))$ . This turns out to be quite delicate! Unlike the JL case, we don't even have  $\mathbb{E}_M[F(\mathbf{M})] = \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}}$ . What we do show however is that,

$$\left| \mathbb{E}_M[F(\mathbf{M})] - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq o_{n_0}(1) \quad \text{and} \quad \text{Var}_M(F(\mathbf{M})) \leq o_{n_0}(1),$$

that is, both are converging to 0 at an explicit rate determined by  $n_0$  (with some dependence on the degree  $d$  of  $A$  and  $B$ ). Interestingly however, in the case of  $d = 1$ , it turns out that  $F(\mathbf{M})$  is in fact an unbiased estimator of  $\langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}}$ . Indeed, this is not a coincidence! We leave it to the interested reader to figure out that in the case of  $d = 1$ , our transformation is in fact identical to the above described JL transformation on the  $n$ -dimensional space of Hermite coefficients of  $A$  and  $B$ .

Our actual transformation is slightly different, namely  $\Psi_M A(a) = A(\mathbf{M}a/\|\mathbf{a}\|_2)$ . This is to ensure item 2, about preserving the closeness of the output of  $A$  to  $\Delta_k$ . The proof gets a little more technical due to this change, but is intuitively similar to the above transformation since  $\|\mathbf{a}\|_2$  is tightly concentrated around  $\sqrt{n_0}$ . It is important to note that item 2 is quite critical to the entire approach. If it were not for this restriction, item 1 is very easy to satisfy on its own by other more direct dimension reduction operations on the Hermite coefficients.

The mean and variance bounds on  $F(\mathbf{M})$  are presented as Lemma 10. This is the most technical part of this work, but we stress that the main ideas are conceptually simple and elementary (for the most part). We provide a brief sketch of the proof in subsection 3.1 that illustrates all the main ideas in under a page, and defer all details to Appendix A. To prove these mean and variance bounds, we first analyze the case when  $A$  and  $B$  are multi-linear monomials (subsection A.2) and then combine these monomial calculations to obtain bounds for general multilinear polynomials (subsection A.3).

### 1.5 Comparisons with recent works of De, Mossel & Neeman

Our main theorems (Theorems 1, 2, 5) significantly improve the bounds in the versions proved by De, Mossel & Neeman [19, 20]. Our work was inspired by [19, 20] through several high-level ideas, such as the use of the transformation to low-degree and multilinear polynomials (although these transformations are technically different in our case). However, it seems that the key insight into “*why dimension reduction is possible*” provided by the works of De Mossel & Neeman and the current work are fundamentally different.

The key insight for dimension reduction in the work of De, Mossel & Neeman is (quoting [19]): “*the fact that a collection of homogeneous polynomials can be replaced by polynomials in bounded dimensions is a tensor analogue of the fact that for any  $k$  vectors in  $\mathbb{R}^n$ , there exist  $k$  vectors in  $\mathbb{R}^k$  with the same matrix of inner products*”. By contrast, the main intuition in our work is an “oblivious” dimension reduction technique, very similar to the Johnson-Lindenstrauss Lemma, as described in subsection 1.4.

Also, we point out a minor difference in our versions of Theorem 1. In [19] the function  $\tilde{f}$  maps to  $[k]$ , while in our theorem  $\tilde{f}$  maps to  $\Delta_k$ . Interestingly however, this is not a major difference and it follows from a thresholding lemma in [19, Lemma 15 & 16] that any such  $\tilde{f}$  can be modified to have range  $[k]$ , while preserving  $\mathbb{E}[\tilde{f}]$  without decreasing the noise stability.

## 1.6 Other Related Work and Future Directions

### Information Theory

Several previous works in information theory and theoretical computer science study “non-interactive simulation” type of questions. For instance, the non-interactive simulation of joint distributions question studied in this work is a generalization of the “non-interactive correlation distillation” problem<sup>6</sup> which was studied by [43, 45]. Moreover, recent works in the information theory community [37, 8] derive analytical tools (based on hypercontractivity and the so-called *strong data processing constant*) to prove impossibility results for NIS. While these results provide stronger bounds for some sources, they are not tight in general. Finally, the “non-interactive agreement distillation” problem studied by [9] can also be viewed as a particular case of the NIS setup.

### Randomness in Computation

As discussed in [29], one motivation for studying NIS problems stems from the study of the role of randomness in distributed computing. Specifically, recent works in cryptography [2, 3, 11, 17, 41, 51], quantum computing [47, 16, 23] and communication complexity [6, 15, 28, 26] study how the ability to solve various computational tasks gets affected by weakening the source of shared randomness. In this context, it is very natural to ask how well can a source of randomness be transformed into another (more structured) one, which is precisely the setup of non-interactive simulation.

The classic Newman’s theorem [46] tells us that any communication protocol with  $n$ -bit inputs and 0-1 outputs can be simulated with only  $O(\log n)$  bits of randomness. On the other hand, if we consider the setting where Alice and Bob run a communication protocol with correlated randomness, such as those defined in [6, 15], then reducing the randomness requirement of such protocols is not clear. Theorem 5 implies randomness reduction for zero-communication or even simultaneous message protocols, and hence can be seen as a first step towards understanding the randomness requirements of arbitrary (one or two way) communication protocols with access to correlated randomness.

### Tensor Power problems

Another motivation comes from the fact that NIS belongs to the class of *tensor power* problems, which have been very challenging to analyze. In such questions, the goal is to understand how some combinatorial quantity behaves in terms of the dimensionality of the problem as the dimension tends to infinity. A famous instance of such problems is the *Shannon capacity of a graph* [53, 40] where the aim is to understand how the independence number of the power of a graph behaves in terms of the exponent. The question of showing the computability of the Shannon capacity remains open to this day [4]. Other examples of such open problems (which are more closely related to NIS) arise in the problems of *local state transformation of quantum entanglement* [7, 22], the problem of computing the *entangled value of a 2-prover 1-round game* (see for, e.g., [38] and also the open problems [1]). Another example is the problem of computing the *amortized value of parallel repetitions of a 2-prover 1-round game* [49, 34, 48, 50, 5]. While we don’t have computability results for the amortized value, there has been a recent work that tries to characterize it in terms of an information theoretic quantity [12]. Yet another example of a tensor-power problem is the

---

<sup>6</sup> which considered the problem of maximizing agreement on a single bit, in various multi-party settings.

task of computing the *amortized communication complexity of a communication problem*. Braverman-Rao [13] showed that this equals the information complexity of the communication problem, however the computability of information complexity was shown only recently [14].

We hope that the recent progress on the Non-Interactive Simulation problem would stimulate progress on these other notable tensor-power problems. A concrete question is whether the techniques used for NIS (regularity lemma, invariance principle, etc.) can be translated to any of the above mentioned setups.

## Deterministic Approximate Counting

We also point out that the notions of eigenregularity used in [19, 20] were originally introduced and used in [21] to give the only known fixed-polynomial time *deterministic* approximate counting algorithm for polynomial threshold functions (PTFs). Our randomized techniques don't seem directly applicable to the PTF counting problem, as the emphasis there is on being *deterministic*. However, it will be interesting if our techniques could yield some further insights into approximate counting problems and pseudorandomness in general.

## 1.7 Organization of the Paper

In section 2, we summarize some useful definitions and provide a simple lemma that will be useful later. In section 3, we state our main technique of dimension reduction for polynomials (Theorem 6) and provide a brief sketch of the proof, with most details deferred to Appendix A. In section 4, we describe the transformations to make functions low-degree and multilinear, with proofs deferred to Appendix B. Finally, in section 5, we prove Theorem 2 (which implies Theorem 1 as a corollary).

## 2 Preliminaries

### 2.1 Gaussian Probability Spaces

Throughout this paper, we deal with the  $n$ -dimensional Gaussian space, i.e.  $\mathbb{R}^n$  equipped with Gaussian measure  $\gamma_n$  given by the density function

$$\frac{d\gamma_n}{d\mathbf{X}} := \frac{1}{(2\pi)^{n/2}} \cdot \exp\left(-\frac{1}{2} \cdot \|\mathbf{X}\|_{\mathbb{R}^n}^2\right).$$

where  $\|\cdot\|_{\mathbb{R}^n}$  denotes the  $\ell_2$  norm of a vector. We use letters such as  $X, Y$  to denote points in  $\mathbb{R}^n$ , bold symbols such as  $\mathbf{X}, \mathbf{Y}$  to denote random variables, subscripts such as  $X_i$  or  $\mathbf{X}_i$  denote the  $i$ -th coordinate.

The  $\ell_2$ -norm of a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is defined as  $\|f\|_2 := \left[ \mathbb{E}_{\mathbf{X} \sim \gamma_n} f(\mathbf{X})^2 \right]^{1/2}$ . We use  $L^2(\mathbb{R}^n, \gamma_n)$  to denote the space of all  $\ell_2$ -integrable functions  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , i.e.  $\|f\|_2 < \infty$ . All functions we consider will be  $\ell_2$ -integrable. The inner product of  $f, g \in L^2(\mathbb{R}^n, \gamma_n)$  is defined as  $\langle f, g \rangle_{\gamma_n} := \mathbb{E}_{\mathbf{X} \sim \gamma_n} [f(\mathbf{X})g(\mathbf{X})]$ .

The joint distribution of  $\rho$ -correlated Gaussians is denoted as  $\mathcal{G}_\rho$ , which is a 2-dimensional Gaussian distribution  $(\mathbf{X}, \mathbf{Y})$ , where  $\mathbf{X}$  and  $\mathbf{Y}$  are marginally distributed according to  $\gamma_1$ , with  $\mathbb{E}[\mathbf{X}\mathbf{Y}] = \rho$ . For  $A, B \in L^2(\mathbb{R}^n, \gamma_n)$ , the *noisy correlation between A and B over  $\mathcal{G}_\rho^{\otimes n}$*  is defined as,

$$\langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} := \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{G}_\rho^{\otimes n}} [A(\mathbf{X}) \cdot B(\mathbf{Y})]$$

Finally, the total variation distance between two distributions  $\mu$  and  $\nu$  over domain  $\Omega$  is defined as,

$$d_{\text{TV}}(\mu, \nu) := \sup_{S \subseteq \Omega} |\mu(S) - \nu(S)|.$$

## 2.2 Hermite Analysis

The set of Hermite polynomials  $\{H_r : \mathbb{R} \rightarrow \mathbb{R} : r \in \mathbb{Z}_{\geq 0}\}$  form an orthonormal basis for functions in  $L^2(\mathbb{R}, \gamma_1)$  with respect to the inner product  $\langle \cdot, \cdot \rangle_{\gamma_1}$ . The  $r$ -th Hermite polynomial  $H_r : \mathbb{R} \rightarrow \mathbb{R}$  (for  $r \in \mathbb{Z}_{\geq 0}$ ) is defined as,

$$H_0(x) = 1; \quad H_1(x) = x; \quad H_r(x) = \frac{(-1)^r}{\sqrt{r!}} e^{x^2/2} \cdot \frac{d^r}{dx^r} e^{-x^2/2}.$$

Hermite polynomials can also be obtained via the generating function,  $e^{xt - \frac{t^2}{2}} = \sum_{r=0}^{\infty} \frac{H_r(x)}{\sqrt{r!}} t^r$ .

For any  $\sigma = (\sigma_1, \dots, \sigma_n) \in \mathbb{Z}_{\geq 0}^n$ , define  $H_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}$  as  $H_\sigma(\mathbf{X}) = \prod_{i=1}^n H_{\sigma_i}(\mathbf{X}_i)$ . It easily follows that the set  $\{H_\sigma : \sigma \in \mathbb{Z}_{\geq 0}^n\}$  forms an orthonormal basis for  $L^2(\mathbb{R}^n, \gamma_n)$ . Thus, every  $A \in L^2(\mathbb{R}^n, \gamma_n)$  has a *Hermite expansion* given by  $A(\mathbf{X}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}(\sigma) \cdot H_\sigma(\mathbf{X})$ , where the  $\hat{A}(\sigma)$ 's are the *Hermite coefficients* of  $A$  obtained as  $\hat{A}(\sigma) = \langle A, H_\sigma \rangle_{\gamma_n}$ . The degree of  $\sigma$  is defined as  $|\sigma| := \sum_{i \in [n]} \sigma_i$ , and the degree of  $A$  is the largest  $|\sigma|$  for which  $\hat{A}(\sigma) \neq 0$ . We say that  $A \in L^2(\mathbb{R}^n, \gamma_n)$  is *multilinear* if  $\hat{A}(\sigma)$  is non-zero only if  $\sigma_i \in \{0, 1\}$  for all  $i \in [n]$ .

We list several useful facts about Hermite coefficients:

- (1) Parseval's identity:  $\|A\|_2^2 = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}(\sigma)^2$  and  $\text{Var}(A) = \sum_{\mathbf{0} \neq \sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}(\sigma)^2$ .
- (2) Plancherel's identity:  $\langle A, A' \rangle_{\gamma_n} = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}(\sigma) \hat{A}'(\sigma)$ .
- (3) Ornstein-Uhlenbeck operator:  $U_\rho A(X) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \rho^{|\sigma|} \cdot \hat{A}(\sigma) \cdot H_\sigma(X)$ .
- (4) Noisy Correlation:  $\langle A, B \rangle_{\mathcal{G}_{\rho^n}} = \langle A, U_\rho B \rangle_{\gamma_n} = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \rho^{|\sigma|} \hat{A}(\sigma) \hat{B}(\sigma)$

For convenience,  $U_\rho(X)$  denotes the distribution  $(\rho X + \sqrt{1 - \rho^2} \mathbf{Z})$  where  $\mathbf{Z} \sim \gamma_n$ , for any  $X \in \mathbb{R}^n$ .

## 2.3 Vector-valued functions

For any function  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$ , we will interpret  $A$  as a vector of functions  $(A_1, \dots, A_k)$ , where  $A_i : \mathbb{R}^n \rightarrow \mathbb{R}$  is the  $i$ -th coordinate of the output of  $A$ . The definitions of Hermite analysis extend naturally to vector-valued functions as follows. For  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$ , the Hermite coefficient  $\hat{A}(\sigma)$  is  $(\hat{A}_1(\sigma), \dots, \hat{A}_k(\sigma)) \in \mathbb{R}^k$ . We can extend the definition of  $\ell_2$ -norm as  $\|A\|^2 := \mathbb{E}_{\mathbf{X} \sim \gamma_n} \|A(\mathbf{X})\|^2$  or equivalently  $\|A\|^2 = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \|\hat{A}(\sigma)\|^2$ . Also,  $\text{deg}(A) := \max_{i \in [k]} \text{deg}(A_i)$ . Again, all the vector-valued functions with domain  $\mathbb{R}^n$  that we consider will be such that the function in each coordinate is in  $L^2(\mathbb{R}^n, \gamma_n)$ .

For  $k \in \mathbb{N}$  and  $i \in [k]$ , let  $e_i$  be the unit vector along coordinate  $i$  in  $\mathbb{R}^k$ . The simplex  $\Delta_k$  is defined as the convex hull formed by  $\{e_i : i \in [k]\}$ . Equivalently,  $\Delta_k = \{v \in \mathbb{R}^k : \|v\|_1 = 1\}$  is the set of probability distributions over  $[k]$ . While we will consider vector-valued functions mapping to  $\mathbb{R}^k$ , we will be primarily interested in functions which map to  $\Delta_k$ . The rounding operator  $\mathcal{R}^{(k)} : \mathbb{R}^k \rightarrow \Delta_k$  maps any  $v \in \mathbb{R}^k$  to its closest point in  $\Delta_k$ . In particular, it is the identity map on  $\Delta_k$ . We will drop the superscript on  $\mathcal{R}$ , as  $k$  is fixed throughout this paper. Similar to our notation for vector-valued functions,  $\mathcal{R}_i$  denotes the  $i$ -th coordinate of  $\mathcal{R}$ . Thus, while the  $i$ -th coordinate of  $A$  is denoted by  $A_i$ , the  $i$ -th coordinate of  $\mathcal{R}(A)$  is denoted by  $\mathcal{R}_i(A)$ .

As mentioned already, an important relaxation in our work is to consider functions that do not map to  $\Delta_k$ , but instead map to  $\mathbb{R}^k$ . For such functions to be meaningful, we will require that the outputs are *usually close* to  $\Delta_k$ , in which case, we will be rounding them to the simplex  $\Delta_k$ . Towards this end, the following simple proposition will be very useful, which says that if we modify the strategies of Alice and Bob slightly (in  $\ell_2$ -distance), then the correlation between the strategies does not change significantly. The proof follows by a simple triangle inequality and the Cauchy-Schwarz inequality.

► **Proposition 7** (Close strategies, have similar correlations). *Let  $A, \tilde{A}, B, \tilde{B} \in L^2(\mathbb{R}^n, \gamma_n)$  such that  $\|A\|_2, \|\tilde{A}\|_2, \|B\|_2, \|\tilde{B}\|_2 \leq 1$ . If  $\|A - \tilde{A}\|_2 \leq \varepsilon$  and  $\|B - \tilde{B}\|_2 \leq \varepsilon$ , then it holds that,*

$$\left| \langle \tilde{A}, \tilde{B} \rangle_{\mathcal{G}_\rho^{\otimes n}} - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq 2\varepsilon.$$

### 3 Dimension Reduction for Low-Degree Multilinear Polynomials

In this section, we present our main technique of dimension reduction for low-degree multilinear polynomials over Gaussian space. Theorem 6 is obtained immediately as a combination of Theorem 8 and Proposition 9 stated below.

► **Theorem 8.** *Given  $d \in \mathbb{Z}_{>0}$ ,  $\rho \in [0, 1]$  and  $\delta > 0$ , there exists an explicitly computable  $n_0 = n_0(d, \delta)$ , such that the following holds:*

*Let  $A : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}$  be degree- $d$  multilinear polynomials, s.t.  $\|A\|_2, \|B\|_2 \leq 1$ . For  $\mathbf{M} \in \mathbb{R}^{n \times n_0}$  with entries i.i.d. sampled from  $\gamma_1$ , define the functions<sup>7</sup>  $A_{\mathbf{M}} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$  and  $B_{\mathbf{M}} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$  as*

$$A_{\mathbf{M}}(a) = A\left(\frac{\mathbf{M}a}{\|a\|_2}\right) \quad \text{and} \quad B_{\mathbf{M}}(b) = B\left(\frac{\mathbf{M}b}{\|b\|_2}\right) \quad \text{for } a, b \in \mathbb{R}^{n_0} \setminus \{0\}.$$

*Then, with probability at least  $1 - \delta$  (over the choice of  $\mathbf{M}$ ), it holds that,*

$$\left| \langle A_{\mathbf{M}}, B_{\mathbf{M}} \rangle_{\mathcal{G}_\rho^{\otimes n_0}} - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| < \delta.$$

*In particular, the explicit choice of  $n_0$  is upper bounded as  $\frac{d^{O(d)}}{\delta^4}$ .*

In other words, for a typical choice of  $\mathbf{M} \sim \gamma_1^{\otimes (n \times n_0)}$ , the correlation between  $A$  and  $B$  is approximately preserved if we replace  $(\mathbf{X}, \mathbf{Y}) \sim \mathcal{G}_\rho^{\otimes n}$  by  $(\mathbf{M}\mathbf{a}/\|\mathbf{a}\|_2, \mathbf{M}\mathbf{b}/\|\mathbf{b}\|_2)$ , where  $(\mathbf{a}, \mathbf{b}) \sim \mathcal{G}_\rho^{\otimes n_0}$ . Intuitively,  $\mathbf{M}$  can be thought of as a means to “stretch”  $n_0$  coordinates of  $\mathcal{G}_\rho$  into effectively  $n$  coordinates of  $\mathcal{G}_\rho$ , while “fooling” correlations between degree- $d$  multilinear polynomials.

Before we prove the above theorem, we prove a simple proposition that completely handles item 2 of Theorem 6 by showing that if this dimension reduction were applied to vector-valued functions whose outputs lie close to the simplex  $\Delta_k$ , then with high probability, even the dimension-reduced functions will have outputs close to the simplex. More formally,

► **Proposition 9.** *Let  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$ , such that  $\|\mathcal{R}(A) - A\|_2, \|\mathcal{R}(B) - B\|_2 \leq \delta$ . Then, with probability at least  $1 - 2\delta$  (over choice of  $\mathbf{M}$ ), it holds that,*

$$\|\mathcal{R}(A_{\mathbf{M}}) - A_{\mathbf{M}}\|_2 \leq \sqrt{\delta} \quad \text{and} \quad \|\mathcal{R}(B_{\mathbf{M}}) - B_{\mathbf{M}}\|_2 \leq \sqrt{\delta}.$$

<sup>7</sup>  $A_{\mathbf{M}}$  and  $B_{\mathbf{M}}$  can be defined arbitrarily on  $0 \in \mathbb{R}^{n_0}$ . This is inconsequential as  $\{0\}$  is a measure zero set under  $\gamma_n$ .

**Proof.** Observe that even for a fixed non-zero  $a \in \mathbb{R}^{n_0}$ , the distribution of  $\frac{\mathbf{M}a}{\|a\|_2}$  is identical to that of a standard  $n$ -variate Gaussian distribution  $\gamma_n$ . Thus, we immediately have that,

$$\begin{aligned} \mathbb{E}_{\mathbf{M}} \mathbb{E}_a \left\| \mathcal{R} \left( A \left( \frac{\mathbf{M}a}{\|a\|_2} \right) \right) - A \left( \frac{\mathbf{M}a}{\|a\|_2} \right) \right\|_2^2 &= \mathbb{E}_{\mathbf{X}} \left\| \mathcal{R}(A(\mathbf{X})) - A(\mathbf{X}) \right\|_2^2 \\ \text{Alternately, } \mathbb{E}_{\mathbf{M}} \left\| \mathcal{R}(A_{\mathbf{M}}) - A_{\mathbf{M}} \right\|_2^2 &= \left\| \mathcal{R}(A) - A \right\|_2^2 \leq \delta^2 \end{aligned}$$

Thus, by Markov's inequality,  $\left\| \mathcal{R}(A_{\mathbf{M}}) - A_{\mathbf{M}} \right\|_2 \leq \sqrt{\delta}$  holds with probability at least  $1 - \delta$ . We can similarly argue for  $B_{\mathbf{M}}$ , and a union bound completes the proof.  $\blacktriangleleft$

To prove Theorem 8, we primarily use the second moment method (i.e., Chebyshev's inequality). In particular, let  $F(\mathbf{M})$  be defined as,

$$F(\mathbf{M}) \stackrel{\text{def}}{=} \langle A_{\mathbf{M}}, B_{\mathbf{M}} \rangle_{\mathcal{G}_\rho^{\otimes n_0}}$$

The most technical part of this work is to show sufficiently good bounds on the mean and variance of  $F(\mathbf{M})$  for a random choice of  $\mathbf{M} \sim \gamma_1^{\otimes (n \times n_0)}$ , given by the following lemma.

**► Lemma 10.** (Mean & Variance Bound). *Given  $d$  and  $\delta$ , there exists an explicitly computable  $n_0 := n_0(d, \delta)$  such that for  $\mathbf{M} \sim \gamma_1^{\otimes (n \times n_0)}$ ,*

$$\begin{aligned} \left| \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| &\leq \delta && \text{(Mean bound)} \\ \text{Var}_{\mathbf{M}}(F(\mathbf{M})) &\leq \delta && \text{(Variance bound)} \end{aligned}$$

In particular, one may take  $n_0 = \frac{d^{O(d)}}{\delta^2}$ .

We provide a little sketch of the proof of Lemma 10 below, with the full details in Appendix A. Assuming Lemma 10, we can easily prove Theorem 8.

**Proof of Theorem 8.** We invoke Lemma 10 with parameters  $d$  and  $\delta^2/2$ , to get a choice of  $n_0 = \frac{d^{O(d)}}{\delta^4}$ . Using Chebyshev's inequality and the Variance bound in Lemma 10, we have that for any  $\eta > 0$ ,

$$\Pr_{\mathbf{M}} \left[ |F(\mathbf{M}) - \mathbb{E}_{\mathbf{M}} F(\mathbf{M})| > \eta \right] \leq \frac{\delta^2}{2\eta}.$$

Using the triangle inequality, and the Mean bound in Lemma 10, we get

$$\begin{aligned} &\Pr_{\mathbf{M}} \left[ \left| F(\mathbf{M}) - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| > \delta \right] \\ &\leq \Pr_{\mathbf{M}} \left[ \left| F(\mathbf{M}) - \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) \right| + \left| \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| > \delta \right] \\ &\leq \Pr_{\mathbf{M}} \left[ \left| F(\mathbf{M}) - \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) \right| > \delta - \delta^2 \right] \leq \delta. \end{aligned} \quad \blacktriangleleft$$

### 3.1 Proof Sketch of Lemma 10

While the proof of Lemma 10 is somewhat technical as a whole, the main driver of the entire lemma is a simple combinatorial fact that if we sample  $d$  times with replacement from a bag with  $n_0$  items, then the probability of not sampling distinct items is at most  $O(d^2/n_0) = o_{n_0}(1)$ . We briefly illustrate this idea at play by proving a *simpler* version of the mean bound. For this section, let's consider a different dimension reduction of setting  $A_{\mathbf{M}}$  and  $B_{\mathbf{M}}$  as,  $A_{\mathbf{M}}(a) = A(\mathbf{M}a/\sqrt{n_0})$  and  $B_{\mathbf{M}}(b) = B(\mathbf{M}b/\sqrt{n_0})$ , where  $\mathbf{M} \sim \gamma_1^{\otimes (n \times n_0)}$ .



Let  $\mathbf{m}_i \in \mathbb{R}^{n_0}$  denote the vector corresponding to the  $i$ -th row of  $\mathbf{M}$ . Consider the mean of  $F(\mathbf{M}) = \langle A_{\mathbf{M}}, B_{\mathbf{M}} \rangle_{\mathcal{G}_\rho^{\otimes n_0}}$ :

$$\begin{aligned} \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) &= \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} A\left(\frac{\mathbf{M}\mathbf{a}}{\sqrt{n_0}}\right) B\left(\frac{\mathbf{M}\mathbf{b}}{\sqrt{n_0}}\right) \\ &= \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \sum_{\sigma, \kappa} \frac{\hat{A}(\sigma)\hat{B}(\kappa)}{n_0^{(|\sigma|+|\kappa|)/2}} \cdot \prod_{i: \sigma_i=1} \langle \mathbf{m}_i, \mathbf{a} \rangle \cdot \prod_{j: \kappa_j=1} \langle \mathbf{m}_j, \mathbf{b} \rangle \end{aligned}$$

where, recall that  $A$  and  $B$  are multilinear, and so the relevant  $\sigma$  and  $\kappa$  are in  $\{0, 1\}^n$ , with  $|\sigma|, |\kappa| \leq d$ . Next, observe that  $\mathbb{E} \mathbf{m}_i \mathbf{m}_i^T = I_{n_0 \times n_0}$ , and hence we get that,

$$\mathbb{E}_{\mathbf{M}} \prod_{i: \sigma_i=1} \langle \mathbf{m}_i, \mathbf{a} \rangle \cdot \prod_{j: \kappa_j=1} \langle \mathbf{m}_j, \mathbf{b} \rangle = \begin{cases} \langle \mathbf{a}, \mathbf{b} \rangle^{|\sigma|} & \text{if } \sigma = \kappa \\ 0 & \text{if } \sigma \neq \kappa \end{cases}.$$

Finally, we observe that if we expand  $\langle \mathbf{a}, \mathbf{b} \rangle^d$  as  $\sum_{i_1, \dots, i_d \in [n_0]} \mathbf{a}_{i_1} \mathbf{b}_{i_1} \dots \mathbf{a}_{i_d} \mathbf{b}_{i_d}$ , then from the combinatorial fact above, except for a  $O(d^2) \cdot n_0^{d-1}$  out of total  $n_0^d$  terms, the indices  $i_1, \dots, i_d$  are all distinct. It is immediate to see that if all the  $i_j$ 's are distinct then  $\mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbf{a}_{i_1} \mathbf{b}_{i_1} \dots \mathbf{a}_{i_d} \mathbf{b}_{i_d} = \rho^d$ . Additionally, we show that if the  $i_j$ 's are not all distinct then  $|\mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbf{a}_{i_1} \mathbf{b}_{i_1} \dots \mathbf{a}_{i_d} \mathbf{b}_{i_d}| \leq d^{O(d)}$  (this follows from the fact that for the  $d$ -th moments of  $\gamma_1$  are at most  $d^{O(d)}$ ). Putting this together we get for any  $\sigma$  (with  $|\sigma| \leq d$ ) that,

$$\mathbb{E}_{\mathbf{a}, \mathbf{b}} \frac{\langle \mathbf{a}, \mathbf{b} \rangle^{|\sigma|}}{n_0^{|\sigma|}} = \rho^{|\sigma|} \pm \frac{d^{O(d)}}{n_0}$$

Putting everything together we get,

$$\mathbb{E}_{\mathbf{M}} F(\mathbf{M}) = \sum_{\sigma} \hat{A}(\sigma)\hat{B}(\sigma) \cdot \left( \rho^{|\sigma|} \pm \frac{d^{O(d)}}{n_0} \right) = \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \pm \sum_{\sigma} \hat{A}(\sigma)\hat{B}(\sigma) \cdot \frac{d^{O(d)}}{n_0}$$

And hence,

$$\left| \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \frac{d^{O(d)}}{n_0} \cdot \sum_{\sigma} \hat{A}(\sigma)\hat{B}(\sigma) \leq \frac{d^{O(d)}}{n_0} \cdot \|A\|_2 \cdot \|B\|_2 \leq \delta,$$

where we use the Cauchy-Schwarz inequality and that  $n_0 \geq d^{O(d)}/\delta$ . This completes a proof sketch of the mean bound in Lemma 10. Replacing  $\sqrt{n_0}$  by  $\|\mathbf{a}\|_2$  introduces a minor technicality, but still works because  $\|\mathbf{a}\|_2$  is tightly concentrated around  $\sqrt{n_0}$ . The variance bound is slightly more complicated with the use of a hypercontractive inequality instead of Cauchy-Schwarz. The full details of the proof are in Appendix A.

## 4 Transformation to Low-Degree Multilinear Polynomials

While Theorem 6 applies only for low-degree multilinear polynomials, we can extend it for all functions by using the following lemma that transforms  $k$ -dimensional  $\ell_2$ -integrable functions  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$  into low-degree multilinear polynomials while approximately preserving all correlations and also not deviating much from the simplex  $\Delta_k$  (although slightly increasing the number of variables).

► **Lemma 11 (Low-Degree Multilinear Transformation).** *Given parameters  $\rho \in [0, 1]$ ,  $\delta > 0$ ,  $k \in \mathbb{N}$ , there exists an explicit  $d = d(k, \rho, \delta)$  and  $t := t(k, d, \delta)$  such that the following holds: Let  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$ , s.t. for any  $i \in [k]$ , it holds that  $\text{Var}(A_i), \text{Var}(B_i) \leq 1$ . Then, there exist functions  $\tilde{A} : \mathbb{R}^{nt} \rightarrow \mathbb{R}^k$  and  $\tilde{B} : \mathbb{R}^{nt} \rightarrow \mathbb{R}^k$  such that the following statements hold.*

1.  $\tilde{A}$  and  $\tilde{B}$  are multilinear with degree at most  $d$ .
2. For any  $i \in [k]$ , it holds that  $\text{Var}(\tilde{A}_i) \leq \text{Var}(A_i) \leq 1$  and  $\text{Var}(\tilde{B}_i) \leq \text{Var}(B_i) \leq 1$ .
3.  $\|\mathcal{R}(\tilde{A}) - \tilde{A}\|_2 \leq \|\mathcal{R}(A) - A\|_2 + \delta$  and  $\|\mathcal{R}(\tilde{B}) - \tilde{B}\|_2 \leq \|\mathcal{R}(B) - B\|_2 + \delta$
4. For every  $i, j \in [k]$ ,

$$\left| \left\langle \tilde{A}_i, \tilde{B}_j \right\rangle_{\mathcal{G}_\rho^{\otimes nt}} - \langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \frac{\delta}{\sqrt{k}}$$

In particular, one may take  $d = O\left(\frac{\sqrt{k} \log^2(k/\delta)}{\delta(1-\rho)}\right)$  and  $t = O\left(\frac{kd^2}{\delta^2}\right)$ .

This lemma is itself proved in two stages. The first stage transforms general functions to low-degree polynomials by applying a small noise operator (making the functions have “decaying Hermite tails”) followed by truncation of the higher degree terms. The second stage transforms low-degree polynomials into multilinear ones, by replacing each variable by a normalized sum of new variables (making the functions have low mass on non-multilinear terms) followed by truncation of the non-multilinear terms.

These techniques are quite standard in literature. For the use of noise operator in the first stage see e.g. [44, 42]. For the substitution of variables in the second stage see e.g. [19]. However, since we are stating particular quantitative versions of the lemmas, we provide the proofs in Appendix B for completeness.

## 5 Non-Interactive Simulation from Correlated Gaussian Sources

In this section, we complete the proof of our main theorem regarding non-interactive simulation from correlated Gaussian sources, i.e. Theorem 2. Recall that it immediately implies Theorem 1 by setting  $A = B = f$  and obtaining  $\tilde{f} = \tilde{A} = \tilde{B}$ .

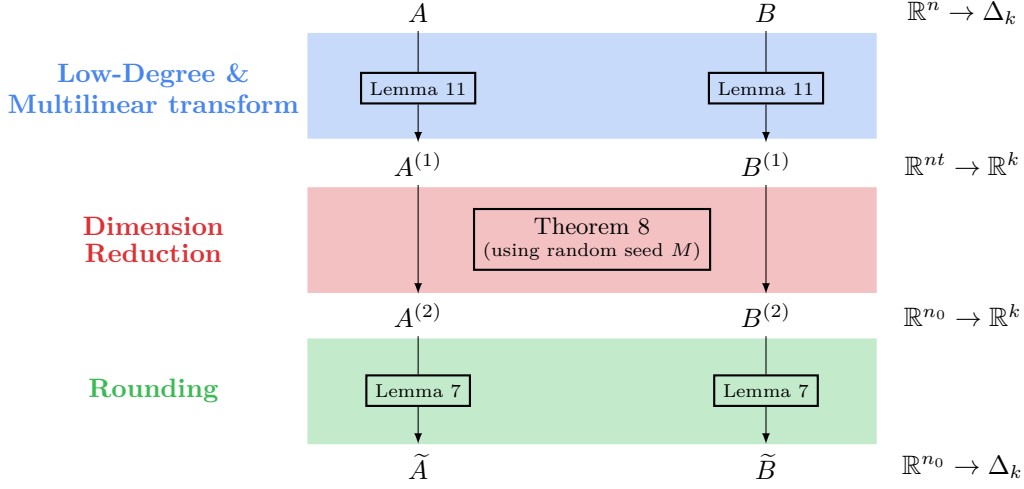
**Proof of Theorem 2.** Starting with functions  $A : \mathbb{R}^n \rightarrow \Delta_k$  and  $B : \mathbb{R}^n \rightarrow \Delta_k$ , we first apply Lemma 11 to transform  $A$  and  $B$  to low-degree and multilinear polynomials, and subsequently apply Theorem 8. Unfortunately after these transformations, the range is no longer restricted to  $\Delta_k$ . Nevertheless, we do have that these transformations ensure that the functions still output something “close” to the simplex  $\Delta_k$ . This allows us to apply the rounding operator and get the range as  $\Delta_k$  again (using Lemma 7). An overview of the transformations done is presented in Figure 3.

We thus transform  $A$  and  $B$  through each of the following steps. At each step, we approximately preserve the correlation  $\langle A_i, B_j \rangle$  for every  $i, j \in [k]$ . Additionally, in each step  $\|\mathcal{R}(A) - A\|_2$  and  $\|\mathcal{R}(B) - B\|_2$  doesn’t increase significantly (note that, to begin with, the range of  $A$  and  $B$  is  $\Delta_k$  and hence we start with  $\|\mathcal{R}(A) - A\|_2 = \|\mathcal{R}(B) - B\|_2 = 0$ ).

1. **Transformation to Low-Degree & Multilinear:** We apply Lemma 11 on  $A$  and  $B$  with parameter  $\delta$  (chosen later), setting  $d = d(\rho, k, \delta)$  and  $t = t(d, k, \delta)$  as required, to get degree- $d$  and multilinear  $A^{(1)} : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B^{(1)} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ . Moreover, we have that for every  $i, j \in [k]$ ,

$$\left| \left\langle A_i^{(1)}, B_j^{(1)} \right\rangle_{\mathcal{G}_\rho^{\otimes nt}} - \langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \delta \quad (2)$$

Additionally, we have  $\|\mathcal{R}(A^{(1)}) - A^{(1)}\|_2 \leq \|\mathcal{R}(A) - A\|_2 + \delta \leq \delta$  and similarly for  $B^{(1)}$ .



■ **Figure 3** Transformations for Non-interactive simulation from Correlated Gaussian Sources

2. **Dimension reduction:** We apply Theorem 8 with parameter  $\delta/k^2$ , setting  $n_0 = n_0(d, \rho, \delta/k^2)$  as required, on individual coordinates of  $A^{(1)}$  and  $B^{(1)}$  to obtain functions  $A^{(2)} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^k$  and  $B^{(2)} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^k$ . Taking a union bound, we have that with probability at least  $1 - \delta$ , it holds for every  $i, j \in [k]$  that,

$$\left| \left\langle A_i^{(2)}, B_j^{(2)} \right\rangle_{\mathcal{G}_\rho^{\otimes n_0}} - \left\langle A_i^{(1)}, B_j^{(1)} \right\rangle_{\mathcal{G}_\rho^{\otimes nt}} \right| \leq \delta \quad (3)$$

From Proposition 9, we have that with probability  $1 - 4\delta$ ,

$$\begin{aligned} \|\mathcal{R}(A^{(2)}) - A^{(2)}\|_2 &\leq \sqrt{\|\mathcal{R}(A^{(1)}) - A^{(1)}\|_2} \leq \sqrt{\delta} \\ \|\mathcal{R}(B^{(2)}) - B^{(2)}\|_2 &\leq \sqrt{\|\mathcal{R}(B^{(1)}) - B^{(1)}\|_2} \leq \sqrt{\delta} \end{aligned}$$

Note that this is the only randomized step in the entire transformation, and it succeeds in obtaining the above three statements with probability at least  $1 - 5\delta$ .

3. **Rounding to  $\Delta_k$ :** Finally, we set  $\tilde{A} = \mathcal{R}(A^{(2)})$  and  $\tilde{B} = \mathcal{R}(B^{(2)})$ . Thus, assuming the previous step succeeds, we have that  $\|\tilde{A}_i - A_i^{(2)}\|_2 \leq \sqrt{\delta}$  and  $\|\tilde{B}_j - B_j^{(2)}\|_2 \leq \sqrt{\delta}$ . Hence we can invoke Lemma 7, to conclude that,

$$\left| \left\langle \tilde{A}_i, \tilde{B}_j \right\rangle_{\mathcal{G}_\rho^{\otimes n_0}} - \left\langle A_i^{(2)}, B_j^{(2)} \right\rangle_{\mathcal{G}_\rho^{\otimes n_0}} \right| \leq 2\sqrt{\delta}. \quad (4)$$

Thus we started with functions  $A : \mathbb{R}^n \rightarrow \Delta_k$  and  $B : \mathbb{R}^n \rightarrow \Delta_k$  and ended with functions  $\tilde{A} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  and  $\tilde{B} : \mathbb{R}^{n_0} \rightarrow \Delta_k$  such that for every  $i, j \in [k]$  (by combining Equations 2, 3 and 4) it holds that,

$$\left| \left\langle \tilde{A}_i, \tilde{B}_j \right\rangle_{\mathcal{G}_\rho^{\otimes n_0}} - \left\langle A_i, B_j \right\rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq O(\sqrt{\delta}).$$

Thus, more strongly, if we instantiate  $\delta = O(\varepsilon^2/k^4)$ , then we get that our entire transformation succeeds with probability  $1 - \varepsilon$  in obtaining  $\tilde{A}$  and  $\tilde{B}$  such that,

$$d_{\text{TV}} \left( (A(\mathbf{X}), B(\mathbf{Y}))_{\mathcal{G}_\rho^{\otimes n}}, (\tilde{A}(\mathbf{a}), \tilde{B}(\mathbf{b}))_{\mathcal{G}_\rho^{\otimes n_0}} \right) \leq \varepsilon.$$

It is easy to see that  $n_0$  works out to be

$$n_0 = \frac{d^{O(d)}}{\delta^4} = \exp\left(\tilde{O}\left(\frac{k^{4.5}}{\varepsilon^2(1-\rho)}\right)\right) = \exp\left(\text{poly}\left(k, \frac{1}{\varepsilon}, \frac{1}{1-\rho}\right)\right). \quad \blacktriangleleft$$

---

## References

- 1 OpenQIPproblemsWiki - All the Bell Inequalities. <http://qig.itp.uni-hannover.de/qiproblems/1>. Accessed: 2016-07-12.
- 2 Rudolf Ahlswede and Imre Csiszár. Common randomness in information theory and cryptography. part i: secret sharing. *IEEE Transactions on Information Theory*, 39(4), 1993.
- 3 Rudolf Ahlswede and Imre Csiszár. Common randomness in information theory and cryptography. ii. cr capacity. *Information Theory, IEEE Transactions on*, 44(1):225–240, 1998.
- 4 Noga Alon and Eyal Lubetzky. The shannon capacity of a graph and the independence numbers of its powers. *Information Theory, IEEE Transactions on*, 52(5):2172–2176, 2006.
- 5 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. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.55.
- 6 Mohammad Bavarian, Dmitry Gavinsky, and Tsuyoshi Ito. On the role of shared randomness in simultaneous communication. In *Automata, Languages, and Programming*, pages 150–162. Springer, 2014.
- 7 Salman Beigi. A new quantum data processing inequality. *CoRR*, abs/1210.1689, 2012. arXiv:1210.1689.
- 8 Salman Beigi and Amin Gohari. On the duality of additivity and tensorization. *arXiv preprint arXiv:1502.00827*, 2015.
- 9 Andrej Bogdanov and Elchanan Mossel. On extracting common random bits from correlated sources. *Information Theory, IEEE Transactions on*, 57(10):6351–6355, 2011.
- 10 Christer Borell. Geometric bounds on the ornstein-uhlenbeck velocity process. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 70(1):1–13, 1985.
- 11 Gilles Brassard and Louis Salvail. Secret-key reconciliation by public discussion. In *advances in Cryptology—EUROCRYPT’93*, pages 410–423. Springer, 1994.
- 12 Mark Braverman and Young Kun-Ko. Information value of two-prover games. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPICs*, pages 12:1–12:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPICs.ITCS.2018.12.
- 13 Mark Braverman and Anup Rao. Information equals amortized communication. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 748–757. IEEE, 2011.
- 14 Mark Braverman and Jon Schneider. Information complexity is computable. *arXiv preprint arXiv:1502.02971*, 2015.
- 15 Clement Canonne, Venkat Guruswami, Raghu Meka, and Madhu Sudan. Communication with imperfectly shared randomness. *ITCS*, 2014.
- 16 Eric Chitambar, Runyao Duan, and Yaoyun Shi. Tripartite entanglement transformations and tensor rank. *Physical review letters*, 101(14):140502, 2008.
- 17 Imre Csiszár and Prakash Narayan. Common randomness and secret key generation with a helper. *Information Theory, IEEE Transactions on*, 46(2):344–366, 2000.
- 18 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003. doi:10.1002/rsa.10073.

- 19 Anindya De, Elchanan Mossel, and Joe Neeman. Noise stability is computable and approximately low-dimensional. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPICs*, pages 10:1–10:11. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CCC.2017.10.
- 20 Anindya De, Elchanan Mossel, and Joe Neeman. Non interactive simulation of correlated distributions is decidable. 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 2728–2746. SIAM, 2018. doi:10.1137/1.9781611975031.174.
- 21 Anindya De and Rocco A Servedio. Efficient deterministic approximate counting for low-degree polynomial threshold functions. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 832–841. ACM, 2014.
- 22 Payam Delgosha and Salman Beigi. Impossibility of local state transformation via hypercontractivity. *CoRR*, abs/1307.2747, 2013. arXiv:1307.2747.
- 23 Payam Delgosha and Salman Beigi. Impossibility of local state transformation via hypercontractivity. *Communications in Mathematical Physics*, 332(1):449–476, 2014.
- 24 Peter Gács and János Körner. Common information is far less than mutual information. *Problems of Control and Information Theory*, 2(2):149–162, 1973.
- 25 Hans Gebelein. Das statistische problem der korrelation als variations-und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 21(6):364–379, 1941.
- 26 Badih Ghazi and T. S. Jayram. Resource-efficient common randomness and secret-key schemes. 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 1834–1853. SIAM, 2018. doi:10.1137/1.9781611975031.120.
- 27 Badih Ghazi, Pritish Kamath, and Prasad Raghavendra. Dimension reduction for polynomials over gaussian space and applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:125, 2017. URL: <https://eccc.weizmann.ac.il/report/2017/125>.
- 28 Badih Ghazi, Pritish Kamath, and Madhu Sudan. Communication complexity of permutation-invariant functions. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1902–1921. SIAM, 2016. doi:10.1137/1.9781611974331.ch134.
- 29 Badih Ghazi, Pritish Kamath, and Madhu Sudan. Decidability of non-interactive simulation of joint distributions. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 545–554. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.65.
- 30 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.
- 31 Steven Heilman. Euclidean partitions optimizing noise stability. *CoRR*, abs/1211.7138, 2012. arXiv:1211.7138.
- 32 Steven Heilman, Elchanan Mossel, and Joe Neeman. Standard simplices and pluralities are not the most noise stable. *Israel Journal of Mathematics*, 213(1):33–53, 2016.
- 33 Hermann O Hirschfeld. A connection between correlation and contingency. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 520–524. Cambridge Univ Press, 1935.
- 34 Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(1):141–172, 2009. doi:10.4086/toc.2009.v005a008.

- 35 Marcus Isaksson and Elchanan Mossel. Maximally stable gaussian partitions with discrete applications. *Israel Journal of Mathematics*, 189(1):347–396, 2012.
- 36 William Johnson and Joram Lindenstrauss. Extensions of lipschitz maps into a hilbert space. 26:189–206, 01 1984.
- 37 Sudeep Kamath and Venkat Anantharam. On non-interactive simulation of joint distributions. *IEEE Trans. Information Theory*, 62(6):3419–3435, 2016. doi:10.1109/TIT.2016.2553672.
- 38 Julia Kempe, Hirotada Kobayashi, Keiji Matsumoto, Ben Toner, and Thomas Vidick. Entangled games are hard to approximate. *SIAM J. Comput.*, 40(3):848–877, 2011. doi:10.1137/090751293.
- 39 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O’Donnell. Optimal inapproximability results for max-cut and other 2-variable csp’s? *SIAM Journal on Computing*, 37(1):319–357, 2007.
- 40 László Lovász. On the shannon capacity of a graph. *Information Theory, IEEE Transactions on*, 25(1):1–7, 1979.
- 41 Ueli M Maurer. Secret key agreement by public discussion from common information. *Information Theory, IEEE Transactions on*, 39(3):733–742, 1993.
- 42 Elchanan Mossel. Gaussian bounds for noise correlation of functions. *Geometric and Functional Analysis*, 19(6):1713–1756, 2010.
- 43 Elchanan Mossel and Ryan O’Donnell. Coin flipping from a cosmic source: On error correction of truly random bits. *arXiv preprint math/0406504*, 2004.
- 44 Elchanan Mossel, Ryan O’Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 21–30. IEEE, 2005.
- 45 Elchanan Mossel, Ryan O’Donnell, Oded Regev, Jeffrey E Steif, and Benny Sudakov. Non-interactive correlation distillation, inhomogeneous markov chains, and the reverse bonami-beckner inequality. *Israel Journal of Mathematics*, 154(1):299–336, 2006.
- 46 Ilan Newman. Private vs. common random bits in communication complexity. *Information processing letters*, 39(2):67–71, 1991.
- 47 Michael A Nielsen. Conditions for a class of entanglement transformations. *Physical Review Letters*, 83(2):436, 1999.
- 48 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM J. Comput.*, 40(6):1871–1891, 2011. doi:10.1137/080734042.
- 49 Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. doi:10.1137/S0097539795280895.
- 50 Ran Raz. A counterexample to strong parallel repetition. *SIAM J. Comput.*, 40(3):771–777, 2011. doi:10.1137/090747270.
- 51 Renato Renner and Stefan Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *Advances in cryptology-ASIACRYPT 2005*, pages 199–216. Springer, 2005.
- 52 Alfréd Rényi. On measures of dependence. *Acta mathematica hungarica*, 10(3-4):441–451, 1959.
- 53 Claude E Shannon. The zero error capacity of a noisy channel. *Information Theory, IRE Transactions on*, 2(3):8–19, 1956.
- 54 Hans S Witsenhausen. On sequences of pairs of dependent random variables. *SIAM Journal on Applied Mathematics*, 28(1):100–113, 1975.
- 55 Pawel Wolff. Hypercontractivity of simple random variables. *Studia Mathematica*, 180(3):219–236, 2007.
- 56 Aaron D. Wyner. The common information of two dependent random variables. *IEEE Trans. Information Theory*, 21(2):163–179, 1975. doi:10.1109/TIT.1975.1055346.

## A Proofs of Mean and Variance Bounds in Dimension Reduction

In this section, we provide the proof of Lemma 10. This is the main new technical component introduced in this paper. Even though the calculations might seem cumbersome, they involve mostly elementary steps. To understand the high level picture, we recommend the reader to go through a short proof sketch presented in subsection 3.1.

Recall that starting with degree  $d$  multilinear polynomials  $A : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}$ , we defined functions  $A_M : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$  and  $B_M : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$ , for  $\mathbf{M} \sim \gamma_1^{\otimes(n \times n_0)}$ , as

$$A_M(a) = A\left(\frac{\mathbf{M}a}{\|a\|_2}\right) \quad \text{and} \quad B_M(b) = B\left(\frac{\mathbf{M}b}{\|b\|_2}\right) \quad \text{for } a, b \in \mathbb{R}^{n_0} \setminus \{0\} .$$

and we defined their correlation as  $F(\mathbf{M}) \stackrel{\text{def}}{=} \langle A_M, B_M \rangle_{\mathcal{G}_\rho^{\otimes n_0}}$ . Lemma 10 proves bounds on the mean and variance of  $F(\mathbf{M})$ , which we restate below for convenience.

► **Lemma 12.** (Mean & Variance Bound). *Given  $d$  and  $\delta$ , there exists an explicitly computable  $n_0 := n_0(d, \delta)$  such that for  $\mathbf{M} \sim \gamma_1^{\otimes(n \times n_0)}$ ,*

$$\begin{aligned} \left| \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| &\leq \delta && \text{(Mean bound)} \\ \text{Var}_{\mathbf{M}}(F(\mathbf{M})) &\leq \delta && \text{(Variance bound)} \end{aligned}$$

In particular, one may take  $n_0 = \frac{d^{O(d)}}{\delta^2}$ .

We break down the full proof into the following three modular steps.

1. In subsection A.1, we prove a *meta-lemma* (Lemma 13) that will help us prove both the mean and variance bounds; indeed this meta-lemma is at the heart of why Theorem 8 holds. Morally, this lemma says that if we have an expectation of a product of a small number of inner products of normalized correlated Gaussian vectors, then, we can exchange the product and the expectations while incurring only a small additive error. Lemma 13 is the main take away from this subsection, and the reader may skip to subsection A.2 and subsection A.3 to see the rest of the proof.
2. In subsection A.2, we prove bounds on the mean and co-variances of degree- $d$  multilinear monomials, under the above transformation of replacing  $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^n$  (inputs to  $A$  and  $B$ ) by  $\frac{\mathbf{M}\mathbf{a}}{\|\mathbf{a}\|_2}$  and  $\frac{\mathbf{M}\mathbf{b}}{\|\mathbf{b}\|_2}$  respectively.
3. In subsection A.3, we finally use the above bounds on mean and co-variances of degree- $d$  multilinear monomials in order to prove Lemma 10.

► **Remark.** To make our notations convenient, we will often write equations such as  $\alpha = \beta \pm \varepsilon$  which is to be interpreted as  $|\alpha - \beta| \leq \varepsilon$ .

### A.1 Product of Inner Products of Normalized Correlated Gaussian Vectors

The following is the main lemma in this subsection (this is the *meta-lemma* alluded to earlier).

► **Lemma 13.** *Given  $d, D \in \mathbb{Z}_{\geq 0}$  and  $\delta > 0$  (with  $D$  sufficiently larger than  $d$ ), let  $(\mathbf{u}_1, \dots, \mathbf{u}_d, \mathbf{v}_1, \dots, \mathbf{v}_d)$  be a  $2dD$ -dimensional multivariate Gaussian distribution such that,*

- *each  $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^D$  are marginally distributed as standard  $D$ -dimensional Gaussians  $\gamma_D$ .*
- *for each  $j \in [D]$ , the joint distribution  $(\mathbf{u}_{1,j}, \dots, \mathbf{u}_{d,j}, \mathbf{v}_{1,j}, \dots, \mathbf{v}_{d,j})$ , is independent across different values of  $j$ .*

Then,

$$\left| \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}_i} \left[ \prod_{i=1}^d \frac{\langle \mathbf{u}_i, \mathbf{v}_i \rangle}{\|\mathbf{u}_i\|_2 \|\mathbf{v}_i\|_2} \right] - \prod_{i=1}^d \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}_i} \left[ \frac{\langle \mathbf{u}_i, \mathbf{v}_i \rangle}{D} \right] \right| \leq \frac{d^{O(d)}}{D}.$$

We point out that there are two steps taking place in Lemma 13:

- (i) the replacement of  $\|\mathbf{u}_i\|_2$  (and  $\|\mathbf{v}_i\|_2$ ) by  $\sqrt{D}$  (around which it is tightly concentrated),
- (ii) the interchanging of the expectation and the product.

We will handle each of these changes one by one.

### Product of Negative Moments of $\ell_2$ -norm of Correlated Gaussian vectors

In order to handle the replacement of  $\|\mathbf{u}_i\|_2$  (and  $\|\mathbf{v}_i\|_2$ ) by  $\sqrt{D}$ , we will prove some bounds on the mean and variance of products of negative powers of the  $\ell_2$ -norm of a standard Gaussian vector.

► **Lemma 14.** *Let  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_\ell$  be (possibly correlated) multivariate Gaussians where each  $\mathbf{w}_i \in \mathbb{R}^D$  is marginally distributed as  $\gamma_D$ , and let  $d_1, d_2, \dots, d_\ell$  be non-negative integers with  $d := \sum_{i=1}^\ell d_i$ . Then,*

$$\left| \mathbb{E} \left[ \prod_{i=1}^\ell \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] - \frac{1}{D^{d/2}} \right| \leq O\left(\frac{d^5}{D^{\frac{d}{2}+1}}\right),$$

$$\text{Var} \left[ \prod_{i=1}^\ell \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] \leq O\left(\frac{d^5}{D^{d+1}}\right).$$

► **Remark.** It is conceivable that the bounds in Lemma 14 could be improved in terms of the dependence on  $d$ . However, this was not central to our application, so we go ahead with the stated bounds. The main point to note in the above lemma is the extra factor of  $D$  in the denominator.

We start out by first proving the base case where we have a single vector  $\mathbf{w}$ , that is,  $\ell = 1$ .

► **Proposition 15.** *There exists an absolute constant  $C$  such that for sufficiently large  $d, D \in \mathbb{Z}_{>0}$  satisfying  $D > Cd^2$ , we have that for  $\mathbf{w} \sim \gamma_D$ ,*

$$\left| \mathbb{E}_{\mathbf{w}} \left[ \frac{1}{\|\mathbf{w}\|_2^d} \right] - \frac{1}{D^{d/2}} \right| \leq C \cdot \left( \frac{d^2}{D^{\frac{d}{2}+1}} \right), \quad (5)$$

$$\text{Var}_{\mathbf{w}} \left[ \frac{1}{\|\mathbf{w}\|_2^d} \right] \leq 8C \cdot \left( \frac{d^2}{D^{d+1}} \right). \quad (6)$$

**Proof.** It is well-known that the distribution of  $\|\mathbf{w}\|_2$  follows a  $\chi$ -distribution with parameter  $D$ , and whose probability density function is given by

$$f_D(x) = \frac{x^{D-1} \cdot e^{-\frac{x^2}{2}}}{2^{\frac{D}{2}-1} \cdot \Gamma(\frac{D}{2})}, \quad (x \in \mathbb{R}_{\geq 0})$$

where  $\Gamma(\cdot)$  denotes the Gamma function. Thus, we have that

$$\begin{aligned} \mathbb{E}_{\mathbf{w}} \left[ \frac{1}{\|\mathbf{w}\|_2^d} \right] &= \int_0^\infty \frac{1}{x^d} \cdot f_D(x) dx = \int_0^\infty \frac{x^{D-d-1} \cdot e^{-\frac{x^2}{2}}}{2^{\frac{D}{2}-1} \cdot \Gamma(\frac{D}{2})} dx \\ &= \frac{2^{\frac{D-d-1}{2}} \cdot \Gamma(\frac{D-d}{2})}{2^{\frac{D}{2}-1} \cdot \Gamma(\frac{D}{2})} = \frac{1}{D^{d/2}} \cdot \left( 1 \pm O\left(\frac{d^2}{D}\right) \right), \end{aligned}$$



where the last equality follows from the Stirling's approximation of the Gamma function, which holds for every real number  $z > 0$ :

$$\Gamma(z + 1) = \sqrt{2\pi z} \cdot \left(\frac{z}{e}\right)^z \cdot \left(1 \pm O\left(\frac{1}{z}\right)\right).$$

This completes the proof of Equation 5, for the explicit constant  $C$  that can be derived from the Stirling's approximation. Now, Equation 6 immediately follows as:

$$\begin{aligned} \text{Var}_{\mathbf{w}} \left[ \frac{1}{\|\mathbf{w}\|^d} \right] &= \mathbb{E}_{\mathbf{w}} \left[ \frac{1}{\|\mathbf{w}\|^{2d}} \right] - \mathbb{E}_{\mathbf{w}} \left[ \frac{1}{\|\mathbf{w}\|^d} \right]^2 \\ &= \left( \frac{1}{D^d} \pm C \cdot \left( \frac{(2d)^2}{D^{d+1}} \right) \right) - \left( \frac{1}{D^{d/2}} \pm C \cdot \left( \frac{d^2}{D^{d/2+1}} \right) \right)^2 \\ &\leq 8C \cdot \left( \frac{d^2}{D^{d+1}} \right), \end{aligned}$$

where, we use that  $D$  is sufficiently large that  $C^2 \left( \frac{d^4}{D^{d+2}} \right) < 2C \cdot \left( \frac{d^2}{D^{d+1}} \right)$ , i.e.  $D > Cd^2$ . ◀

We now show how to generalize the above to prove Lemma 14.

**Proof of Lemma 14.** More specifically, we will show that,

$$\left| \mathbb{E} \left[ \prod_{i=1}^{\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] - \frac{1}{D^{d/2}} \right| \leq C \cdot \ell^3 \cdot \left( \frac{d^2}{D^{d/2+1}} \right) \quad (7)$$

$$\text{Var} \left[ \prod_{i=1}^{\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] \leq 8C \cdot \ell^3 \cdot \left( \frac{d^2}{D^{d+1}} \right) \quad (8)$$

where  $C$  is the absolute constant (as obtained in Proposition 15). This implies the lemma since  $\ell \leq d$ .

We proceed by induction on  $\ell$  (more specifically on  $\log \ell$ ). For  $\ell = 1$ , the bound immediately follows from Proposition 15. For the inductive step, we assume that the bound in Equations 7 and 8 holds for  $\ell$ , and we prove that the bound also holds for  $2\ell$ . While it may seem that our bounds are being proven only when  $\ell$  is a power of 2, it is not hard to see that our proof could be done for non powers of 2 as well, giving a bound that is monotonically increasing in  $\ell$  and hence it suffices having proved it for  $\ell$  that are powers of 2. Let  $d_1, d_2, \dots, d_{2\ell}$  be non-negative integers with  $d := \sum_{i=1}^{2\ell} d_i$ . For notational convenience, let  $s_1 = \sum_{i=1}^{\ell} d_i$  and  $s_2 = \sum_{i=\ell+1}^{2\ell} d_i$ , and so  $d = s_1 + s_2$ .

We will first prove Equation 7 inductively by using the following idea: for any two random variables  $\mathbf{X}$  and  $\mathbf{Y}$ , we have  $\mathbb{E}[\mathbf{X}\mathbf{Y}] = \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{Y}] + \text{Cov}[\mathbf{X}, \mathbf{Y}]$  and  $|\text{Cov}[\mathbf{X}, \mathbf{Y}]| \leq \sqrt{\text{Var}[\mathbf{X}] \cdot \text{Var}[\mathbf{Y}]}$  and hence  $\mathbb{E}[\mathbf{X}\mathbf{Y}] = \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{Y}] \pm \sqrt{\text{Var}[\mathbf{X}] \cdot \text{Var}[\mathbf{Y}]}$ . Thus, we get,

$$\begin{aligned} \mathbb{E} \left[ \prod_{i=1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] &= \mathbb{E} \left[ \prod_{i=1}^{\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] \cdot \mathbb{E} \left[ \prod_{i=\ell+1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] \\ &\pm \sqrt{\text{Var} \left[ \prod_{i=1}^{\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] \cdot \text{Var} \left[ \prod_{i=\ell+1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right]}. \end{aligned} \quad (9)$$

Using the inductive assumption w.r.t.  $\ell$ , we get that,

$$\mathbb{E} \left[ \prod_{i=1}^{\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] = \frac{1}{D^{s_1/2}} \left( 1 \pm C \cdot \ell^3 \cdot \left( \frac{s_1^2}{D} \right) \right) \quad (10)$$

$$\mathbb{E} \left[ \prod_{i=\ell+1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] = \frac{1}{D^{s_2/2}} \left( 1 \pm C \cdot \ell^3 \cdot \left( \frac{s_2^2}{D} \right) \right) \quad (11)$$

and

$$\text{Var} \left[ \prod_{i=1}^{\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] \leq \frac{1}{D^{s_1}} \cdot 8C \cdot \ell^3 \cdot \left( \frac{s_1^2}{D} \right) \quad (12)$$

$$\text{Var} \left[ \prod_{i=\ell+1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] \leq \frac{1}{D^{s_2}} \cdot 8C \cdot \ell^3 \cdot \left( \frac{s_2^2}{D} \right) \quad (13)$$

Plugging Equations 10, 11, 12 and 13 in Equation 9, it is not hard to see that,

$$\mathbb{E} \left[ \prod_{i=1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] = \frac{1}{D^{d/2}} \left( 1 \pm C \cdot (2\ell)^3 \cdot \left( \frac{d^2}{D} \right) \right).$$

This completes the proof of Equation 7. Now, Equation 8 follows easily as,

$$\begin{aligned} \text{Var} \left[ \prod_{i=1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right] &= \mathbb{E} \left[ \prod_{i=1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{2d_i}} \right] - \mathbb{E} \left[ \prod_{i=1}^{2\ell} \frac{1}{\|\mathbf{w}_i\|_2^{d_i}} \right]^2 \\ &= \left( \frac{1}{D^d} \pm C \cdot (2\ell)^3 \cdot \left( \frac{(2d)^2}{D^{d+1}} \right) \right) - \left( \frac{1}{D^{d/2}} \pm C \cdot (2\ell)^3 \cdot \left( \frac{d^2}{D^{d/2+1}} \right) \right)^2 \\ &\leq 8C \cdot (2\ell)^3 \cdot \left( \frac{d^2}{D^{d+1}} \right). \quad \blacktriangleleft \end{aligned}$$

## Interchanging Product and Expectation

In order to handle the interchanging of the product and expectation operations, we will show the following lemma.

► **Lemma 16.** *Let  $(\mathbf{u}_1, \dots, \mathbf{u}_d, \mathbf{v}_1, \dots, \mathbf{v}_d)$  be distributed as in Lemma 13. Then,*

$$\left| \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}_i} \left[ \prod_{i=1}^d \langle \mathbf{u}_i, \mathbf{v}_i \rangle \right] - \prod_{i=1}^d \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}_i} [\langle \mathbf{u}_i, \mathbf{v}_i \rangle] \right| \leq d^{O(d)} \cdot D^{d-1}.$$

► **Remark.** The  $d^{O(d)}$  term has an explicit expression, although we only highlight its qualitative nature for clarity. Again, it is conceivable that the bounds in Lemma 16 could be improved in terms of the dependence on  $d$ , although we suspect that it is tight upto constant factors in the exponent. Anyhow, this was not central to our application, so we go ahead with the stated bounds. The main point to note in the above lemma is that the exponent of  $D$  is  $(d-1)$  instead of  $d$ .

To prove the lemma, we first obtain the following proposition on moments of a multivariate Gaussian.

► **Proposition 17.** *Let  $\mathbf{w} \in \mathbb{R}^\ell$  be any multivariate Gaussian vector with each coordinate marginally distributed according to  $\gamma_1$ . Let  $d_1, d_2, \dots, d_\ell$  be non-negative integers such that  $d := \sum_{i=1}^\ell d_i$ . Then,*

$$\left| \mathbb{E} \left[ \prod_{i=1}^\ell \mathbf{w}_i^{d_i} \right] \right| \leq (2d)^{3d}.$$

**Proof.** More specifically we will show that when  $\ell$  is a power of 2,

$$\left| \mathbb{E} \left[ \prod_{i=1}^\ell \mathbf{w}_i^{d_i} \right] \right| \leq 2^{\ell-1} (\ell d)^d. \tag{14}$$

It is easy to see that this immediately implies the bound of  $2^d \cdot d^{2d}$  in the main lemma, since  $\ell \leq d$ . However if  $\ell$  is not a power of 2 we can round it up to the nearest power of 2, which amounts to substituting  $\ell \leq 2d$  in the above, obtaining a bound of  $2^{3d} \cdot d^{2d} \leq (2d)^{3d}$ .

We proceed by induction on  $\ell$  (more specifically on  $\log \ell$ ). For  $\ell = 1$ , we use the well-known fact that for  $w \sim \gamma_1$ ,

$$|\mathbb{E}[w^d]| = \begin{cases} 0 & \text{if } d \text{ is odd} \\ (d-1)!! & \text{if } d \text{ is even} \end{cases} \leq d^d,$$

where  $(d-1)!!$  denotes the double factorial of  $(d-1)$ , i.e., the product of all integers from 1 to  $d-1$  that have the same parity as  $d-1$ . For the inductive step, we assume that the bound in (14) holds for  $\ell$  and we show that it also holds for  $2\ell$ . For notational convenience, let  $s_1 = \sum_{i=1}^\ell d_i$  and  $s_2 = \sum_{i=\ell+1}^{2\ell} d_i$ , and so  $d = s_1 + s_2$ .

The main idea to prove the inductive step is simply the Cauchy-Schwarz inequality.

$$\begin{aligned} \left| \mathbb{E} \left[ \prod_{i=1}^{2\ell} \mathbf{w}_i^{d_i} \right] \right| &\leq \sqrt{\mathbb{E} \left[ \prod_{i=1}^\ell \mathbf{w}_i^{2d_i} \right] \cdot \mathbb{E} \left[ \prod_{i=\ell+1}^{2\ell} \mathbf{w}_i^{2d_i} \right]} \\ &\leq \sqrt{2^{\ell-1} (2\ell s_1)^{2s_1} \cdot 2^{\ell-1} (2\ell s_2)^{2s_2}} \leq 2^{2\ell-1} (2\ell d)^d, \end{aligned}$$

where, we use the inductive assumption regarding product of  $\ell$  terms and that  $s_1 + s_2 = d$ . ◀

Using the above proposition, we are now able to prove Lemma 16.

**Proof of Lemma 16.** Let  $S \subseteq [D]^d$  be the set of all tuples  $c \in [D]^d$  such that  $c_j \neq c_k$  for all  $j \neq k \in [d]$ . Let  $\bar{S}$  denote the complement of  $S$  in  $[D]^d$ . Note that  $|\bar{S}| \leq d^2 \cdot D^{d-1}$ . We have that

$$\begin{aligned} \mathbb{E} \left[ \prod_{i=1}^d \langle \mathbf{u}_i, \mathbf{v}_i \rangle \right] &= \mathbb{E} \left[ \prod_{i=1}^d \sum_{k=1}^D \mathbf{u}_{i,k} \mathbf{v}_{i,k} \right] = \sum_{c \in [D]^d} \mathbb{E} \left[ \prod_{i=1}^d \mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i} \right] \\ &= \sum_{c \in S} \mathbb{E} \left[ \prod_{i=1}^d \mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i} \right] + \sum_{c \in \bar{S}} \mathbb{E} \left[ \prod_{i=1}^d \mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i} \right] \\ &= \sum_{c \in S} \prod_{i=1}^d \mathbb{E}[\mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i}] + \sum_{c \in \bar{S}} \mathbb{E} \left[ \prod_{i=1}^d \mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i} \right], \end{aligned} \tag{15}$$

where the last equality follows from the assumption that the distribution of the  $j$ -th coordinates  $(\mathbf{u}_{1,j}, \dots, \mathbf{u}_{d,j}, \mathbf{v}_{1,j}, \dots, \mathbf{v}_{d,j})$  is independent across  $j \in [D]$ . On the other hand, we have

that

$$\begin{aligned}
 \prod_{i=1}^d \mathbb{E}[\langle \mathbf{u}_i, \mathbf{v}_i \rangle] &= \prod_{i=1}^d \mathbb{E} \left[ \sum_{k=1}^D \mathbf{u}_{i,k} \mathbf{v}_{i,k} \right] = \sum_{c \in [D]^d} \prod_{i=1}^d \mathbb{E}[\mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i}] \\
 &= \sum_{c \in \mathcal{S}} \prod_{i=1}^d \mathbb{E}[\mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i}] + \sum_{c \in \bar{\mathcal{S}}} \prod_{i=1}^d \mathbb{E}[\mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i}]
 \end{aligned} \tag{16}$$

Combining Equations 15 and 16, we get

$$\begin{aligned}
 \left| \mathbb{E} \left[ \prod_{i=1}^d \langle \mathbf{u}_i, \mathbf{v}_i \rangle \right] - \prod_{i=1}^d \mathbb{E}[\langle \mathbf{u}_i, \mathbf{v}_i \rangle] \right| &= \left| \sum_{c \in \bar{\mathcal{S}}} \left( \mathbb{E} \left[ \prod_{i=1}^d \mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i} \right] - \prod_{i=1}^d \mathbb{E}[\mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i}] \right) \right| \\
 &\leq |\bar{\mathcal{S}}| \cdot \max_{c \in \bar{\mathcal{S}}} \left| \mathbb{E} \left[ \prod_{i=1}^d \mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i} \right] - \prod_{i=1}^d \mathbb{E}[\mathbf{u}_{i,c_i} \mathbf{v}_{i,c_i}] \right| \\
 &\leq d^2 \cdot D^{d-1} \cdot ((2d)^{3d} + 1) \leq d^{O(d)} \cdot D^{d-1},
 \end{aligned}$$

where the second last inequality follows from the fact that  $|\bar{\mathcal{S}}| \leq d^2 \cdot D^{d-1}$  and from Proposition 17.  $\blacktriangleleft$

### Putting things together to prove Lemma 13

**Proof of Lemma 13.** We show the following bounds, which immediately imply Lemma 13.

$$\left| \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}} \left[ \prod_{i=1}^d \frac{\langle \mathbf{u}_i, \mathbf{v}_i \rangle}{\|\mathbf{u}_i\|_2 \|\mathbf{v}_i\|_2} \right] - \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}} \left[ \prod_{i=1}^d \frac{\langle \mathbf{u}_i, \mathbf{v}_i \rangle}{D} \right] \right| \leq \frac{d^{O(d)}}{D}. \tag{17}$$

$$\left| \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}} \left[ \prod_{i=1}^d \frac{\langle \mathbf{u}_i, \mathbf{v}_i \rangle}{D} \right] - \prod_{i=1}^d \mathbb{E}_{\{\mathbf{u}_i, \mathbf{v}_i\}} \left[ \frac{\langle \mathbf{u}_i, \mathbf{v}_i \rangle}{D} \right] \right| \leq \frac{d^{O(d)}}{D}. \tag{18}$$

Note that Equation 18 is simply a restatement of Lemma 16. To prove Equation 17, we define the random variables

$$\mathbf{W} := \prod_{i=1}^d \langle \mathbf{u}_i, \mathbf{v}_i \rangle \quad \text{and} \quad \mathbf{Z} := \prod_{i=1}^d \frac{1}{\|\mathbf{u}_i\|_2 \|\mathbf{v}_i\|_2} - \frac{1}{D^d}.$$

Note that Equation 17 is equivalent to showing bounds on  $|\mathbb{E}[\mathbf{W} \cdot \mathbf{Z}]|$ . In order to do so, we use the following four bounds:

1.  $|\mathbb{E}[\mathbf{W}]| \leq D^d + d^{O(d)} \cdot D^{d-1}$ . Since, by Lemma 16, we have that

$$|\mathbb{E}[\mathbf{W}]| \leq \left| \prod_{i=1}^d \mathbb{E}[\langle \mathbf{u}_i, \mathbf{v}_i \rangle] \right| + d^{O(d)} \cdot D^{d-1} \leq D^d + d^{O(d)} \cdot D^{d-1}$$

2.  $\text{Var}[\mathbf{W}] \leq d^{O(d)} \cdot D^{2d-1}$ . Since,

$$\begin{aligned}
 \text{Var}[\mathbf{W}] &= \mathbb{E}[\mathbf{W}^2] - [\mathbb{E} \mathbf{W}]^2 \\
 &= \mathbb{E} \left[ \prod_{i=1}^d \langle \mathbf{u}_i, \mathbf{v}_i \rangle^2 \right] - \left[ \mathbb{E} \prod_{i=1}^d \langle \mathbf{u}_i, \mathbf{v}_i \rangle \right]^2 \\
 &\leq d^{O(d)} \cdot D^{2d-1} \quad \dots \text{(from Lemma 16)}
 \end{aligned}$$

3.  $|\mathbb{E}[\mathbf{Z}]| = O\left(\frac{d^5}{D^{d+1}}\right)$  (follows exactly from Lemma 14).

4.  $\text{Var}[\mathbf{Z}] = O\left(\frac{d^5}{D^{2d+1}}\right)$  (follows exactly from Lemma 14).

Thus, we can bound  $|\mathbb{E}[\mathbf{W} \cdot \mathbf{Z}]|$  as,

$$|\mathbb{E}[\mathbf{W} \cdot \mathbf{Z}]| \leq |\mathbb{E}[\mathbf{W}]| \cdot |\mathbb{E}[\mathbf{Z}]| + \sqrt{\text{Var}[\mathbf{W}] \cdot \text{Var}[\mathbf{Z}]} \leq \frac{d^{O(d)}}{D}.$$

This completes the proof of Equation 17 and hence of Lemma 13.  $\blacktriangleleft$

## A.2 Mean & Variance Bounds for Multilinear Monomials

For the rest of this section, we simplify our notations as follows:

- For  $(\mathbf{a}, \mathbf{b}) \sim \mathcal{G}_\rho^{\otimes n_0}$ , we will use  $\tilde{\mathbf{a}}$  and  $\tilde{\mathbf{b}}$  to denote the normalized vectors  $\frac{\mathbf{a}}{\|\mathbf{a}\|_2}$  and  $\frac{\mathbf{b}}{\|\mathbf{b}\|_2}$  respectively.
- We will use  $\mathbf{U} \in \mathbb{R}^n$  to denote  $\mathbf{M}\tilde{\mathbf{a}}$  and similarly  $\mathbf{V} \in \mathbb{R}^n$  to denote  $\mathbf{M}\tilde{\mathbf{b}}$ . We will also have independent variables  $(\mathbf{a}', \mathbf{b}') \sim \mathcal{G}_\rho^{\otimes n_0}$ , for which we use  $\mathbf{U}' = \mathbf{M}\tilde{\mathbf{a}}'$  and  $\mathbf{V}' = \mathbf{M}\tilde{\mathbf{b}}'$ .
- $U_i$  denotes the  $i$ -th coordinate of  $\mathbf{U}$ . Similarly,  $\mathbf{m}_i \in \mathbb{R}^{n_0}$  is the  $i$ -th row of  $\mathbf{M}$ . Note that  $U_i = \langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle$ . For  $S \subseteq [n]$ , let  $\mathbf{U}_S$  denote  $\prod_{i \in S} U_i = \prod_{i \in S} \langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle$ . Similarly for  $\mathbf{V}_S$ .
- We will take expectations over random variables  $\mathbf{M}, \mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'$ . It will be understood that we are sampling  $\mathbf{M} \sim \gamma_1^{\otimes (n \times n_0)}$ . Also,  $(\mathbf{a}, \mathbf{b})$  and  $(\mathbf{a}', \mathbf{b}')$  are independently sampled from  $\mathcal{G}_\rho^{\otimes n_0}$ .

► **Lemma 18** (Mean bounds for monomials). *Given parameter  $d$  and  $\delta$ , there exists an explicitly computable  $n_0 := n_0(d, \delta)$  such that the following holds: For any subsets  $S, T \subseteq [n]$  satisfying  $|S|, |T| \leq d$ , it holds that,*

$$\mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} U_S V_T = \begin{cases} 0 & \text{if } S \neq T \\ \rho^{|S|} \pm \delta & \text{if } S = T \end{cases}.$$

In particular, one may take  $n_0 = \frac{d^{O(d)}}{\delta}$ .

**Proof.** We have that

$$\begin{aligned} \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} U_S V_T &= \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \left[ \prod_{i \in S} U_i \cdot \prod_{i \in T} V_i \right] \\ &= \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{M}} \left[ \prod_{i \in S \cap T} U_i V_i \cdot \prod_{i \in S \setminus T} U_i \cdot \prod_{i \in T \setminus S} V_i \right] \\ &= \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{M}} \left[ \prod_{i \in S \cap T} \langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle \langle \mathbf{m}_i, \tilde{\mathbf{b}} \rangle \cdot \prod_{i \in S \setminus T} \langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle \cdot \prod_{i \in T \setminus S} \langle \mathbf{m}_i, \tilde{\mathbf{b}} \rangle \right] \\ \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} U_S V_T &= \mathbb{E}_{\mathbf{a}, \mathbf{b}} \left[ \prod_{i \in S \cap T} \mathbb{E}_{\mathbf{m}_i} \langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle \langle \mathbf{m}_i, \tilde{\mathbf{b}} \rangle \cdot \prod_{i \in S \setminus T} \mathbb{E}_{\mathbf{m}_i} \langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle \cdot \prod_{i \in T \setminus S} \mathbb{E}_{\mathbf{m}_i} \langle \mathbf{m}_i, \tilde{\mathbf{b}} \rangle \right], \end{aligned} \quad (19)$$

where the last equality follows from the independence of the  $\mathbf{m}_i$ 's.

If  $S \neq T$ , one of  $\prod_{i \in S \setminus T} \mathbb{E}_{\mathbf{m}_i}[\langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle]$  or  $\prod_{i \in T \setminus S} \mathbb{E}_{\mathbf{m}_i}[\langle \mathbf{m}_i, \tilde{\mathbf{b}} \rangle]$  is 0. This is because even for any fixed vector  $\mathbf{a}$  and for each  $i \in [n]$ , the random variable  $\langle \mathbf{m}_i, \tilde{\mathbf{a}} \rangle$  has zero-mean (and similarly for  $\langle \mathbf{m}_i, \tilde{\mathbf{b}} \rangle$ ). The first part of the lemma now follows from Equation 19.

If  $S = T$ , Equation 19 becomes

$$\begin{aligned}
 \mathbb{E}_{\mathcal{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} U_S V_T &= \mathbb{E}_{\mathbf{a}, \mathbf{b}} \left[ \prod_{i \in S} \mathbb{E}_{\mathbf{m}_i} \frac{\langle \mathbf{m}_i, \mathbf{a} \rangle \langle \mathbf{m}_i, \mathbf{b} \rangle}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2} \right] \\
 &= \mathbb{E}_{\mathbf{a}, \mathbf{b}} \left[ \prod_{i \in S} \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2} \right] \quad \left[ \text{since } \mathbb{E}_{\mathbf{m}_i} \mathbf{m}_i \cdot \mathbf{m}_i^T = I_{n_0 \times n_0} \right] \\
 &= \prod_{i \in S} \left[ \frac{\mathbb{E}_{\mathbf{a}, \mathbf{b}} \langle \mathbf{a}, \mathbf{b} \rangle}{n_0} \right] \pm \delta \quad \left[ \text{from Lemma 13, for } n_0 = \frac{d^{O(d)}}{\delta} \right] \\
 &= \rho^{|S|} \pm \delta. \quad \blacktriangleleft
 \end{aligned}$$

► **Lemma 19** (Covariance bounds for monomials). *Given parameters  $d$  and  $\delta$ , there exists an explicitly computable  $n_0 := n_0(d, \delta)$  such that the following holds: For any subsets  $S, T, S', T' \subseteq [n]$  satisfying  $|S|, |T|, |S'|, |T'| \leq d$ , it holds that,*

$$\left| \mathbb{E}_{\mathcal{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'} [U_S V_T U_{S'} V_{T'}] - \left( \mathbb{E}_{\mathcal{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [U_S V_T] \right) \cdot \left( \mathbb{E}_{\mathcal{M}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [U_{S'} V_{T'}] \right) \right| \begin{cases} = 0 & \text{if } S \Delta T \Delta S' \Delta T' \neq \emptyset \\ \leq \delta & \text{if } S \Delta T \Delta S' \Delta T' = \emptyset \end{cases}$$

Here,  $S \Delta T \Delta S' \Delta T'$  is the symmetric difference of the sets  $S, T, S', T'$ , equivalently, the set of all  $i \in [n]$  which appear an odd number of times in the multiset  $S \sqcup T \sqcup S' \sqcup T'$ .

In particular, one may take  $n_0 = \frac{d^{O(d)}}{\delta^2}$ .

In order to prove Lemma 19, we need the following lemma.

► **Lemma 20.** *For  $\mathbf{m} \sim \gamma_{n_0}$ ,*

$$\mathbb{E}_{\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'} \left[ \left( \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}} \rangle \langle \mathbf{m}, \tilde{\mathbf{b}} \rangle \langle \mathbf{m}, \tilde{\mathbf{a}}' \rangle \langle \mathbf{m}, \tilde{\mathbf{b}}' \rangle] - \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}} \rangle \langle \mathbf{m}, \tilde{\mathbf{b}} \rangle] \cdot \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}}' \rangle \langle \mathbf{m}, \tilde{\mathbf{b}}' \rangle] \right)^2 \right] \leq O\left(\frac{1}{n_0}\right)$$

and

$$\mathbb{E}_{\mathbf{a}, \mathbf{a}'} \left[ \left( \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}} \rangle \langle \mathbf{m}, \tilde{\mathbf{a}}' \rangle] - \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}} \rangle] \cdot \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}}' \rangle] \right)^2 \right] \leq O\left(\frac{1}{n_0}\right).$$

**Proof.** To prove the first part of the lemma, consider the quantity

$$\begin{aligned}
 T(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') &:= \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}} \rangle \langle \mathbf{m}, \tilde{\mathbf{b}} \rangle \langle \mathbf{m}, \tilde{\mathbf{a}}' \rangle \langle \mathbf{m}, \tilde{\mathbf{b}}' \rangle] - \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}} \rangle \langle \mathbf{m}, \tilde{\mathbf{b}} \rangle] \cdot \mathbb{E}_{\mathbf{m}} [\langle \mathbf{m}, \tilde{\mathbf{a}}' \rangle \langle \mathbf{m}, \tilde{\mathbf{b}}' \rangle] \\
 &= \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}} \rangle \langle \tilde{\mathbf{a}}', \tilde{\mathbf{b}}' \rangle + \langle \tilde{\mathbf{a}}, \tilde{\mathbf{a}}' \rangle \langle \tilde{\mathbf{b}}, \tilde{\mathbf{b}}' \rangle + \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}}' \rangle \langle \tilde{\mathbf{a}}', \tilde{\mathbf{b}} \rangle - \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}} \rangle \langle \tilde{\mathbf{a}}', \tilde{\mathbf{b}}' \rangle \\
 &= \langle \tilde{\mathbf{a}}, \tilde{\mathbf{a}}' \rangle \langle \tilde{\mathbf{b}}, \tilde{\mathbf{b}}' \rangle + \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}}' \rangle \langle \tilde{\mathbf{a}}', \tilde{\mathbf{b}} \rangle.
 \end{aligned}$$

where we use that for any  $j \in [n_0]$ , it holds that  $\mathbb{E}_{\mathbf{m}}[\mathbf{m}_j^4] = 3$  and  $\mathbb{E}_{\mathbf{m}}[\mathbf{m}_j^2] = 1$ . Thus,

$$\begin{aligned}
 &\mathbb{E}_{\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'} [T(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}')^2] \\
 &= \mathbb{E}_{\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'} \left[ \left[ \langle \tilde{\mathbf{a}}, \tilde{\mathbf{a}}' \rangle \langle \tilde{\mathbf{b}}, \tilde{\mathbf{b}}' \rangle + \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}}' \rangle \langle \tilde{\mathbf{a}}', \tilde{\mathbf{b}} \rangle \right]^2 \right] \\
 &\leq 2 \cdot \mathbb{E}_{\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'} \left[ \langle \tilde{\mathbf{a}}, \tilde{\mathbf{a}}' \rangle^2 \langle \tilde{\mathbf{b}}, \tilde{\mathbf{b}}' \rangle^2 \right] + 2 \cdot \mathbb{E}_{\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}'} \left[ \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}}' \rangle^2 \langle \tilde{\mathbf{a}}', \tilde{\mathbf{b}} \rangle^2 \right] \\
 &\leq O\left(\frac{1}{n_0}\right),
 \end{aligned}$$

where the last step follows by two applications of Lemma 13 (with  $d = 4$ ). This completes the proof of the first part of the lemma. The second part of the lemma similarly follows from Lemma 13 (with  $d = 2$ ) along with the fact that  $\mathbb{E}_{\mathbf{m}}[\langle \mathbf{m}, \tilde{\mathbf{a}} \rangle] = 0$ .  $\blacktriangleleft$

**Proof of Lemma 19.** Let  $\mathbf{1}(E)$  denote the 0/1 indicator function of an event  $E$ . We have that

$$\begin{aligned} & \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [U_S V_T U'_{S'} V'_{T'}] \\ &= \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} \left[ \prod_{i \in S \cup T \cup S' \cup T'} U_i^{\mathbf{1}(i \in S)} V_i^{\mathbf{1}(i \in T)} U'_i{}^{\mathbf{1}(i \in S')} V'_i{}^{\mathbf{1}(i \in T')} \right] \\ &= \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} \left[ \prod_{i \in S \cup T \cup S' \cup T'} \mathbb{E}_{\mathbf{m}_i} \left[ U_i^{\mathbf{1}(i \in S)} V_i^{\mathbf{1}(i \in T)} U'_i{}^{\mathbf{1}(i \in S')} V'_i{}^{\mathbf{1}(i \in T')} \right] \right]. \end{aligned} \quad (20)$$

On the other hand, we have that

$$\mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [U_S V_T] = \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{M}} \left[ \prod_{i \in S \cup T} U_i^{\mathbf{1}(i \in S)} V_i^{\mathbf{1}(i \in T)} \right] = \mathbb{E}_{\mathbf{a}, \mathbf{b}} \left[ \prod_{i \in S \cup T} \mathbb{E}_{\mathbf{m}_i} \left[ U_i^{\mathbf{1}(i \in S)} V_i^{\mathbf{1}(i \in T)} \right] \right], \quad (21)$$

$$\text{and similarly, } \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [U'_{S'} V'_{T'}] = \mathbb{E}_{\mathbf{a}', \mathbf{b}'} \left[ \prod_{i \in S' \cup T'} \mathbb{E}_{\mathbf{m}_i} \left[ U'_i{}^{\mathbf{1}(i \in S')} V'_i{}^{\mathbf{1}(i \in T')} \right] \right]. \quad (22)$$

If there exists  $i \in S \cup T \cup S' \cup T'$  that appears in an odd number of  $S$ ,  $T$ ,  $S'$  and  $T'$ , then it can be seen that the expectation in Equation 20 is equal to 0, and that at least one of the expectations in Equations 21 and 22 is equal to 0. This already handles the case that  $S \Delta T \Delta S' \Delta T' \neq \emptyset$ .

Henceforth, we assume that each  $i \in S \cup T \cup S' \cup T'$  appears in an even number of  $S$ ,  $T$ ,  $S'$  and  $T'$ . Assume for ease of notation that  $S \cup T \cup S' \cup T' \subseteq [4d]$ . Define

$$g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') := \mathbb{E}_{\mathbf{m}_i} \left[ U_i^{\mathbf{1}(i \in S)} V_i^{\mathbf{1}(i \in T)} U'_i{}^{\mathbf{1}(i \in S')} V'_i{}^{\mathbf{1}(i \in T')} \right] \quad (23)$$

$$h_i(\mathbf{a}, \mathbf{b}) := \mathbb{E}_{\mathbf{m}_i} \left[ U_i^{\mathbf{1}(i \in S)} V_i^{\mathbf{1}(i \in T)} \right]. \quad (24)$$

$$h'_i(\mathbf{a}', \mathbf{b}') := \mathbb{E}_{\mathbf{m}_i} \left[ U'_i{}^{\mathbf{1}(i \in S')} V'_i{}^{\mathbf{1}(i \in T')} \right]. \quad (25)$$

Combining Equations 20, 21 and 22 along with the definitions in 23, 24 and 25, we get

$$\begin{aligned} & \left| \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [U_S V_T U'_{S'} V'_{T'}] - \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [U_S V_T] \cdot \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [U'_{S'} V'_{T'}] \right| \\ &= \left| \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} \left[ \prod_{i=1}^{4d} g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') - \prod_{i=1}^{4d} h_i(\mathbf{a}, \mathbf{b}) \cdot h'_i(\mathbf{a}', \mathbf{b}') \right] \right| \\ &= \left| \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} \left[ \sum_{j=1}^{4d} \left[ \prod_{i=1}^{j-1} h_i(\mathbf{a}, \mathbf{b}) \cdot h'_i(\mathbf{a}', \mathbf{b}') \prod_{i=j}^{4d} g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') \right] - \prod_{i=1}^j h_i(\mathbf{a}, \mathbf{b}) \cdot h'_i(\mathbf{a}', \mathbf{b}') \prod_{i=j+1}^{4d} g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') \right] \right| \\ &\leq \sum_{j=1}^{4d} \left| \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} \left[ \prod_{i=1}^{j-1} h_i(\mathbf{a}, \mathbf{b}) \cdot h'_i(\mathbf{a}', \mathbf{b}') \prod_{i=j+1}^{4d} g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') \cdot \begin{bmatrix} g_j(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') \\ -h_j(\mathbf{a}, \mathbf{b}) \cdot h'_j(\mathbf{a}', \mathbf{b}') \end{bmatrix} \right] \right| \\ &\leq 4 \cdot d \cdot \sqrt{\tau \cdot \kappa}, \end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality with

$$\begin{aligned}\tau &:= \max_{j \in [4d]} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} \left[ \prod_{i=1}^{j-1} h_i(\mathbf{a}, \mathbf{b})^2 \cdot h_i(\mathbf{a}', \mathbf{b}')^2 \prod_{i=j+1}^{4d} g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}')^2 \right] \\ \kappa &:= \max_{j \in [4d]} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [g_j(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') - h_j(\mathbf{a}, \mathbf{b}) \cdot h_j(\mathbf{a}', \mathbf{b}')]^2\end{aligned}$$

Lemma 20 implies that  $\kappa \leq O(1/n_0)$ . We now show that  $\tau \leq 2^{O(d)}$ . Note that for any  $i \in [n_0]$ , it holds that,

$$\begin{aligned}h_i(\mathbf{a}, \mathbf{b}) &= \begin{cases} \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}} \rangle & \text{if } i \in S \text{ and } i \in T \\ 1 & \text{if } i \notin S \text{ and } i \notin T \\ 0 & \text{otherwise} \end{cases} \\ h'_i(\mathbf{a}', \mathbf{b}') &= \begin{cases} \langle \tilde{\mathbf{a}'}, \tilde{\mathbf{b}'} \rangle & \text{if } i \in S' \text{ and } i \in T' \\ 1 & \text{if } i \notin S' \text{ and } i \notin T' \\ 0 & \text{otherwise} \end{cases} \\ g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}') &= \begin{cases} \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}} \rangle \langle \tilde{\mathbf{a}'}, \tilde{\mathbf{b}'} \rangle + \langle \tilde{\mathbf{a}}, \tilde{\mathbf{a}'} \rangle \langle \tilde{\mathbf{b}}, \tilde{\mathbf{b}'} \rangle + \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}'} \rangle \langle \tilde{\mathbf{a}'}, \tilde{\mathbf{b}} \rangle & \text{if } i \in S \cap T \cap S' \cap T' \\ \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}} \rangle & \text{if } i \in S \cap T, i \notin S' \cup T' \\ \langle \tilde{\mathbf{a}}, \tilde{\mathbf{a}'} \rangle & \text{if } i \in S \cap S', i \notin T \cup T' \\ \langle \tilde{\mathbf{a}}, \tilde{\mathbf{b}'} \rangle & \text{if } i \in S \cap T', i \notin S' \cup T \\ \langle \tilde{\mathbf{a}'}, \tilde{\mathbf{b}} \rangle & \text{if } i \in S' \cap T, i \notin S \cup T' \\ \langle \tilde{\mathbf{a}'}, \tilde{\mathbf{b}'} \rangle & \text{if } i \in S' \cap T', i \notin S \cup T \\ \langle \tilde{\mathbf{b}}, \tilde{\mathbf{b}'} \rangle & \text{if } i \in T \cap T', i \notin S \cup S' \\ 1 & \text{otherwise} \end{cases}\end{aligned}$$

Thus, if we expand out a single term  $\prod_{i=1}^{j-1} h_i(\mathbf{a}, \mathbf{b})^2 \cdot h_i(\mathbf{a}', \mathbf{b}')^2 \prod_{i=j+1}^{4d} g_i(\mathbf{a}, \mathbf{b}, \mathbf{a}', \mathbf{b}')^2$ , we get at most  $3^{8d}$  terms (since each  $g_i$  can increase the number of terms by a factor of at most 3). Each of these terms is the expectation of the product of inner product of some correlated Gaussian vectors. We have from Lemma 13 that each such term is at most  $1 + \delta$  and thus  $\tau \leq 2^{O(d)}$ . Thus, for an explicit choice of  $n_0$  that is upper bounded by  $d^{O(d)}/\delta^2$ , we get that  $4d\sqrt{\tau\kappa} \leq \delta$ , which concludes the proof of the lemma.  $\blacktriangleleft$

### A.3 Mean & Variance Bounds for Multilinear Polynomials

We are now ready to prove Lemma 10. Recall again that,

$$F(\mathbf{M}) = \mathbb{E}_{\mathbf{a}, \mathbf{b}} [A(\mathbf{U}) \cdot B(\mathbf{V})] \quad \text{where, } \mathbf{U} = \frac{M\mathbf{a}}{\|\mathbf{a}\|_2} \text{ and } \mathbf{V} = \frac{M\mathbf{b}}{\|\mathbf{b}\|_2}.$$

We wish to bound the mean and variance of  $F(\mathbf{M})$ . These proofs work by considering the Hermite expansions of  $A$  and  $B$  given by,

$$A(\mathbf{X}) = \sum_{S \subseteq [n]} \hat{A}_S \mathbf{X}_S \quad \text{and} \quad B(\mathbf{X}) = \sum_{T \subseteq [n]} \hat{B}_T \mathbf{Y}_T.$$

The basic definitions and facts related to Hermite polynomials were given in section 2.

**Proof of Lemma 10.** We start out by proving the bound on  $\left| \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right|$ . To this end, we will use Lemma 18 with parameters  $d$  and  $\delta$ . Thus, for a choice of  $n_0 = d^{O(d)}/\delta^2$ ,



we have that,

$$\begin{aligned}
 & \left| \mathbb{E}_{\mathbf{M}} F(\mathbf{M}) - \langle A, B \rangle_{\mathcal{G}_{\rho}^{\otimes n}} \right| \\
 &= \left| \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [A(\mathbf{U}) \cdot B(\mathbf{V})] - \mathbb{E}_{\mathbf{X}, \mathbf{Y} \sim \mathcal{G}_{\rho}^{\otimes n}} [A(\mathbf{X}) \cdot B(\mathbf{Y})] \right| \\
 &= \left| \sum_{S, T \subseteq [n]} \widehat{A}_S \widehat{B}_T \cdot \left( \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [U_S \cdot V_T] - \mathbb{E}_{\mathbf{X}, \mathbf{Y} \sim \mathcal{G}_{\rho}^{\otimes n}} [X_S \cdot Y_T] \right) \right| \\
 &= \left| \sum_{S \subseteq [n]} \widehat{A}_S \widehat{B}_S \cdot \left( \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [U_S \cdot V_S] - \rho^{|S|} \right) \right| \dots (\text{terms corresponding to } S \neq T \text{ are 0.}) \\
 &\leq \sum_{S \subseteq [n]} |\widehat{A}_S \widehat{B}_S| \cdot \delta \quad \dots \dots (\text{using Lemma 18}) \\
 &\leq \|A\|_2 \cdot \|B\|_2 \cdot \delta \quad \dots \dots (\text{Cauchy-Schwarz inequality}) \\
 &\leq \delta \quad \dots \dots (\|A\|_2, \|B\|_2 \leq 1) \quad \blacktriangleleft
 \end{aligned}$$

We now move to proving the bound on  $\text{Var}_{\mathbf{M}}(F(\mathbf{M}))$ . To this end, we will use Lemma 19 with parameters  $d$  and  $\delta/9^d$ . Thus, for a choice of  $n_0 = d^{O(d)}/\delta^2$ , we have that,

$$\begin{aligned}
 & \mathbb{E}_{\mathbf{M}} \left( \mathbb{E}_{\mathbf{a}, \mathbf{b}} A(\mathbf{U}) \cdot B(\mathbf{V}) \right)^2 - \left( \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} A(\mathbf{U}) \cdot B(\mathbf{V}) \right)^2 \\
 &= \left| \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [A(\mathbf{U})B(\mathbf{V})A(\mathbf{U}')B(\mathbf{V}')] - \left( \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [A(\mathbf{U})B(\mathbf{V})] \right) \cdot \left( \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [A(\mathbf{U}')B(\mathbf{V}')] \right) \right| \\
 &\leq \sum_{\substack{S, T \subseteq [n] \\ S', T' \subseteq [n]}} \left| \widehat{A}_S \widehat{B}_T \widehat{A}_{S'} \widehat{B}_{T'} \right| \cdot \left| \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [U_S V_T U_{S'} V_{T'}] - \left( \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}, \mathbf{b}} [U_S V_T] \right) \cdot \left( \mathbb{E}_{\mathbf{M}} \mathbb{E}_{\mathbf{a}', \mathbf{b}'} [U_{S'} V_{T'}] \right) \right| \\
 &\leq \frac{\delta}{9^d} \cdot \sum_{\substack{S, T, S', T' \subseteq [n] \\ S \Delta T \Delta S' \Delta T' = \emptyset}} \left| \widehat{A}_S \widehat{B}_T \widehat{A}_{S'} \widehat{B}_{T'} \right|.
 \end{aligned}$$

To finish the proof, we will show that,

$$\sum_{\substack{S, T, S', T' \subseteq [n] \\ S \Delta T \Delta S' \Delta T' = \emptyset}} \left| \widehat{A}_S \widehat{B}_T \widehat{A}_{S'} \widehat{B}_{T'} \right| \leq 9^d \cdot \|A\|_2^2 \cdot \|B\|_2^2.$$

Define functions  $f : \{1, -1\}^n \rightarrow \mathbb{R}$ ,  $g : \{1, -1\}^n \rightarrow \mathbb{R}$  over the boolean hypercube as,

$$f(x) = \sum_{\substack{S \subseteq [n] \\ |S| \leq d}} \widehat{A}_S \mathcal{X}_S(x) \quad \text{and} \quad g(x) = \sum_{\substack{S \subseteq [n] \\ |S| \leq d}} \widehat{B}_S \mathcal{X}_S(x).$$

Hypercontractivity bounds [55] for degree- $d$  polynomials over the boolean hypercube imply that,

$$\mathbb{E}_x [f(x)^4] \leq 9^d \left( \mathbb{E}_x [f(x)^2] \right)^2 \quad \text{and} \quad \mathbb{E}_x [g(x)^4] \leq 9^d \left( \mathbb{E}_x [g(x)^2] \right)^2.$$

We now finish the proof as follows,

$$\begin{aligned}
 \sum_{\substack{S, T, S', T' \subseteq [n] \\ S \Delta T \Delta S' \Delta T' = \emptyset}} |\widehat{A}_S \widehat{B}_T \widehat{A}_{S'} \widehat{B}_{T'}| &= \mathbb{E}_x [f(x)^2 g(x)^2] \\
 &\leq \left( \mathbb{E}_x [f(x)^4] \right)^{1/2} \cdot \left( \mathbb{E}_x [g(x)^4] \right)^{1/2} \dots (\text{Cauchy-Schwarz}) \\
 &\leq 9^d \cdot \left( \mathbb{E}_x [f(x)^2] \right) \cdot \left( \mathbb{E}_x [g(x)^2] \right) \dots (\text{Hypercontractivity}) \\
 &= 9^d \cdot \|A\|_2^2 \cdot \|B\|_2^2.
 \end{aligned}$$

Thus, overall we get that,  $\text{Var}_{\mathcal{M}}(F(\mathcal{M})) \leq \delta$ .

This completes the proof of Lemma 10 for an explicit choice of  $n_0 \leq d^{O(d)}/\delta^2$ .

## B Proof of Low-Degree Multilinear Transformation Lemma

The goal of this section is to prove Lemma 11, which follows immediately by putting together the following two lemmas. The first lemma transforms general functions to low-degree polynomials and second lemma subsequently transforms it to multilinear polynomials.

► **Lemma 21** (Low Degree Transformation). *Given parameters  $\rho \in [0, 1]$ ,  $\delta > 0$ ,  $k \in \mathbb{N}$ , there exists an explicit  $d = d(\rho, k, \delta)$  such that the following holds:*

*Let  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$ , such that, for any  $j \in [k]$  :  $\text{Var}(A_j), \text{Var}(B_j) \leq 1$ .*

*Then, there exist functions  $\widetilde{A} : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $\widetilde{B} : \mathbb{R}^n \rightarrow \mathbb{R}^k$  such that the following hold.*

1.  $\widetilde{A}$  and  $\widetilde{B}$  have degree at most  $d$ .
2. For any  $i \in [k]$ , it holds that  $\text{Var}(\widetilde{A}_i) \leq \text{Var}(A_i) \leq 1$  and  $\text{Var}(\widetilde{B}_i) \leq \text{Var}(B_i) \leq 1$ .
3.  $\left\| \mathcal{R}(\widetilde{A}) - \widetilde{A} \right\|_2 \leq \left\| \mathcal{R}(A) - A \right\|_2 + \delta$  and  $\left\| \mathcal{R}(\widetilde{B}) - \widetilde{B} \right\|_2 \leq \left\| \mathcal{R}(B) - B \right\|_2 + \delta$
4. For every  $i, j \in [k]$ ,

$$\left| \left\langle \widetilde{A}_i, \widetilde{B}_j \right\rangle_{\mathcal{G}_\rho^{\otimes n}} - \langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \frac{\delta}{\sqrt{k}}$$

*In particular, one may take  $d = O\left(\frac{\sqrt{k} \log^2(k/\delta)}{\delta(1-\rho)}\right)$ .*

► **Lemma 22** (Multi-linear Transformation). *Given parameters  $\rho \in [0, 1]$ ,  $\delta > 0$ ,  $d, k \in \mathbb{Z}_{\geq 0}$ , there exists an explicit  $t = t(k, d, \delta)$  such that the following holds:*

*Let  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$  be degree- $d$  polynomials, such that, for any  $j \in [k]$  :  $\text{Var}(A_j), \text{Var}(B_j) \leq 1$ .*

*Then, there exist functions  $\widetilde{A} : \mathbb{R}^{nt} \rightarrow \mathbb{R}^k$  and  $\widetilde{B} : \mathbb{R}^{nt} \rightarrow \mathbb{R}^k$  such that the following hold:*

1.  $\widetilde{A}$  and  $\widetilde{B}$  are multilinear with degree at most  $d$ .
2. For any  $i \in [k]$ , it holds that  $\text{Var}(\widetilde{A}_i) \leq \text{Var}(A_i) \leq 1$  and  $\text{Var}(\widetilde{B}_i) \leq \text{Var}(B_i) \leq 1$ .
3.  $\left\| \mathcal{R}(\widetilde{A}) - \widetilde{A} \right\|_2 \leq \left\| \mathcal{R}(A) - A \right\|_2 + \delta$  and  $\left\| \mathcal{R}(\widetilde{B}) - \widetilde{B} \right\|_2 \leq \left\| \mathcal{R}(B) - B \right\|_2 + \delta$
4. For every  $i, j \in [k]$ ,

$$\left| \left\langle \widetilde{A}_i, \widetilde{B}_j \right\rangle_{\mathcal{G}_\rho^{\otimes nt}} - \langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \frac{\delta}{\sqrt{k}}$$

*In particular, one may take  $t = O\left(\frac{kd^2}{\delta^2}\right)$ .*

### Simple Proposition for Rounding

Before getting to the proofs of the above lemmas, we present a simple proposition that will be useful. It says that if we have two strategies which are close in  $\ell_2$ -distance, and one of them is *close* to the simplex  $\Delta_k$ , then so is the other. The proof follows by a straightforward triangle inequality.

► **Proposition 23.** For  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $\tilde{A} : \mathbb{R}^n \rightarrow \mathbb{R}^k$  s.t.  $\|A\|_2, \|\tilde{A}\|_2 \leq 1$ , it holds that,

$$\|\mathcal{R}(\tilde{A}) - \tilde{A}\|_2 \leq \|\mathcal{R}(A) - A\|_2 + \|A - \tilde{A}\|_2.$$

### B.1 Transformation to Low-Degree

The key idea behind Lemma 21 is quite standard, that applying a “small” amount of noise (via the Ornstein-Uhlenbeck operator) to a pair of functions doesn’t hurt their correlation “significantly”. In particular, we have the following lemma.

► **Lemma 24.** Let  $P, Q \in L^2(\mathbb{R}^n, \gamma_n)$  and  $\varepsilon > 0$ . There exists  $\nu = \nu(\rho, \varepsilon)$  such that,

$$\left| \langle P, Q \rangle_{\mathcal{G}_\rho^{\otimes n}} - \langle U_\nu P, U_\nu Q \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \varepsilon \cdot \sqrt{\text{Var}[P] \text{Var}[Q]}$$

In particular, one may take  $\nu := (1 - \varepsilon)^{\log \rho / (\log \varepsilon + \log \rho)}$ , or even  $\nu := 1 - C \frac{(1-\rho)\varepsilon}{\log(1/\varepsilon)}$  for some constant  $C > 0$ .

**Proof.** Consider the Hermite expansions of  $P$  and  $Q$ . That is,

$$P(\mathbf{X}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{P}(\sigma) H_\sigma(\mathbf{X}) \quad \text{and} \quad Q(\mathbf{Y}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{Q}(\sigma) H_\sigma(\mathbf{Y}).$$

Using properties of Hermite polynomials, namely,  $U_\nu H_\sigma = \nu^{|\sigma|} H_\sigma$ , we get that,

$$U_\nu P(\mathbf{X}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \nu^{|\sigma|} \hat{P}(\sigma) H_\sigma(\mathbf{X}) \quad \text{and} \quad U_\nu Q(\mathbf{Y}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \nu^{|\sigma|} \hat{Q}(\sigma) H_\sigma(\mathbf{Y}).$$

Our choice of  $\nu$  was to ensure that  $\rho^d (1 - \nu^{2d}) \leq \varepsilon$  for all  $d \in \mathbb{N}$ . Thus, we get that,

$$\begin{aligned} & \left| \langle P, Q \rangle_{\mathcal{G}_\rho^{\otimes n}} - \langle U_\nu P, U_\nu Q \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \\ &= \left| \sum_{\sigma \neq \mathbf{0}} \rho^{|\sigma|} \cdot \hat{P}(\sigma) \hat{Q}(\sigma) \cdot (1 - \nu^{2|\sigma|}) \right| \\ &\leq \sum_{\sigma \neq \mathbf{0}} \left| \hat{P}(\sigma) \hat{Q}(\sigma) \right| \cdot \rho^{|\sigma|} (1 - \nu^{2|\sigma|}) \\ &\leq \varepsilon \cdot \sum_{\sigma \neq \mathbf{0}} \left| \hat{P}(\sigma) \hat{Q}(\sigma) \right| \quad \dots (\text{since, } \rho^d (1 - \nu^{2d}) \leq \varepsilon \text{ for all } d \in \mathbb{N}) \\ &\leq \varepsilon \cdot \sqrt{\text{Var}[P] \text{Var}[Q]} \quad \dots (\text{Cauchy-Schwarz inequality}) \quad \blacktriangleleft \end{aligned}$$

The above lemma transforms general functions into functions which are concentrated on low-degree. Thus, to complete the proof of Lemma 21, we consider the definition of *low-degree truncation*.

► **Definition 25** (Low-degree truncation). Let  $A \in L^2(\mathbb{R}^n, \gamma_n)$  is given by the Hermite expansion  $A(\mathbf{X}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}_\sigma H_\sigma(\mathbf{X})$ . The *degree- $d$  truncation* of  $A$  is defined as the function  $A^{\leq d} \in L^2(\mathbb{R}^n, \gamma_n)$  given by

$$A^{\leq d}(\mathbf{X}) := \sum_{\substack{\sigma \in \mathbb{Z}_{\geq 0}^n \\ |\sigma| \leq d}} \hat{A}_\sigma H_\sigma(\mathbf{X}).$$

That is,  $A^{\leq d}$  is obtained by retaining only the terms with degree at most  $d$  in the Hermite expansion of  $A$ , where recall that for  $\sigma \in \mathbb{Z}_{\geq 0}^n$ , its degree is defined as  $|\sigma| = \sum_{i=1}^n \sigma_i$ . For convenience, define  $A^{>d} := A - A^{\leq d}$ . Also, for vector valued functions  $A$ , we define  $A^{\leq d}$  as the function obtained by applying the above low-degree truncation on each coordinate.

**Proof of Lemma 21.** We obtain  $\tilde{A}$  and  $\tilde{B}$  by first applying some suitable amount of noise to the functions such that the functions have decaying Hermite tails and then truncating the Hermite coefficients corresponding to terms larger than degree  $d$ .

In particular, given parameter  $\delta$ , we first choose  $\varepsilon$  and  $\nu$  in Lemma 24, such that  $\varepsilon = \frac{\delta}{2\sqrt{k}}$  and then  $\nu = 1 - C \frac{(1-\rho)\varepsilon}{\log(1/\varepsilon)}$  as required. We choose  $d$  to be large enough such that  $\nu^{2d} \leq \frac{\delta}{4\sqrt{k}}$ , that is,  $d = O\left(\frac{\log(k/\delta)}{\log(1/\nu)}\right) = O\left(\frac{\sqrt{k} \log^2(k/\delta)}{\delta(1-\rho)}\right)$ . Finally, we let  $\tilde{A} := (U_\nu A)^{\leq d}$  and  $\tilde{B} := (U_\nu B)^{\leq d}$ .

We now verify the four properties required of the lemma.

1. By definition,  $\tilde{A}$  and  $\tilde{B}$  have degree at most  $d$ .
2.  $\text{Var}(\tilde{A}_i) = \sum_{\substack{\sigma \neq \mathbf{0} \\ |\sigma| \leq d}} \nu^{2|\sigma|} \cdot \hat{A}_i(\sigma)^2 \leq \text{Var}(A_i)$ . Similarly,  $\text{Var}(\tilde{B}_i) \leq \text{Var}(B_i)$ .
3. For convenience, define  $\bar{A} := U_\nu A$ , and hence  $\tilde{A} = \bar{A}^{\leq d}$ . Observe that, since  $\Delta_k$  is a convex body,  $\|\mathcal{R}(v) - v\|_2^2$  is a convex function in  $v \in \mathbb{R}^k$ . Thus, we have that,

$$\begin{aligned} \|\mathcal{R}(\bar{A}) - \bar{A}\|_2^2 &= \mathbb{E}_{\mathbf{X} \sim \gamma_n} \|\mathcal{R}(\bar{A}(\mathbf{X})) - \bar{A}(\mathbf{X})\|_2^2 \\ &= \mathbb{E}_{\mathbf{X} \sim \gamma_n} \left\| \mathcal{R} \left( \mathbb{E}_{\mathbf{X}' \sim U_\nu(\mathbf{X})} A(\mathbf{X}') \right) - \mathbb{E}_{\mathbf{X}' \sim U_\nu(\mathbf{X})} A(\mathbf{X}') \right\|_2^2 \\ &\leq \mathbb{E}_{\mathbf{X} \sim \gamma_n} \mathbb{E}_{\mathbf{X}' \sim U_\nu(\mathbf{X})} \|\mathcal{R}(A(\mathbf{X}')) - A(\mathbf{X}')\|_2^2 \quad \dots \text{(using convexity of } \|\mathcal{R}(v) - v\|_2^2 \text{)} \\ &= \mathbb{E}_{\mathbf{X}' \sim \gamma_n} \|\mathcal{R}(A(\mathbf{X}')) - A(\mathbf{X}')\|_2^2 \\ &= \|\mathcal{R}(A) - A\|_2^2. \end{aligned}$$

Next, observe that,  $\|\bar{A}^{>d}\|_2^2 = \sum_{|\sigma| > d} \nu^{2|\sigma|} \cdot \|\hat{A}(\sigma)\|_2^2 \leq \nu^{2d} \cdot \sqrt{k} \leq \frac{\delta}{4}$ . Thus, we get that,

$$\begin{aligned} \|\mathcal{R}(\tilde{A}) - \tilde{A}\|_2 &\leq \|\mathcal{R}(\bar{A}) - \bar{A}\|_2 + \|\bar{A} - \tilde{A}\|_2 \quad \dots \text{(Proposition 23)} \\ &= \|\mathcal{R}(\bar{A}) - \bar{A}\|_2 + \|\bar{A}^{>d}\|_2 \\ &\leq \|\mathcal{R}(A) - A\|_2 + \delta/4. \end{aligned}$$

Similar argument holds for  $\tilde{B}$ .

4. For every  $i, j \in [k]$ , we simply have from Lemma 24 that

$$\left| \langle \bar{A}_i, \bar{B}_j \rangle_{\mathcal{G}_\rho^{\otimes n}} - \langle A_i, B_j \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \varepsilon = \frac{\delta}{2\sqrt{k}}.$$

Additionally, since  $\|\tilde{A}_i - \bar{A}_i\|_2 \leq \frac{\delta}{4\sqrt{k}}$  and  $\|\tilde{B}_j - \bar{B}_j\|_2 \leq \frac{\delta}{4\sqrt{k}}$ , we get using Lemma 7 that  $\left| \langle \tilde{A}_i, \tilde{B}_j \rangle_{\mathcal{G}_\rho^{\otimes n}} - \langle \bar{A}_i, \bar{B}_j \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \frac{\delta}{2\sqrt{k}}$ . We get the desired statement by combining the two above statements.  $\blacktriangleleft$

## B.2 Transformation to Multi-linear

The key idea behind Lemma 22 is similar to that of Lemma 21 in that, we first apply a transformation on our polynomials that makes it concentrated on multilinear terms, while slightly increasing the number of variables. Subsequently, we apply a *multi-linear truncation* defined as follows.

► **Definition 26** (Multilinear truncation). Suppose  $A \in L^2(\mathbb{R}^n, \gamma_n)$  is given by the Hermite expansion  $A(x) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}_\sigma H_\sigma(x)$ . The *multilinear truncation* of  $A$  is defined as the function  $A^{\text{ml}} \in L^2(\mathbb{R}^n, \gamma_n)$  given by

$$A^{\text{ml}}(x) := \sum_{\sigma \in \{0,1\}^n} \hat{A}_\sigma H_\sigma(x).$$

That is,  $A^{\text{ml}}$  is obtained by retaining only the multilinear terms in the Hermite expansion of  $A$ .

For convenience, also define  $A^{\text{nmml}} := A - A^{\text{ml}}$ . Also, for vector valued functions  $A$ , we define  $A^{\text{ml}}$  as the function obtained by applying the above multilinear truncation on each coordinate.

► **Lemma 27.** *Given parameters  $\rho \in [0, 1]$ ,  $\delta > 0$  and  $d \in \mathbb{Z}_{\geq 0}$ , there exists  $t = t(d, \delta)$  such that the following holds:*

*Let  $A, B \in L^2(\mathbb{R}^n, \gamma_n)$  be degree- $d$  polynomials, such that  $\|A\|_2, \|B\|_2 \leq 1$ . Define polynomials  $\bar{A}, \bar{B} \in L^2(\mathbb{R}^{nt}, \gamma_{nt})$  over variables  $\bar{\mathbf{X}} := \{\mathbf{X}_j^{(i)} : (i, j) \in [n] \times [t]\}$  and  $\bar{\mathbf{Y}} := \{\mathbf{Y}_j^{(i)} : (i, j) \in [n] \times [t]\}$  respectively, as,*

$$\bar{A}(\bar{\mathbf{X}}) := A(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(n)}) \quad \text{and} \quad \bar{B}(\bar{\mathbf{Y}}) := B(\mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(n)})$$

*where  $\mathbf{X}^{(i)} = (\mathbf{X}_1^{(i)} + \dots + \mathbf{X}_t^{(i)})/\sqrt{t}$  and  $\mathbf{Y}^{(i)} = (\mathbf{Y}_1^{(i)} + \dots + \mathbf{Y}_t^{(i)})/\sqrt{t}$ .*

*Since  $(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})$  is distributed according to  $\mathcal{G}_\rho$ , this transformation doesn't change the "structure" of  $A$  and  $B$ . In particular, it follows that,*

$$\langle \bar{A}, \bar{B} \rangle_{\mathcal{G}_\rho^{\otimes nt}} = \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \quad \text{and} \quad \|\bar{A}\|_2 = \|A\|_2 \quad \text{and} \quad \|\bar{B}\|_2 = \|B\|_2$$

*Next, let  $\bar{A}^{\text{ml}}, \bar{B}^{\text{ml}} \in L^2(\mathbb{R}^{nt}, \gamma_{nt})$  be the multilinear truncations of  $\bar{A}$  and  $\bar{B}$  respectively. Then the following hold,*

1.  $\bar{A}^{\text{ml}}$  and  $\bar{B}^{\text{ml}}$  are multilinear with degree at most  $d$ .
2.  $\text{Var}(\bar{A}^{\text{ml}}) \leq \text{Var}(A) \leq 1$  and  $\text{Var}(\bar{B}^{\text{ml}}) \leq \text{Var}(B) \leq 1$ .
3.  $\|\bar{A}^{\text{ml}} - \bar{A}\|_2, \|\bar{B}^{\text{ml}} - \bar{B}\|_2 \leq \delta/2$ .
4.  $\left| \langle \bar{A}^{\text{ml}}, \bar{B}^{\text{ml}} \rangle_{\mathcal{G}_\rho^{\otimes nt}} - \langle A, B \rangle_{\mathcal{G}_\rho^{\otimes n}} \right| \leq \delta$ .

*In particular, one may take  $t = O\left(\frac{d^2}{\delta^2}\right)$ .*

In order to prove Lemma 27, we will need the following multinomial theorem for Hermite polynomials. It can be proved quite easily using the generating function for Hermite polynomials.

► **Fact 28** (Multinomial theorem for Hermite polynomials). *Let  $\beta_1, \dots, \beta_t \in \mathbb{R}$  satisfying  $\sum_{i=1}^t \beta_i^2 = 1$ . Then, for any  $d \in \mathbb{N}$ , it holds that*

$$H_d(\beta_1 X_1 + \dots + \beta_t X_t) = \sum_{\substack{d_1, \dots, d_t \in \mathbb{Z}_{\geq 0} \\ d_1 + \dots + d_t = d}} \sqrt{\frac{d!}{d_1! \dots d_t!}} \cdot \prod_{i=1}^t \beta_i^{d_i} H_{d_i}(X_i).$$

**Proof of Lemma 27.** Before we prove the theorem, we will first understand the effect of the transformation from  $X$  to  $\bar{X}$  for a univariate Hermite polynomial. Instantiating  $\beta_i$ 's in Fact 28 with  $1/\sqrt{t}$ , we get that,

$$H_d\left(\frac{X_1 + \dots + X_t}{\sqrt{t}}\right) = \sum_{\substack{d_1, \dots, d_t \in \mathbb{Z}_{\geq 0} \\ d_1 + \dots + d_t = d}} \sqrt{\frac{d!}{d_1! \dots d_t!}} \cdot \frac{\prod_{i=1}^t H_{d_i}(X_i)}{t^{d/2}}.$$

We will split the terms into multilinear and non-multilinear terms, writing the above as  $H_d^{\text{ml}} + H_d^{\text{nmml}}$ . Note that there are at most  $O\left(\frac{d^2 t^{d-1}}{d!}\right)$  non-multilinear terms (for  $t \gg d^2$ ). Also, note that each coefficient  $\frac{1}{t^{d/2}} \cdot \sqrt{\frac{d!}{d_1! \dots d_t!}}$  is at most  $\sqrt{\frac{d!}{t^d}}$ . Thus, we can bound  $\|H_d^{\text{nmml}}\|_2$  as follows,

$$\|H_d^{\text{nmml}}\|_2^2 = \sum_{\substack{d_1, \dots, d_t \in \mathbb{Z}_{\geq 0} \\ d_1 + \dots + d_t = d \\ \exists i \ d_i \geq 2}} \left( \frac{1}{t^{d/2}} \cdot \sqrt{\frac{d!}{d_1! \dots d_t!}} \right)^2 \leq O\left(\frac{d^2 t^{d-1}}{d!}\right) \cdot \frac{d!}{t^d} \leq O\left(\frac{d^2}{t}\right) \quad (26)$$

More generally, if we consider a term  $\bar{H}_\sigma(\bar{X}) = H_{\sigma_1}(X^{(1)}) \cdot H_{\sigma_2}(X^{(2)}) \dots H_{\sigma_n}(X^{(n)})$ , where each  $X^{(i)} = (X_1^{(i)} + \dots + X_t^{(i)})/\sqrt{t}$ . Let's write  $\bar{H}_\sigma(\bar{X}) = \bar{H}_\sigma^{\text{ml}}(\bar{X}) + \bar{H}_\sigma^{\text{nmml}}(\bar{X})$ , that is, separating out the multilinear and non-multilinear terms. Similarly, for any  $i$ , let  $H_{\sigma_i}(X^{(i)}) = H_{\sigma_i}^{\text{ml}}(X^{(i)}) + H_{\sigma_i}^{\text{nmml}}(X^{(i)})$ . We wish to bound  $\|\bar{H}_\sigma^{\text{nmml}}\|_2$ , which can be done as follows,

$$\begin{aligned} \|\bar{H}_\sigma^{\text{nmml}}\|_2^2 &= \left\| \prod_{i=1}^n (H_{\sigma_i}^{\text{ml}} + H_{\sigma_i}^{\text{nmml}}) - \prod_{i=1}^n H_{\sigma_i}^{\text{ml}} \right\|_2^2 \\ &\leq \prod_{i=1}^n \left( 1 + O\left(\frac{\sigma_i^2}{t}\right) \right) - 1 && \text{(from Equation 26)} \\ &\leq O\left(\frac{|\sigma|^2}{t}\right) && \text{(since, } t \gg |\sigma|^2) \end{aligned}$$

$$\text{Thus, } \|\bar{H}_\sigma^{\text{nmml}}\|_2^2 < \delta^2/4. \quad \text{(for } t = \Theta(d^2/\delta^2)) \quad (27)$$

We are now ready to prove the parts of Lemma 27.

1. It holds by definition that  $\bar{A}^{\text{ml}}$  and  $\bar{B}^{\text{ml}}$  are multilinear. Also, note that the transformation from  $A$  to  $\bar{A}$  and finally to  $\bar{A}^{\text{ml}}$  does not increase the degree. So both  $\bar{A}^{\text{ml}}$  and  $\bar{B}^{\text{ml}}$  have degree at most  $d$ .

2. It is easy to see that  $\text{Var}(\bar{A}) = \text{Var}(A)$ . Since  $\bar{A}^{\text{ml}}$  is obtained by truncating certain Hermite coefficients of  $\bar{A}$ , it immediately follows that  $\text{Var}(\bar{A}^{\text{ml}}) \leq \text{Var}(\bar{A}) = \text{Var}(A) \leq 1$ . Similarly,  $\text{Var}(\bar{B}^{\text{ml}}) \leq \text{Var}(B) \leq 1$ .
3. Recall that  $\bar{A}^{\text{nmml}} = \bar{A} - \bar{A}^{\text{ml}}$ . We wish to bound  $\|\bar{A}^{\text{nmml}}\|_2^2 \leq \delta^2/4$ . Consider the Hermite expansion of  $A$ , namely  $A(\mathbf{X}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}(\sigma) \cdot H_{\sigma}(\mathbf{X})$ . Note that,  $\bar{A}^{\text{nmml}}(\bar{\mathbf{X}}) = \sum_{\sigma \in \mathbb{Z}_{\geq 0}^n} \hat{A}(\sigma) \cdot \bar{H}_{\sigma}^{\text{nmml}}(\bar{\mathbf{X}})$ , where recall that  $\bar{H}_{\sigma}^{\text{nmml}}$  is the non-multilinear part of  $\bar{H}_{\sigma}(\bar{\mathbf{X}}) = H_{\sigma_1}(X^{(1)}) \cdot H_{\sigma_2}(X^{(2)}) \cdots H_{\sigma_n}(X^{(n)})$ , where each  $X^{(i)} = (X_1^{(i)} + \cdots + X_t^{(i)})/\sqrt{t}$ .

From Equation 27, we have that for any  $\sigma \in \mathbb{Z}_{\geq 0}^n$ , it holds that  $\|\bar{H}_{\sigma}^{\text{nmml}}\|_2^2 < \delta^2/4$ . And hence we get that,

$$\|\bar{A}^{\text{nmml}}\|_2^2 = \sum_{\sigma} \hat{A}(\sigma)^2 \cdot \|\bar{H}_{\sigma}^{\text{nmml}}\|_2^2 \leq \sum_{\sigma} \hat{A}(\sigma)^2 \cdot (\delta^2/4) = (\delta^2/4) \|A\|_2^2 \leq (\delta^2/4).$$

Note that, here we use that  $\bar{H}_{\sigma}(\bar{\mathbf{X}})$  are mutually orthogonal for different  $\sigma$ . Similarly, we can also get that  $\|\bar{B}^{\text{nmml}}\|_2^2 \leq \delta^2/4$ .

4. Note that we already have,

$$\langle \bar{A}, \bar{B} \rangle_{\mathcal{G}_{\rho}^{\otimes nt}} = \langle A, B \rangle_{\mathcal{G}_{\rho}^{\otimes n}}.$$

And combining Part 3 and Lemma 7, we immediately get that

$$\left| \langle \bar{A}^{\text{ml}}, \bar{B}^{\text{ml}} \rangle_{\mathcal{G}_{\rho}^{\otimes nt}} - \langle \bar{A}, \bar{B} \rangle_{\mathcal{G}_{\rho}^{\otimes nt}} \right| \leq \delta$$

where we use that  $\|\bar{B}^{\text{ml}}\|_2 \leq \|\bar{B}\|_2 \leq 1$  and  $\|\bar{A}^{\text{ml}}\|_2 \leq \|\bar{A}\|_2 \leq 1$ . ◀

**Proof of Lemma 22.** We apply the transformation in Lemma 27, with parameter  $\delta$  being  $\delta/\sqrt{k}$ , to each of the  $k$ -coordinates of  $A : \mathbb{R}^n \rightarrow \mathbb{R}^k$  and  $B : \mathbb{R}^n \rightarrow \mathbb{R}^k$  to get polynomials  $\tilde{A} : \mathbb{R}^{nt} \rightarrow \mathbb{R}^k$  and  $\tilde{B} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ . Namely, for any  $j \in [k]$ , we set  $\tilde{A}_j(\bar{\mathbf{X}}) = \bar{A}_j^{\text{ml}}(\bar{\mathbf{X}})$  and  $\tilde{B}_j(\bar{\mathbf{Y}}) = \bar{B}_j^{\text{ml}}(\bar{\mathbf{Y}})$  as described in Lemma 27.

It is easy to see that parts 1, 2, 4 follow immediately from the conditions satisfied in Lemma 27. For part 3, we have that  $\|\bar{A}_j^{\text{ml}} - \bar{A}_j\|_2 \leq \delta/\sqrt{k}$  for every  $j \in [k]$ , which implies that  $\|\bar{A}^{\text{ml}} - \bar{A}\|_2 \leq \delta$ . Using Proposition 23, we immediately get that,

$$\|\mathcal{R}(\bar{A}^{\text{ml}}) - \bar{A}^{\text{ml}}\|_2 \leq \|\mathcal{R}(\bar{A}) - \bar{A}\|_2 + \delta.$$

Finally, it is a simple observation that  $\|\mathcal{R}(\bar{A}) - \bar{A}\|_2 = \|\mathcal{R}(A) - A\|_2$ , and hence,

$$\|\mathcal{R}(\tilde{A}) - \tilde{A}\|_2 \leq \|\mathcal{R}(A) - A\|_2 + \delta.$$

Similarly,  $\|\mathcal{R}(\tilde{B}) - \tilde{B}\|_2 \leq \|\mathcal{R}(B) - B\|_2 + \delta$ . This concludes the proof. ◀

